GOLDENSOFTWARE®

Voxler® 4

3D Well & Volumetric Data Visualization

Full User's Guide

Voxler® Registration Information

Your **Voxler** serial number is located on the CD cover or in the email download instructions, depending on how you purchased **Voxler**.

Register your **Voxler** serial number online at www.GoldenSoftware.com. This information will not be redistributed.

Registration entitles you to free technical support, free minor updates, and upgrade pricing on future **Voxler** releases. The serial number is required when you run **Voxler** the first time, contact technical support, or purchase **Voxler** upgrades.

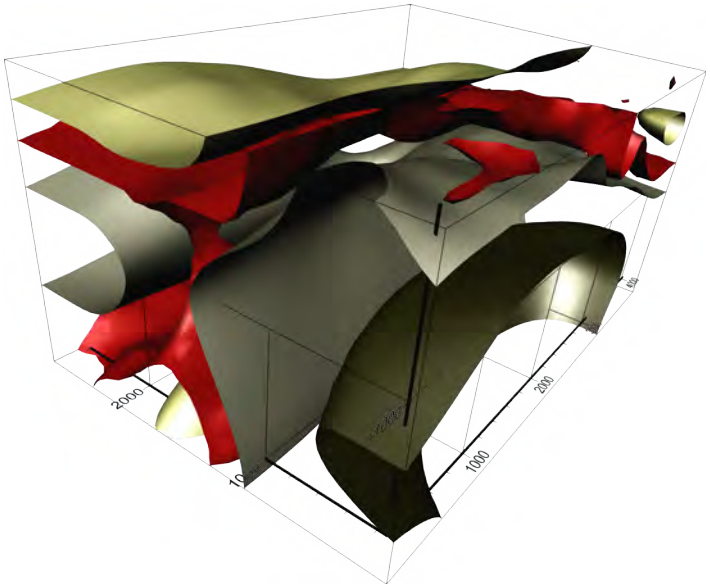
For future reference, write your serial number on the line below.

_____

Voxler®

Full User's Guide

3D Data Visualization



Golden Software, LLC
809 14th Street, Golden, Colorado 80401-1866, U.S.A.
Phone: 303-279-1021 Fax: 303-279-0909
www.GoldenSoftware.com

COPYRIGHT NOTICE

Copyright Golden Software, LLC 2015

The **Voxler**® program is furnished under a license agreement. The **Voxler** software, quick start guide, and full user's guide may be used or copied only in accordance with the terms of the agreement. It is against the law to copy the software, quick start guide, or full user's guide on any medium except as specifically allowed in the license agreement. Contents are subject to change without notice

Voxler is a registered trademark of Golden Software, LLC. All other trademarks are the property of their respective owners.

October 2015

Table of Contents

Chapter 1 - Introducing Voxler 1

Introduction to Voxler® 1

Who Uses Voxler? 1

System Requirements 2

Welcome to Voxler Help 2

Getting Help from the Help Menu 2

Navigating the Help 2

Obtaining Information about Dialogs and Commands 2

Internet Help Resources 2

What's New in Voxler 3

Complete the Voxler Tutorial 3

Voxler Documentation 3

A Note About the Documentation 3

Welcome to Voxler Dialog 4

Three-Minute Tour 6

View Sample Voxler Files 6

Using Voxler 6

Using Scripter 7

Sample Script Files 7

Voxler User Interface Overview 7

Voxler User Interface 8

Changing the Window Layout 9

Working With Voxler Windows 9

Opening and Closing a Window 10

Resizing a Window 10

Changing The Window Layout 10

Tabbed Documents 14

Tab Behavior 14

Unsaved Changes 14

Tab Style 14

No Tabs 14

Resize Window 14

Menu Bar 15

Title Bar 15
Status Bar 15
Show or Hide the Status Bar 15
Status Bar Sections 16
Import Progress 16
Abort Import 16
Introduction to Modules 17
View All Modules 17
View Applicable Modules 17
Import 17
Viewer Window 18
Module Types 18
The Module Manager 20
Showing All Modules 20
Import 20
Samples 21
Add Modules to the Network Manager 21
Expand and Collapse Folders 21
Using Keyboard Commands 21
Show All Modules 22
The Visualization Network 22
Modules 22
Data Sets 23
Data Flow Model 23
The Network Manager 24
Working in the Network Manager 24
Select and Deselect Modules 25
Context Menus 25
Keyboard Commands 26
Viewer Window Module 26
Update Network 26
Auto Update 27
Connect 27
Example - Connecting a Module 27
Connecting Modules 30

Connect Modules 30
Disconnect Modules 31
Arranging Modules 31
Move an Existing Connection 31
Cancel In-Progress Connection 31
Multiple Input or Output Connections 31
Tips on Working with Modules 31
Mismatch Error 32
The Viewer Window 33
Camera 33
Home Position 33
Projection 34
Headlight 34
Draw Style 34
Transparency Mode 34
AutoZoom 35
Customize Mouse Buttons 35
Common Commands 35
Trackball 35
Rotate Object 36
Stop Rotation 36
Prevent Rotation 36
Record Rotation 36
Pan Window 36
Fit to Window 36
Zoom 36
The Property Manager 37
Property Manager Tabs 37
Title Bar 37
Context-Sensitive Help 37
Applying Module Property Changes with Auto Update and Update Now 37
Properties 38
Resize 39

Help 39
Property Manager Controls 39
New Features 42
Graphical Module Features 42
User Friendly 42
Data Features 42
Import and Export Improvements 42
Automation Updates 43
Chapter 2 - Tutorial 45
Tutorial Introduction 45
Advanced Tutorials 45
A Note About the Documentation 45
Using the Tutorial with the Demo Version 46
Starting Voxler 46
Lesson 1 - Loading Data 46
Lesson 2 - Creating Graphics Output Modules 48
2.1 - Creating a Scatter Plot - Tutorial 48
2.2 - Adding a Bounding Box - Tutorial 48
Lesson 3 - Changing Properties 49
3.1 - Changing Symbol Colors - Tutorial 50
3.2 - Displaying Labels - Tutorial 50
3.3 - Changing the Bounding Box Properties - Tutorial 51
3.4 - Rotating Graphics - Tutorial 52
Lesson 4 - Editing Linked Data in the Worksheet 52
Lesson 5 - Using Computational Modules 53
5.1 - Gridding Data - Tutorial 53
5.2 - Creating an Isosurface - Tutorial 54
5.3 - Changing the Isosurface Properties - Tutorial 55
5.4 - A Note About Transparency - Tutorial 57
5.5 - Filtering Data - Tutorial 57
Lesson 6 - Connecting Multiple Modules 60
6.1 - Adding Contours - Tutorial 60
6.2 - Changing the Transparency - Tutorial 62
Lesson 7 - Saving Information 63
7.1 - Saving a Project - Tutorial 63

7.2 - Saving Data - Tutorial 64
7.3 - Saving a Graphic - Tutorial 64
7.4 - Copying a Snapshot - Tutorial 64
7.5 - Capturing a Video - Tutorial 65
Lesson 8 - Importing and Displaying Wells 65
8.1 - Importing Well Collar Data 66
8.2 - Importing Trajectory Data 67
8.3 - Displaying the Wells 67
8.4 - Importing Log Data 67
8.5 - Displaying Log Data on the Wells 68
Advanced Tutorials 68
HeightFields - Advanced Suggestions 69
Math Module - Advanced Suggestions 70
Tutorial Completion 72
Chapter 3 - General Modules 73
Introduction to Modules 73
View All Modules 73
View Applicable Modules 73
Import 73
Viewer Window 74
Module Types 74
General Modules 75
Info 75
Inputs 76
Outputs 76
Properties 76
General Options 76
Info Module - Geometry Page 78
Light 79
Inputs 79
Outputs 79
Properties 79
Viewer Window Properties 82
Changing the Default Viewer Window Properties 82
Editing the Viewer Window Properties 83

Camera Properties....86
Change View with Camera Properties Dialog....86
Change View with Mouse....86
Default Home Position....86
Headlight....87
World Axis Triad....87
Customize Axis Colors....88
Chapter 4 - Data Source Modules....89
Introduction to Modules....89
View All Modules....89
View Applicable Modules....89
Import....89
Viewer Window....90
Module Types....90
Data Types....91
Point Sets....91
Well Data....92
Lattices....92
Geometry....95
Blanked Data....95
Source Modules....96
Inputs....96
Outputs....96
Properties....96
Reloading a Source Module File....96
Changing a Source Module File....96
Import....96
The Import Dialog....97
Data Import Options Dialog....98
File Formats....98
Data Types....98
Loading Multiple Images....99
Image Coordinates....99
Data Import Options Dialog....99
Data Source Module....102

Input Modules 102
Output Modules 103
Properties 103
General Options 103
General Page - Points 105
General Page - Wells 107
WellData Module 110
Inputs 110
Outputs 110
Properties 111
Inputs Options 111
Displaying WellData 112
WellData Module - Wells Page 113
WellData Module - Well Paths Page 114
How to Import and Display Wells 115
Modifying Wells 118
Gridding and Mapping WellData 119
Definitions 120
Lattice Source Module 122
Inputs 122
Outputs 122
Properties 122
General Options 123
Lattice Source Module - Geometry Page 124
Lattice Size Example 125
Lattice View Dialog 126
How can I create an ASCII XYZC lattice file? 127
Geometry Source Module 128
Inputs 128
Outputs 128
Properties 128
Point Source Module 130
Input Modules 130
Output Modules 130

Properties 130
General Options 131
Point Source Module - Geometry Page 132
Data Source Commands 133
Import Data, Well Data, Geometry, or Lattices 133
FunctionLattice 133
TestLattice 135
Geometry Page 138
Data View Dialog 139
Resize Window 140
Close Window 140
How can I display 3D lines? 140
How can I display geometric shapes in Voxler? 141
Other Shapes 149
Chapter 5 - Computational Modules 151
Introduction to Modules 151
View All Modules 151
View Applicable Modules 151
Import 151
Viewer Window 152
Module Types 152
Computational Modules 153
ChangeType 154
Inputs 154
Outputs 154
Properties 154
DuplicateFilter 156
Inputs 156
Outputs 156
Properties 156
ExclusionFilter 158
Inputs 159
Outputs 159
Properties 159
Examples 160

ExtractPoints 160
Inputs 160
Outputs 160
Properties 161
Filter 162
Inputs 162
Outputs 162
Properties 163
Filter References 173
Gradient 173
Inputs 173
Outputs 173
Properties 174
Gridder 175
Inputs 175
Outputs 175
Properties 176
General Options 176
Gridder Module - Geometry Page 179
Gridder Module - Search Page 181
Anisotropy 183
Gridding Methods 186
Gridding Example 190
Can I create a Gridder lattice using the Nearest Neighbor method? 191
How can I create a lattice from an XYZC data file and save the lattice data? 191
How do I create a surface from my XYZ data file? 192
Math 192
Inputs 193
Outputs 193
Properties 193
General Options 193
Geometry Page 195
Math Module Examples 196
Merge 197

Inputs 197
Outputs 197
Properties 197
General Options 198
Merge Module - Geometry Page 199
Resample 200
Inputs 200
Outputs 200
General Options 201
Resample Module - Geometry Page 202
Slice 203
Inputs 203
Outputs 203
Output Components 203
Properties 204
General Options 204
Cutting Plane 206
Properties 206
Subset 208
Inputs 208
Outputs 208
Properties 209
General Options 209
Subset Module - Geometry Page 210
Transform 211
Inputs 211
Outputs 212
Output Components 212
Properties 212
General Options 213
Transform Module - Transform Page 215
Chapter 6 - Axes, Bounding Boxes, and Clipping Planes 219
Introduction to Modules 219
View All Modules 219

View Applicable Modules 219
Import 220
Viewer Window 220
Module Types 220
Graphics Output 221
Axes 222
Inputs 222
Outputs 222
Properties 223
General Options 223
Axes Module - X Axis Page 226
Axes Module - Y Axis Page 229
Axes Module - Z Axis Page 232
How do I change the scale of one of the axes? 235
BoundingBox 236
Inputs 236
Outputs 236
Properties 237
General Options 237
Bounding Box Module - Labels Page 239
ClipPlane 240
Inputs 240
Outputs 240
Properties 241
Connecting Multiple Modules to a Single ClipPlane 243
Chapter 7 - Annotation and Text 245
Introduction to Modules 245
View All Modules 245
View Applicable Modules 245
Import 245
Viewer Window 246
Module Types 246
Graphics Output 247
Annotation 248

Inputs 248
Outputs 249
Properties 249
Text 251
Inputs 252
Outputs 252
Properties 252
Chapter 8 - ScatterPlots 255
ScatterPlot Module 255
Inputs 255
Output 255
Properties 255
General Options 256
ScatterPlot Module - Labels Page 259
Label Format 261
Legend 262
How can I change the data min and data max in the legend? 266
Chapter 9 - Contours 269
Contours Module 269
Inputs 269
Outputs 269
Properties 270
General Options 271
Contours - Cutting Plane Page 274
Contours - Legend Page 277
Drawing Contours 281
How can I change the data min and data max in the legend? 281
Chapter 10 - Isosurfaces 283
Isosurface Module 283
Inputs 283
Outputs 283
Properties 284
General Options 284
Isosurface - Legend Page 287

How can I change the data min and data max in the legend? 291
How can I calculate volume? 291
How do I create an Isosurface with different colors above and below a GRD surface? 292
How can I model geologic surfaces? 292
How do I use interval data for formation tops or ore grade? 293
Chapter 11 - Heightfields 295
HeightField Module 295
Inputs 295
Outputs 295
Properties 296
General Options 296
HeightField - Image Overlay Page 298
HeightField - Vector Overlay Page 300
HeightField - Legend Page 301
HeightField Examples 305
Display Contours on HeightField 305
Display Overlay on HeightField 305
Load a Uniform Lattice at the Correct Z Height 305
How do I create a surface from my XYZ data file? 306
I attach a HeightField to a 2D curvilinear lattice, and when I change the scale from 0, the surface is distorted. Why? 306
When I attach a HeightField to a DEM or GRD file, how do I set the Scale so that the HeightField matches the Z axis? 306
Chapter 12 - StreamLines and VectorPlots 309
StreamLines Module 309
Inputs 309
Outputs 309
Properties 310
General Options 310
StreamLines - Seeds Page 313
VectorPlot Module 315
Inputs 315
Outputs 315
Properties 316
General Options 317
Legend 321

Properties 321
Chapter 13 - OrthoImages and ObliqueImages 327
OrthoImage Module 327
Inputs 327
Outputs 327
Properties 328
General Options 328
ObliqueImage Module 330
Inputs 330
Outputs 330
Properties 331
General Options 331
Cutting Plane 333
Properties 333
Legend 336
Properties 336
Chapter 14 - VolRenders 341
VolRender Module 341
Inputs 341
Outputs 341
Properties 342
General Options 342
VolRender - Legend Page 344
How do I blank a lattice above an irregular topography surface? 348
Chapter 15 - FaceRenders 351
FaceRender Module 351
Inputs 351
Outputs 351
Properties 351
General Options 352
FaceRender Module - Geometry Page 354
FaceRender - Legend Page 355
How can I calculate volume? 359
Chapter 16 - WellRenders 361
WellRender Module 361

Inputs 361
Outputs 361
Properties 362
General Options 362
WellRender Module - Directional Data Page 364
WellRender Module - Interval Data Page 367
WellRender Module - Path Data Page 370
How to Import and Display Wells 371
Primary Data Requirements 371
Importing Well Collar Data 372
Importing Trajectory Data 372
Displaying the Wells 373
Importing Log Data 373
Displaying Log Data on the Wells 373
Gridding and Mapping WellData 374
Extracting WellData 374
Gridding WellData 374
Mapping WellData 375
Modifying Wells 375
Importing Wells 375
Importing Additional Wells 375
Hiding Wells 376
Changing Wells 376
Displaying Wells 376
Setting Well Properties 376
Chapter 17 - Views and Altering the View Style 377
Fit to Window 377
Trackball 377
Spin 378
Dragger 378
View 378
Zoom Realtime 378
Pan 378
Customize 378

Exit Zoom Realtime 378
Undo 378
Pan 379
Customize 379
Exit Pan 379
Undo 379
Headlight 379
World Axis Triad 380
Customize Axis Colors 380
Defined Views 381
Perspective Projection 384
Orthographic Projection 384
Still Draw Style 385
Animating Draw Style 388
Transparency Type 393
Default (Fastest) Transparency Type 393
Best Quality Transparency Type 393
Color Blending 393
Smooth Rotation 393
Notes 393
Transparency Types 394
Home 402
Set Home 402
Camera Properties 402
Change View with Camera Properties Dialog 402
Change View with Mouse 403
Default Home Position 403
Dragger 403
Rotate the Dragger Trackball 404
User Defined Rotation 404
Scale the Size of Trackball 404
Offset the Plane 404
How can I prevent objects from spinning? 404
Chapter 18 - Voxler Worksheet 405
Worksheet Document 405
Worksheet Commands 405

Tab View 405
Worksheet Document 406
Worksheet Window 407
Active Cell 408
Active Cell Edit Box 408
Active Cell Location Box 410
Select Entire Worksheet 410
Selecting Cells 410
Selecting Cells with the Keyboard 412
Selecting Cells with the Mouse 413
Selecting a Column or Row Dividing Line 414
Hiding Columns or Rows 415
With the Format Menu 415
Displaying Hidden Columns or Rows 415
Example 415
With the Format Menu 416
Worksheet Error Codes and Special Numeric Values 416
Worksheet Specifications 417
Editing the Worksheet 418
Edit Menu Commands 418
Cut 418
Copy 418
Paste 419
Paste Special 419
Clear 420
Insert 421
Delete 421
Find and Replace 422
Find Next 426
Worksheet Formatting 426
Format Menu Commands 426
Format Cells 426
Format Cells - Number 427
Format Cells - Alignment 430
Format Cells - Background 431

Date Time Formats 432
Column Width 436
Row Height 437
Data Menu Commands 438
Sort 439
Transform 441
Statistics 445
Text To Number 456
Transpose 457
Assign Coordinate System 457
Define Coordinate System 462
Define Unreferenced Coordinate System 464
New Projected Coordinates 465
Projections 466
What is a Coordinate System? 466
Ellipsoids 466
Datums 468
Understanding Local Datums 469
Custom Datum Definition 470
Latitude and Longitude Coordinates 471
Latitude and Longitude in Decimal Degrees 472
Types of Projections 473
Characteristics of Projections 475
Supported Projections 477
Projection References 510
Chapter 19 - Program Options 511
General 511
Colors 515
Mouse 516
Log Path 517
Chapter 20 - Importing, Exporting, and Printing 519
Import 525
The Import Dialog 525
Data Import Options Dialog 527

File Formats 527
Data Types 527
Loading Multiple Images 528
Image Coordinates 528
Import - Worksheet Document 528
Import Data Dialog 528
Data Link Properties 531
Open 536
The Open Dialog 536
Lattice Import Options Dialog 538
Select Format Dialog 540
Save 541
Save As 541
The Save As Dialog 541
Save Data 542
The Export Dialog 543
Tips on Saving Data 544
Save Copy As 545
Export 546
The Export Dialog 546
Export Options Dialog 548
Copy Snapshot 550
Capture Video 550
Capture Video Dialog 551
Video Capture - Advanced Suggestions 552
Page Setup 554
The Page Setup Dialog 554
Page Setup - Worksheet 555
Page Setup (Worksheet) - Page 556
Page Setup (Worksheet) - Margins 557
Page Setup (Worksheet) - Options 559
Print 561
The Print Dialog 561
Print Dialog - Worksheet 563
Chapter 21 - Editing Modules 565
Edit Menu Commands 565

Undo 565
Undo Levels 565
Undo Limitations 565
Redo 565
Clear Undo History 566
Copy 566
Paste 566
Delete Module 566
When I delete a module, why are all of the downstream modules not deleted? 567
Delete All Modules 567
Rename Module 567
Colors and Colormaps 568
Color Palette 568
Colors Dialog 568
Colormap 572
Colormap Editor 573
Chapter 22 - Automating Voxler 581
Introducing Scripter 581
Scripter Windows 581
Code Window 582
Object and Procedure Lists 582
Immediate Window 582
Sheet Tabs 582
Break Bar 582
Status Bar 582
Three-Minute Tour 583
Using Scripter 583
Sample Script Files 583
Working with Scripts 583
New Script 583
New Modules 584
Existing Scripts and Modules 584
Saving Scripts 584
Closing Scripts 584
Writing Scripts 584
Running Scripts 585

Running Scripts from the Command Line....586
Passing a Command Line Argument to the Script....586
Debugging Scripts....587
Viewing Errors....587
Run-Time Errors....587
Script Runs Incorrectly....587
Debug.Print....587
Stop or Pause....588
Viewing Variable Values....588
Changing Variable Values....588
Step....588
Breakpoint....589
Trace....589
Stack....589
Module Files....590
Type Library References....590
Scripter BASIC Language....590
Program Statements....590
Line Continuation....591
Example....591
Comments....591
Example 1....591
Example 2....591
Double Quotes and Text....591
Operators....591
Flow Control....591
IF…END IF....592
IF…END IF....592
SELECT CASE…END SELECT....592
DO…LOOP....592
WHILE…WEND....592
FOR…NEXT....592
FOR EACH…NEXT....592
Optional Arguments....592
Subroutines and Functions....593

Writing Functions 594
Built-in Functions and Procedures 594
Variables 595
Object Variables 596
Array Variables 596
User-Defined Types 597
Global Variables 597
Module Types 597
The '#Uses Line 598
Example 598
Private and Public Definitions 598
Module Properties 598
Defining Object Properties and Methods 599
Class Module Example 599
Creating Dialogs 600
UserDialog Items 601
UserDialog Example 602
Using Scripter Help 603
Suggested Reading - Scripter 604
Visual BASIC Compatibility 604
Automation Model 605
Application Object 605
Application Property 606
CommandApi Object 607
CommandApi Property 608
Construct Method 608
Import 609
CreateModules 611
ModifyModule 613
ConnectModules 614
Export 616
Do Method 618
DoOnce Method 618
Automation Examples 619
Automation Example - HeightField 619
Automation Example - ScatterPlot 623
Automation Example - VolRender 627
Automation Example - WellData and WellRender 631
Appendix A - File Formats 639
File Descriptions 645

GLOBE DEM Data .?10g File Description 645
Esri ArcInfo Binary Grid .ADF File Description 647
Amira Mesh .AM, .COL Files 648
ACR-NEMA Medical Image .AN? File Description 649
Arc/Info ASCII Grid File .ASC, .AIG, .AGR, .GRD Description 650
Amira Stacked Images .ASI File Description 652
Band Interleaved .BIL, .BIP, .BSQ File Description 654
Golden Software Blanking .BLN File Description 655
Bitmap .BMP Files 657
Atlas Boundary .BNA File Description 657
Color Spectrum CLR File Format 659
CPS-3 Grid .ADX, .DAT, .GRD, .CPS File Description 661
DAT .CSV, . DAT, . TXT File Description 662
ASCII Database .DBF File Description 664
SDTS DEM .DDF Files 664
SDTS Topological Vector Profile and Raster Profile .TVP, .DDF File Description 665
SDTS Digital Elevation Model .DEM File Description 667
USGS Digital Elevation Model .DEM File Description 667
DICOM Medical Image .DIC, .DCM, .AN1, .AN2 Files 669
USGS Digital Line Graph .DLG File Description 670
USGS ETOPO5 .DOS, .DAT File Description 673
DTED .DT0, .DT1, .DT2 Files 674
AutoCAD DXF File Description 675
Esri Format .E00 Grid File Description 678
Esri ArcInfo Export Format .E00 File Description 678
ER Mapper .ECW File Description 679
Encapsulated PostScript .EPS File Description 680
ER Mapper Grid .ERS File Description 680
AVS Field .FLD Files 681
Esri Binary Float Grid .FLT, .HDR File Description 682
Graphics Interchange Format .GIF File Description 683
GRIB Weather Data Grid File Description 685
ASCII Grid File Format 686

Geosoft Binary Grid .GRD File Description 687
Surfer Grid File .GRD Files 688
Surfer 6 Grid File Format 689
Surfer 7 Grid File Format 690
Golden Software Boundary .GSB File Description 692
Golden Software Interchange .GSI File Description 693
Golden Software Reference Files 693
Grid Exchange .GXF File Description 693
Hierarchical Data Format .HDF File Description 695
Analyze 7.5 Medical Image .HDR, .IMG Files 696
GTopo30 .HDR, .STX, .DEM File Description 697
HGT NASA SRTM Grid Data File Description 698
Image (Bitmap) File Description 699
ERDAS Imagine .IMG File Description 702
Leica Confocal Scanning .INFO Files 702
Inventor .IV Files 703
JP2 JPG 2000 File Interchange Format .JP2, J2K, .JPC, .JPT, .JPEG2000, .J2000 File Description 704
JPEG Compressed Bitmap .JPG, .JPEG Files 704
Google Earth Keyhole Markup .KML and. KMZ File Description 705
LAS File Description 706
LAS LiDAR Binary File Description 706
Iris Explorer .LAT Files 708
NetCDF .NC File Description 709
MapInfo Interchange Format .MIF File Description 709
Microsoft Access .ACCDB and .MDB File Description 710
ZSoft Paintbrush .PCX File Description 710
Adobe Portable Document Format .PDF File Description 711
Golden Software PlotCall .PLT File Description 712
Stanford Polygon .PLY Files 713
Portable Network Graphic .PNG File Description 715
Portable Any Map .PNM File Description 716
PLOT-3D .P3D, .XYZ Files 716
Sun Raster Image .RAS, .SUN Files 717

RAW Binary Lattice .RAW, .BIN File Description 718
Silicon Graphics (SGI) RGB Image .RGB, .RGBA, .BW Files 719
Idrisi Raster Image .RST, .IMG File Description 720
.SP1 SEG Standard Data Exchange File Description 720
SEG-Y Seismic Data Log .SGY, .SEGY Files 721
Esri Shapefile .SHP File Description 722
LizardTech MrSID .SID File Description 726
Sylk Spreadsheet .SLK Files 727
Metamorph .STK File Description 728
Truevision Targa .TGA File Description 728
Tagged Image File Format .TIF, .TIF File Description 729
Voxler Data .VDAT Files 730
Voxler Project .VOXB Files 731
Visualization Toolkit .VTK Files 731
AVSX .X, .XIMG Files 732
Excel Spreadsheet .XLS, .XLSM, and .XLSX Files 733
XYZ Grid .DAT File Description 734
Z-MAP Plus Grid .ASC, .DAT, .GRD, .XYZ, .ZMAP, .ZYC, .ZYCOR File Description 734
Import Options Dialogs 735
Data Import Options Dialog 735
Lattice Import Options Dialog 738
Microsoft Access .ACCDB and .MDB Import Options Dialog 740
DICOM Import Options Dialog 742
SDTS Topological Vector Profile .TVP Import Options Dialog 742
USGS Digital Line Graph .DLG, .LGO, .LGS Import Options Dialog 744
AutoCAD DXF Import Options Dialog 749
Esri ArcInfo Export Format .E00 Import Options Dialog 752
ER Mapper .ECW Import Options Dialog 753
GRIB Import Options Dialog 755
LAS Import Options Dialog 755
LiDAR Import Filtering Dialog 757
NC NetCDF Import Options Dialog 759
PDF Import Options Dialog 760

PLOT3D Import Options Dialog 761
RAW Import Options Dialog 763
SEG P1 Import Options Dialog 764
SEGY Import Options Dialog 766
LizardTech MrSID .SID Import Options Dialog 769
Excel .XLSX Import Options Dialog 770
Excel .XLS Import Options Dialog 771
Export Options Dialogs 772
Data Export Options Dialog 772
Export Options Dialog 773
Amira Mesh .AM, .COL Export Options 776
Golden Software Blanking .BLN Export Options Dialog 777
USGS Digital Elevation Model .DEM Export Options Dialog 778
AutoCAD DXF Export Options Dialog 780
Encapsulated PostScript .EPS Export Options Dialog 782
AVS Field .FLD Export Options Dialog 784
Graphics Interchange Format .GIF Export Options Dialog 785
Grid Exchange .GXF Export Options Dialog 786
Hierarchical Data Format .HDF Export Options Dialog 787
Analyze 7.5 Medical Image .HDR, .IMG Export Options Dialog 788
JPEG Image Export Options Dialog 789
Export Options Dialog - JPEG-2000 Options Page 791
Iris Explorer .LAT Export Options Dialog 791
PLOT3D Export Options Dialog 793
Portable Network Graphic .PNG Export Options Dialog 794
Raw Export Options Dialog 795
Select Slices Dialog 797
TIFF Image Export Options Dialog 797
Visual Tool Kit .VTK Export Options Dialog 799
Excel .XLS Export Options Dialog 800
Automation Import Options 801
Esri ArcInfo Binary Grid .ADF Import Automation Options 803
Microsoft Access .ACCDB Import Automation Options 803

Amira Mesh .AM, .COL Import Automation Options 804
Analyze 7.5 Medical Image .IMG Import Automation Options 805
Arc/Info ASCII Grid File .ASC, .AIG, .AGR, .GRD Import Automation Options 806
AVS Field .FLD Import Automation Options 807
AVSX .X, .XIMG Import Automation Options 808
Golden Software Blanking .BLN Import Automation Options 809
Bitmap .BMP Import Automation Options 809
Atlas Boundary [.BNA] Import Automation Options 810
Band Interleaved .BIL, .BIP, .BSQ Import Automation Options 810
CPS-3 Grid .ASC, .ADX, .DAT, .GRD, .CPS, .CPS3 Import Automation Options 812
.CSV Import Automation Options 813
.DAT Import Automation Options 814
ASCII Database .DBF Import Automation Options 816
USGS Digital Elevation Model .DEM Import Automation Options 816
DICOM Medical Image .DIC, .DCM, .AN1, .AN2 Import Automation Options 817
USGS Digital Line Graph .DLG, .LGO, .LGS Import Automation Options 818
DTED .DT0, .DT1, .DT2 Import Automation Options 821
AutoCAD .DXF Import Automation Options 821
Esri ArcInfo Export Format .E00 Import Automation Options 822
ER Mapper [.ECW] Import Automation Options 823
ER Mapper Grid .ERS Import Automation Options 824
USGS ETOPO5 .DOS, .DAT Import Automation Options 824
Esri Binary Float Grid .FLT, .HDR Import Automation Options 825
Geosoft Binary Grid .GRD, .GGF Import Automation Options 825
Graphics Interchange Format .GIF Import Automation Options 826
GLOBE DEM Data .?10g Import Automation Options 827
GRIB Global Weather Data Grid Import Automation Options 828
Golden Software Boundary .GSB Import Automation Options 828
Golden Software Interchange .GSI Import Automation Options 828
GTopo30 .HDR, .STX, .DEM Import Automation Options 829
Grid Exchange .GXF Import Automation Options 830
Hierarchical Data Format .HDF Import Automation Options 830
Inventor .IV Import Automation Options 831

JPEG Compressed Bitmap .JPG, .JPEG Import Automation Options 831
JPEG-2000 File Interchange Format .JP2 Import Automation Options 832
.LAS Import Automation Options 832
LAS LiDAR Binary File Import Automation Options 834
Iris Explorer .LAT Import Automation Options 835
Leica Confocal Scanning .INFO Import Automation Options 835
Microsoft Access .MDB Import Automation Options 836
MapInfo Interchange Format .MIF Import Automation Options 837
DICOM Medical Image .DIC, .DCM, .AN1, .AN2 Import Automation Options 837
NetCDF .NC File Import Automation Options 838
PLOT-3D .P3D, .XYZ Import Automation Options 838
ZSoft Paintbrush .PCX Import Automation Options 839
PDF Adobe Import Automation Options 840
Golden Software PlotCall .PLT Import Automation Options 841
Stanford Polygon .PLY Import Automation Options 841
Portable Network Graphic .PNG Import Automation Options 842
Portable Bitmap .PBM, .PGM, .PPM, .PNM Import Automation Options 842
RAW Binary Lattice .RAW, .BIN Import Automation Options 843
Idrisi Raster Image .RST, .IMG Import Automation Options 845
USGS Digital Elevation Model .DEM Import Automation Options 846
SDTS Topological Vector Profile and Raster Profile .TVP, .DDF Import Automation Options 846
SEG-Y Seismic Data Log .SGY, .SEGY Import Automation Options 847
Silicon Graphics (SGI) RGB Image .RGB, .RGBA, .BW Import Automation Options 849
Surfer Grid .GRD Import Automation Options 849
Esri Shapefile .SHP Import Automation Options 850
LizardTech MrSID .SID Import Automation Options 851
Sylk Spreadsheet .SLK Import Automation Options 853
Amira Stacked Images .ASI Import Automation Options 853
Metamorph .STK Import Automation Options 854
Sun Raster Image .RAS, .SUN Import Automation Options 854
Truevision Targa .TGA Import Automation Options 855
Tagged Image File Format .TIF, .TIFF Import Automation Options 856
Text Data .TXT Import Automation Options 856

Voxler Data .VDAT Import Automation Options 858
Visualization Toolkit .VTK Import Automation Options 858
Excel Spreadsheet .XLS Import Automation Options 859
Excel Spreadsheet .XLSX Import Automation Options 860
Z-MAP Plus Grid .ASC, .DAT, .GRD, .XYZ, .ZMAP, .ZYC, .ZYCOR Import Automation Options 860
Automation Export Options 861
Esri ArcInfo Binary Grid .ADF Export Automation Options 862
Amira Mesh Format .AM, .COL Export Automation Options 863
Analyze 7.5 Medical Image Format .IMG Export Automation Options 864
Esri ASCII Grid Format .ASC, .AIG, .AGR, .GRD Export Automation Options 864
AVS Field Format .FLD Export Automation Options 866
AVS X-Image Format .X, .XIMG Export Automation Options 866
Golden Software BLN File Format .BLN Export Automation Options 869
Windows Bitmap File Format .BMP Export Automation Options 870
Atlas Boundary File Format .BNA Export Automation Options 872
Banded Lattice File Format .BIL, .BIP, .BSQ Export Automation Options 872
Comma Separated Variable File Format .CSV Export Automation Options 873
Golden Software Data File Format .DAT Export Automation Options 874
USGS DEM File Format .DEM Export Automation Options 875
AutoCAD DXF Export Automation Options 876
Encapsulated PostScript File Format .EPS Export Automation Options 877
ER Mapper Grid File Format .ERS Export Automation Options 879
Esri Float Grid File Format .FLT Export Automation Options 880
Geosoft Binary Grid File Format .GGF, .GRD Export Automation Options 881
GIF Image File Format .GIF Export Automation Options 882
Grid eXchange File Format .GXF Export Automation Options 884
Hierarchical Data File Format .HDF Export Automation Options 885
SGI OpenInventor File Format .IV Export Automation Options 886
JPEG Compressed Bitmap File Format .JPG Export Automation Options 886
JPEG-2000 File Interchange Format .JP2 Export Automation Options 888
Iris Explorer File Format .LAT Export Automation Options 890
Plot 3D File Format .P3D, .XYZ Export Automation Options 891
ZSoft Paintbrush File Format .PCX Export Automation Options 892

Raster PDF File Format .PDF Export Automation Options 893
Portable Network Graphic File Format .PNG Export Automation Options 895
Image File Format .PNM, .PPM, .PGM, .PBM Export Automation Options 897
RAW Binary Lattice File Format .RAW, .BIN Export Automation Options 899
SGI-RGB Image File Format .RGB, .RGBA, .BW Export Automation Options 899
Sun Image File Format .RAS, .SUN Export Automation Options 901
Golden Software Surfer 6 ASCII Grid File Format .GRD Export Automation Options 903
Golden Software Surfer 6 Binary Grid File Format .GRD Export Automation Options 903
Golden Software Surfer 7 Binary Grid File Format .GRD Export Automation Options 904
SYLK Spreadsheet File Format .SLK Export Automation Options 905
Targa True Vision Image File Format .TGA Export Automation Options 905
Tagged Image File Format .TIF, .TIFF Export Automation Options 907
Text Data File Format .TXT Export Automation Options 909
Golden Software Voxler Data File Format .VDAT Export Automation Options 910
Visualization Toolkit File Format .VTK Export Automation Options 911
Microsoft Excel Spreadsheet File Format .XLS Export Automation Options 911
XYZ Grid File Format .DAT Export Automation Options 912

Appendix B - Mathematical Functions 915

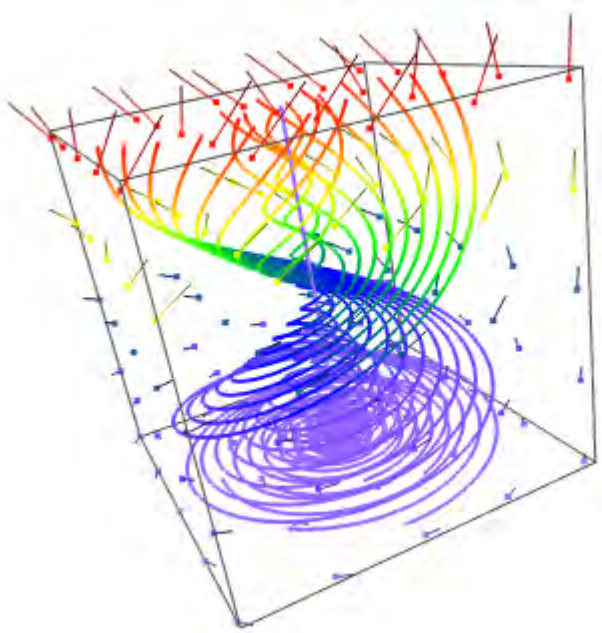
Index 922

Chapter 1 - Introducing Voxler

Introduction to Voxler®

Welcome to **Voxler**, a three-dimensional scientific visualization program oriented primarily toward volumetric rendering and three-dimensional data display. While the emphasis is on three-dimensional volumes, **Voxler** can also utilize two-dimensional grids including DEM files, images, and scattered point data. **Voxler** can display streamlines, vector plots, contour maps, isosurfaces, image slices, three-dimensional scatter plots, direct volume rendering, 3D blocks, well traces, and more. Computational modules include three-dimensional gridding, resampling, numerous lattice operations, and image processing. **Voxler** is designed for displaying XYZC data, where C is a variable at each X, Y, and Z location.

With **Voxler**, you can create stunning graphics output for your true three-dimensional models. Models can be sliced, displayed at any angle, and even animated with a simple mouse movement. Standard or custom colorization can be applied to the models.



Create stunning 3D graphics like this one by combining multiple map types. This example shows a vector plot, stream lines, and a bounding box.

Who Uses Voxler?

People from many different disciplines use **Voxler**. The geosciences generate large amounts of volumetric data from drill cores, seismic studies, ground penetrating radar, subsurface mapping, and remote sensing. Another source of data is from medical imaging generated by CT and MRI scans. Meteorological data, high-resolution microscopy, flow fields, and groundwater modeling are also sources for volumetric data. **Voxler** users include archeologists, climatologists, educators, engineers, doctors, hydrogeologists, geologists, geophysicists, medical researchers, students, and more. Anyone wanting to visualize the relationship of their three-dimensional data with stunning graphical output will benefit from **Voxler's** powerful features!

System Requirements

The minimum system requirements for **Voxler** are:

- Windows XP SP2 or SP3, Vista, 7, 8 (excluding RT), 10, or higher
- 512 MB RAM minimum for simple data sets, 1 GB RAM recommended
- At least 500 MB free hard disk space
- 1024 x 768 or higher monitor resolution with a minimum 16-bit color depth
- Video card with OpenGL acceleration highly recommended

Voxler system requirements can also be found at:
http://www.goldensoftware.com/products/voxler#system-requirements.

Welcome to Voxler Help

There are several ways to obtain help in **Voxler**:

Getting Help from the Help Menu

Within **Voxler**, the online help file is opened with the **Help | Contents** or **Help | Tutorial** commands. Alternatively, press F1 at anytime to open the **Voxler Help** window and access the help books and help pages.

Navigating the Help

You can navigate the **Voxler Help** window using the **Contents**, **Index**, **Search**, and **Favorites** pages. See Help Contents for details on navigating the **Voxler Help** window.

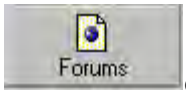
Obtaining Information about Dialogs and Commands

To obtain information about dialogs or highlighted commands:

- Press the F1 button to find out the function of the highlighted menu command or open dialogs.
- Click the  button in dialogs to open the help topic pertaining to that dialog.
- Determine the function of highlighted menu commands or open dialogs by pressing F1.
- Click the  button or press SHIFT+F1 on your keyboard, and click a menu command, toolbar button, or screen region to view information about that item.
- Click the  button at the top of the **Property Manager** to obtain specific information about the selected object type.

Internet Help Resources

There are several Internet help resources.

- Click the 

or 

buttons at the top of the online help to research a question or to post a question.
- Use the **Help | Feedback** commands to send a Problem Report, Suggestion, or Information Request by email.

- Search our website at www.goldensoftware.com or use the **Help | Golden Software on the Web** commands for additional help, including training videos.
- The Golden Software website has a variety of resources including training videos, a support forum, a newsletter, a user image gallery, and a variety of free downloads.

What's New in Voxler

Discover the new features in **Voxler 4** that make the product easier to use. New features are also listed on the Golden Software website.

Complete the Voxler Tutorial

The tutorial is a great way to get started in **Voxler**. The tutorial is designed to introduce you to some of **Voxler's** basic features. After you have completed the tutorial, you should be able to begin creating your own projects. The lessons should be completed in order; however, they do not need to be completed in one session. If you are using the demo version of **Voxler**, you will not be able to complete some of the tutorial steps that require saving or exporting. The demo version is a fully functional read-only version of the program. When this is a factor it is noted in the text and you are directed to proceed to the next step that can be accomplished with the demo.

Voxler Documentation

All of the available **Voxler** user documentation is included in the online help and the quick start guide. Check the **Voxler** support forum, FAQs, knowledge base, and technical support for additional information.

If you prefer printed documentation, the online help file can be printed in part or in full. See Printing the Online Help for more information.

For a full length guide that is arranged in a book-format with an index and table of contents, it is recommended that you purchase the additional user's guide. You can place your order for this guide on our website at www.goldensoftware.com.

A Note About the Documentation

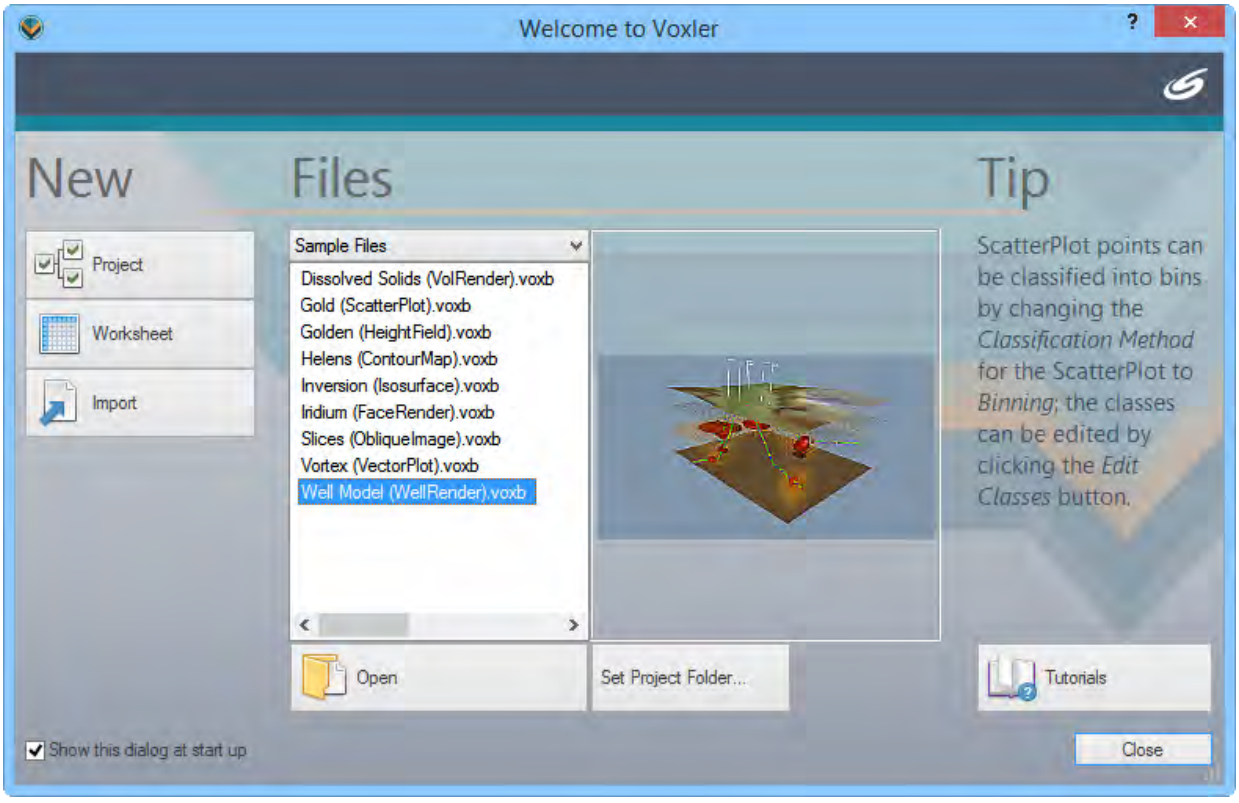
The **Voxler** documentation includes the online help and the quick start guide. Use the **Help | Contents** command in the program to access the detailed online help. Information about each command and feature in **Voxler** is included in the online help. In the event the information cannot be located in the online help, other sources of **Voxler** help include our support forum, frequently asked questions, knowledge base, and contacting our technical support engineers.

Various font styles are used throughout the **Voxler** quick start guide and online help. **Bold** text indicates menu commands, dialog names, and page names. *Italic* text indicates items within a dialog such as modules, group box names, options, and field names. For example, the **Import** dialog contains a *Look in* list. Bold and italic text may occasionally be used for emphasis.

In addition, menu commands appear as **File | Import**. This means, "click on the **File** menu at the top of the plot window and then click on **Import** within the **File** menu list." The first word is always the menu name, followed by the commands within the menu list.

Welcome to Voxler Dialog

The **Welcome to Voxler** dialog is displayed when you first start **Voxler**. The **Welcome to Voxler** dialog provides immediate access to the **File | New | Network**, **File | New | Worksheet**, and **File | Import** commands, sample files, recent files, project files, and the tutorial. The **Welcome to Voxler** dialog also displays a tip about using **Voxler**. Resize the **Welcome to Voxler** dialog by clicking and dragging any side or corner of the dialog.



*The **Welcome to Voxler** dialog helps you get started quickly in **Voxler**.*

New Project

Click the *Project* button to start **Voxler** with a new blank project.

New Worksheet

Click the *Worksheet* button to start **Voxler** with a new worksheet open in the worksheet window.

Import into a New Project

Click the *Import* button to start **Voxler** with a new project and the **Import** dialog open. This button is useful since **File | Import** is often one of the first commands you will use in a new project.

Open Files and File Preview

The *Files* section of the **Welcome to Voxler** dialog displays *Voxler Network VOXB files* in one of three categories. Click the current selection in the *Files Type* list, and select *Recent Files*, *Sample Files*, or *Project Files* to change which files are displayed in the *Files* section. The *Files Type* selection is remembered the next time **Voxler** is started.

Click a file name to see a preview image in the **Welcome to Voxler** dialog. The preview is only available for **Voxler 4** .VOXB files. Click on another file or press the UP ARROW or DOWN ARROW keys to change the file selection. Once a file is selected, the *Open* button is enabled. Click the *Open* button to start **Voxler** with the selected file open in the viewer window. Alternatively, double-click on a file name to start **Voxler** with the selected file open in the viewer window.

If the file you wish to open is not in the *Recent Files*, *Sample Files*, or *Project Files* list, select *Browse* in the *Files Type* list. Next, select the file to open in the **Open** dialog.

Recent Files

When *Recent Files* is selected in the *Files Type* list, the most recently opened **Voxler** files are displayed in the *Files* list. By default the number of files in the *Recent Files* list is 4. Change the number of files in the *Recent Files* list in the **Options** dialog **General** page.

Sample Files

When *Sample Files* is selected in the *Files Type* list, the sample files included with **Voxler** are displayed in the *Files* list.

Project Files

When *Project Files* is selected in the *Files Type* list, the **Voxler** files located in the user defined project folder are displayed in the *Files* list. Change the project folder by clicking the *Set Project Folder* button and selecting the appropriate folder in the **Select Folder** dialog. Alternatively, the *Project folder* option can be set in the **Options** dialog **General** page. However, if the project folder is set in the **Options** dialog, **Voxler** must be closed and reopened for the changes to take effect.

Browse

Select *Browse* in the *Files Type* list to open a file with the **Open** dialog.

Tip

The *Tip* is a useful statement regarding a **Voxler** command or process.

Open

Click the *Open* button to start **Voxler** with the selected file in the *Files* list in the plot window. The *Open* button is enabled when a file is selected in the *Files* list.

Set Project Folder

Change the project folder by clicking the *Set Project Folder* button and selecting the appropriate folder in the **Select Folder** dialog. The **Browse for Folder** dialog will open for users with the Windows XP operating system. The *Files Type* selection is changed to *Project Files* after changing the project folder. If you click *Cancel* in the **Select Folder** dialog, the *Files Type* selection is not changed.

Tutorials

Click the *Tutorials* button to open the online help *Tutorial Introduction* help topic. The tutorial is a useful starting place for users who are new to **Voxler 4**.

Disabling the Welcome to Voxler Dialog

Uncheck the *Show this dialog at start up* check box to start **Voxler** without displaying the **Welcome to Voxler** dialog in the future. The **Welcome to Voxler** dialog can be enabled or disabled by checking or unchecking the *Show welcome on startup* check box in the **Options** dialog **General** page.

Close

Click the *Close* button, the  button, or press ESC to close the **Welcome to Voxler** dialog and start **Voxler** with a new network.

Three-Minute Tour

We have included several sample files with **Voxler** so that you can quickly see some of **Voxler's** capabilities. The sample files do not include all of **Voxler's** many data types, modules, and features. After opening a sample file, the Network Manager is a good source of information as to what is included in each file. Sample files are located at C:\Program Files\Golden Software\Voxler 4\Samples, by default.

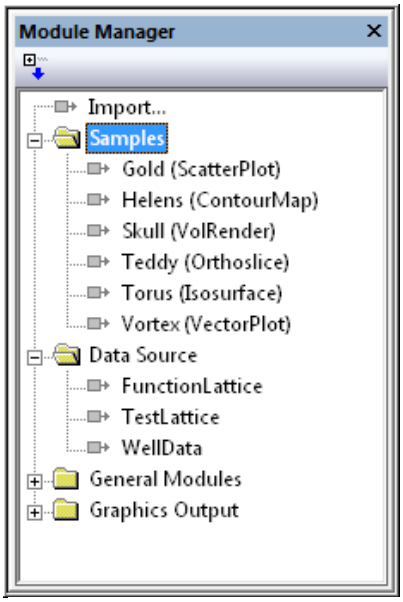
Sample files are a great way to quickly display projects made in **Voxler** by Golden Software. Browse the sample files to get ideas and view different possibilities that **Voxler** has to offer. Sample files can be customized and saved to a new location.

View Sample Voxler Files

To view the example files using the **Module Manager**:

1. Open **Voxler**.
2. Choose the **View | Managers | Module Manager** command to display the **Module Manager**, if it is not already displayed. A check mark indicates the manager is visible. The **Module Manager** is located on the left side of the window by default. The sample files are located in the *Samples* folder at the top of the **Module Manager**.
3. Double-click on a sample file, such as *Torus (Isosurface)* to open the file and display the objects in the Viewer window. Each time you double-click on a file, the new file opens in the **Viewer** window and the previous file closes.

Alternatively, use the **File | Open** command to open example .VOXB files.



*The **Module Manager** is one location to access sample files.*

Using Voxler

To create a three-dimensional model in **Voxler**, you will need to start with data. **Voxler** supports several different data types. Modules are attached to data in order to display the data or make adjustments to the data. Alternatively, data can be gridded to be displayed as isosurfaces, height fields, and image slices. The type of data loaded determines what kind of operations can be performed on it. For detailed information about each type of data, refer to the *Data Types* help page. Refer to the *Introduction to Modules* help page for more information on the types of operations that can be performed.

1. Open **Voxler**.
2. Click the **File | Import** command.
3. In the **Import** dialog, select the data file and click *Open* .
4. In the **Data Import Options** dialog, set the file format options. You can select *Delimiters* and how to treat text. Click *OK*.
5. The data loads into **Voxler** and is displayed as a data module in the **Network Manager**. If you do not see the **Network Manager**, click the **View | Managers | Network Manager** command. A check mark indicates the manager is open. Adding modules to the data set will create a visualization pipeline in the **Network Manager** that will allow you to create the output you desire.
6. Right-click the data module and select **Graphics Output | ScatterPlot**. The output is displayed in the **Viewer** Window.
7. Select the *ScatterPlot* module in the **Network Manager** and the properties are displayed in the **Property Manager**. Adjust the properties as desired.
8. Choose the **File | Save As** command. Enter a *File name* in the **Save As** dialog and click the *Save* button to save your **Voxler** project.

The tutorial lessons contain detailed instructions on using **Voxler**. It is highly recommended that you complete the tutorial before beginning work in **Voxler**. *Advanced tutorial* lessons are also available and recommended.

Using Scripter

Tasks can be automated in **Voxler** using Golden Software's **Scripter** program or any ActiveX Automation-compatible client, such as Visual BASIC. A script is a text file containing a series of instructions for execution when the script is run. **Scripter** can be used to perform almost any task in **Voxler**. You can do practically everything with a script that you can do manually with the mouse or from your keyboard. Scripts are useful for automating repetitive tasks and consolidating a sequence of steps. **Scripter** is installed in the same location as **Voxler**. Refer to the *Voxler Automation* help book for more information about **Scripter**. We have included several example scripts so that you can quickly see some of **Scripter's** capabilities.

Sample Script Files

A variety of automation examples are available. You can run the script as is or you can customize the script. Example scripts are located at C:\Program Files\Golden Software\Voxler 4\Samples\Scripts, by default.

To run a sample script in Scripter:

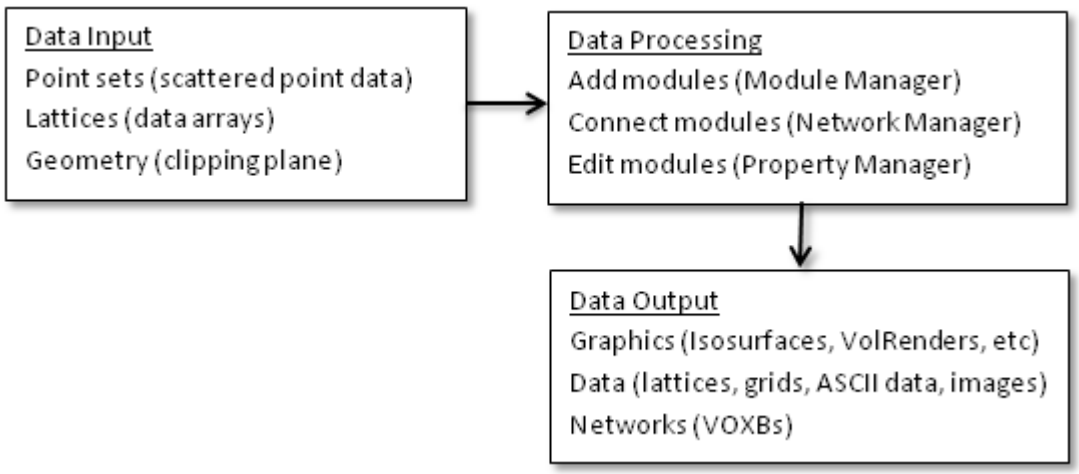
1. Open **Scripter** by navigating to the installation folder, C:\Program Files\Golden Software\Voxler 4\Scripter. Double-click on the Scripter.EXE application file.
2. Click the **File | Open** command and select a sample script .BAS file in the C:\Program Files\Golden Software\Voxler 4\Samples\Scripts folder.
3. Use the **Script | Run** command and the script is executed.

Voxler User Interface Overview

Voxler uses multi-threading to keep the user interface responsive, even with computationally intensive background tasks. The user interface architecture is based on a multiple tabbed documents model. This allows multiple project and worksheet documents to be open simultaneously.

Voxler uses a visualization network to represent data, processing paths, and output. All data and modules for the current project are visible in the **Network Manager**. Select a module to modify the module properties in the **Property Manager**. The graphical output from a module is displayed in the **Viewer** window. The **Network Manager**, **Module Manager**, **Property Manager**, and

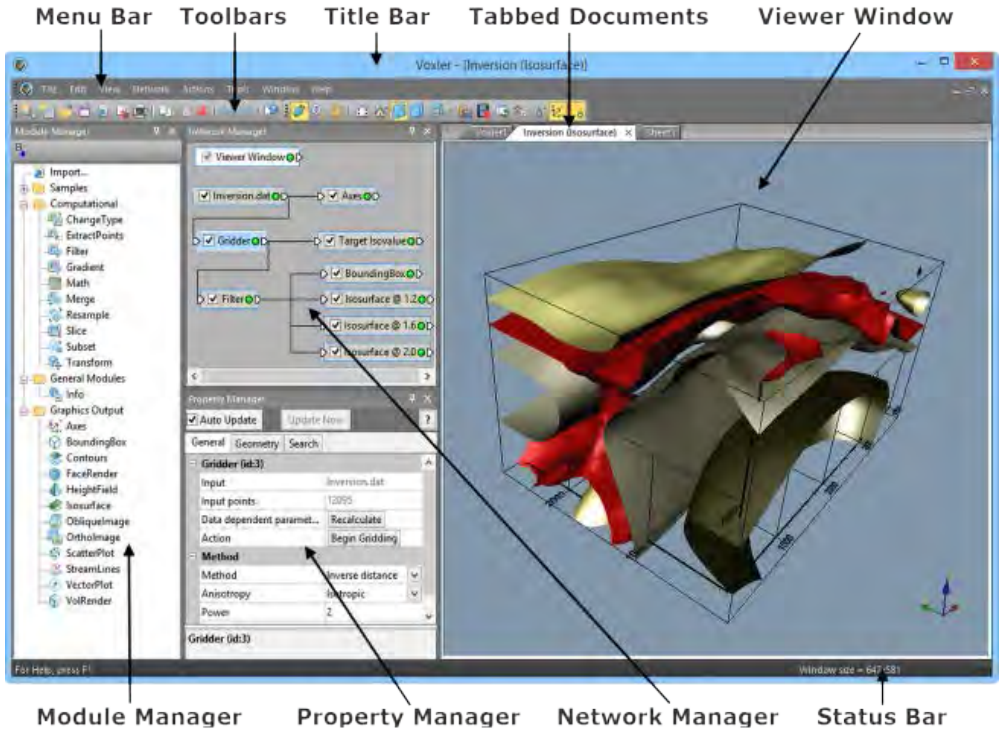
Viewer window are discussed in more detail in the *Voxler User Interface* help topic. **Voxler** files can be saved, exported to a variety of file formats, or recorded using the capture video command.



*Several data types can be loaded into **Voxler**. Once the data are loaded, you can select modules applicable to the data. The data can be visualized and exported.*

Voxler User Interface

The **Voxler** user interface layout consists of the title bar, menu bar, toolbars, **Module Manager**, **Network Manager**, **Property Manager**, **Viewer** window, and status bar. Click on a portion of the image to go to a detailed help page about that component of the user interface.



*The **Voxler** user interface includes several toolbars and windows.*

The following table summarizes the function of each component of the user interface.

Component Name	Component Function
Title Bar	The title bar lists the program name plus the saved **Voxler** .VOXB file name (if any). An asterisk (*) after the file name indicates the file has been modified.
Menu Bar	The menu bar contains the commands used to run **Voxler**.
Toolbars	The toolbars contain **Voxler** tool buttons, which are usually shortcuts to menu commands. Move the cursor over each button to display a screen tip describing the command. Toolbars can be docked or floating.
Module Manager	The **Module Manager** lists example files and provides quick access to modules. The **Module Manager** is initially docked on the far left side of the screen.
Network Manager	The **Network Manager** displays the visualization network, consisting of loaded data files, modules, and connections. The **Network Manager** is initially docked in the middle top position of the screen.
Property Manager	The **Property Manager** displays the properties of the module currently selected in the **Network Manager**. The **Property Manager** is initially docked in the middle bottom position on the screen.
Viewer Window	The **Viewer** window contains the graphics output as directed by the modules in the **Network Manager**. The **Viewer** window is initially located to the far right side of the screen.
Status Bar	The status bar shows information about the activity in **Voxler**. The status bar is divided into three sections. The left section displays help messages and progress text. The middle section displays a progress gauge for various tasks, such as loading large data files. The right section displays the window size and the estimated time remaining for long tasks.

Changing the Window Layout

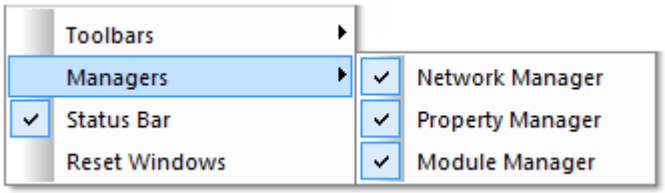
The windows display in a docked view by default; however, they can also be displayed as floating windows. To change the position of a docked window, click the title bar of the window and drag it to a new location. A thin solid black rectangle indicates that the window is docked in the new location. A thick light gray rectangle indicates that the new location is floating. Double-click the window's title bar to toggle between floating and docked modes. See *Changing the Window Layout* for additional information.

Working With Voxler Windows

By default, the **Module Manager** is displayed docked to the left side of the **Voxler** application window, the **Network Manager** is displayed at the upper center, the **Property Manager** is displayed at the lower center, the **Status Bar** is displayed at the bottom , and the **Viewer** window is displayed at the right, as shown in the *Voxler User Interface* topic.

Opening and Closing a Window

If a window is not visible, click the appropriate **View | Managers** menu command. If there is not a check mark next to a particular manager name, **Voxler** does not display that window. Click the manager name to make that manager visible. If there is a check mark next to a manager name and the manager is not visible, the manager may be located off-screen or tabbed behind another window. See *Changing the Window Layout* for more information about displaying managers. Check or uncheck the manager name in the **View | Manager** menu to show or hide the manager.



If there is not a check mark to the left of a command name in the **View** menu, then the window is not displayed. A check mark indicates the window is displayed.

To close a window, click the button or uncheck the manager name in the View menu. Right-click on the following objects to add or remove a check mark and show or hide any of the windows:

- the **Network Manager** title bar;
- the **Module Manager** title bar;
- the **Property Manager** title bar or any area outside the white list of properties;
- the toolbars; or
- the menu bar.

Resizing a Window

Change the size of a window by moving the mouse to the edge of the window. The cursor changes to one of the following cursors:

- docked window horizontal and vertical resizing cursors
- floating window horizontal and vertical resizing cursors

Click and drag the cursors to change the window size or shape.

Changing The Window Layout

The **Voxler** managers, toolbars, and menu bar display in a docked view by default; however, they can also be displayed as floating windows. The visibility, size, and position of each item may also be changed.

Visibility

Use the **View | Toolbars**, **View | Status Bar**, **View | Managers | Network Manager**, **View | Managers | Property Manager**, and **View | Managers | Module Manager** commands to turn these components on or off.

Docked View

To dock a manager or toolbar in a new location, click the grip bar along the edge, hold down the left mouse button, and then drag to a new location.

Floating Window

To display as a floating window, drag the bar away from a window edge. Alternatively, double-click on the window title to toggle between having the window docked and floating.

Restore Previous Position

To restore the original docked position, either double-click on the title bar or click and hold the left mouse button on the title bar and drag the window back to the desired docked location.

Restore Original Position

To restore the default size and position of all managers and toolbars, click the **View | Reset Windows** command. This command is especially handy if your windows or managers become hidden for any reason.

You must restart **Voxler** in order for this command to take effect. Click *Yes* in the dialog, close the program, and reopen **Voxler**. The managers are now in the default locations. To cancel the command, click *No* in the dialog.

Changing Location From Docked to Floating Windows

Voxler windows display in a docked view by default. They can be detached to display as floating windows. To change the position of a docked **Module Manager**, **Network Manager**, or **Property Manager**, click the window title bar and drag it to a new location. Alternatively, double-click the window's title bar to toggle between floating and docked modes.

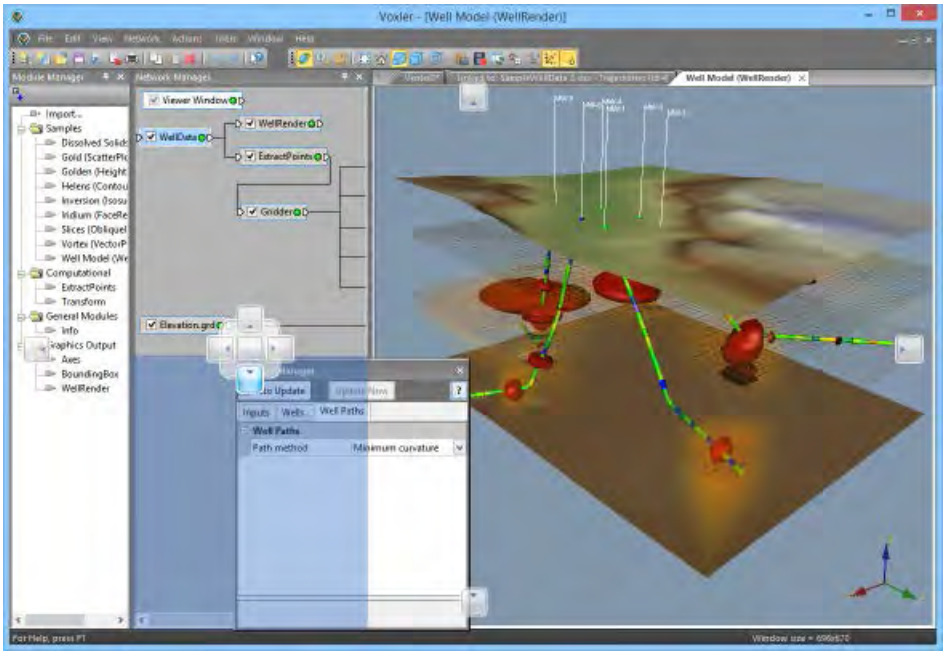
Docking Mechanism

All of the *User interface style* selections in the **Options** dialog display a docking mechanism. To move a manager, left-click the title bar and drag it to a new location while holding down the left mouse button. The docking mechanism displays with arrow indicators as you move the manager. The *Visual Studio 2008*, *Office 2010 Black*, *Office 2010 Blue*, and *Office 2010 Silver* styles use the mechanism on the right in the image below. The other styles use the mechanism on the left.



The docking mechanism allows easy docking of windows.

When the cursor touches one of the docking indicators in the docking mechanism, a blue rectangle shows the window docking position. Release the left mouse button to allow the window to be docked in the specific location.



*In this example, the docking mechanism is being used to position the **Property Manager** back in its default location.*

Creating Stacked and Tabbed Windows

You can view the windows in a stacked or tabbed configuration.

Stacked Windows

If you want to "stack" the windows:

1. Drag one window on top of another window.
2. Position the cursor in either the top or bottom box in the docking mechanism. The screen will turn blue where the manager is to be positioned.

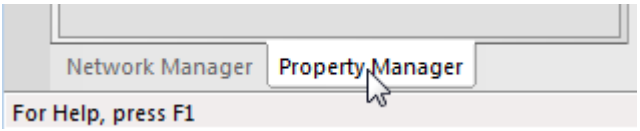
Tabbed Windows

To create tabbed windows:

1. Drag one window on top of another window.
2. Position the cursor in the center of the docking mechanism. The screen will turn blue where the manager is to be positioned. You should see a small tab at the bottom of the managers.

To return to individual windows from tabbed view:

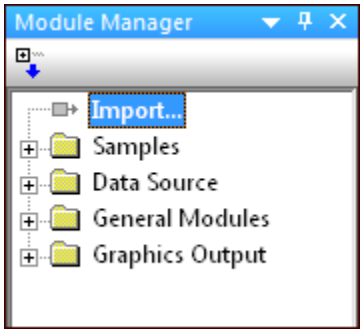
1. Click on the window's name on the tab.
2. Drag the tab to a new location.



Hold down the left mouse button on a window's tab and drag the window to a new location to separate the tabbed windows.

Auto-Hide Windows

You can increase **Viewer** window space by minimizing the other windows with *Auto Hide*. To hide a docked window, click on the button in the upper right corner of the **Module Manager**, **Network Manager**, or **Property Manager** windows. Hold the CTRL key down and click the button to auto hide several docked windows in the same "container." The window hides on the left or right side of **Voxler** with a small tab containing the window name.



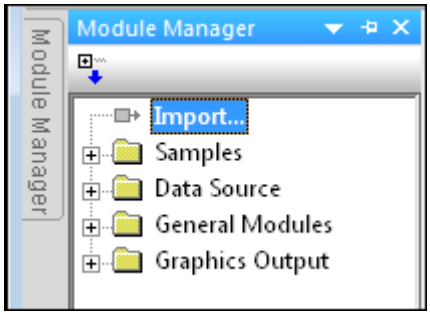
*This example shows the **Module Manager**. Notice the pin button is vertical indicating "show" mode.*

When hidden, the window slides under the left or right side of the **Voxler** main window where it is docked and a tab appears with the window name. Position the mouse pointer over the tab to view the contents of the window. Move your mouse away from the window and the window "hides" again. Click inside the window to anchor it at its current position. Click in another window to release the anchor and hide the window.



*When the button is pressed, the window (**Module Manager** in this example) is hidden on the left side of the screen.*

To view the window, place the cursor directly over the tab. Click in the displayed window to keep it open for use. Move the mouse away from the window or click outside the window to return the window to the hidden position. Click on the button in the upper right corner of a window to return it to the normal display mode and disable the auto hide feature.



When a window is hidden, place the cursor over the title on the tab to display the full window. Notice the pin button is horizontal indicating "hide" mode.

Tabbed Documents

All open **Voxler** worksheet documents and **Viewer** windows are displayed as tabbed documents when the **Tools | Options** *Tab style* is set to a style other than *None.* When more than one window is open, tabs appear at the top of the Viewer window area, allowing you to click on a tab to switch to that window.

Network and worksheet documents can be saved or closed from the context menu by right-clicking on the document tab and selected **Save** or **Close**.

Tab Behavior

When viewing in tabbed document mode, the tabs may be dragged to reorder them. Left-click on a tab, hold the left mouse button, drag to a new location, and release the mouse button to move the tab to a new location.

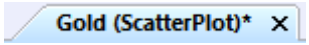
To move to the next tab, press CTRL + F6 to move to the next tab.

The  and  buttons on the sides of the tabs are used to scroll the tabs if there are more tabs than can fit along the top of the window.

Click the  button on the right of the tab to close the selected document.

Unsaved Changes

When a document contains unsaved changes, an asterisk (*) appears next to its tabbed name. The asterisk disappears once the unsaved changes have been saved.



The Gold (ScatterPlot).voxb file has unsaved changes, indicated by the (*) asterisk.

Tab Style

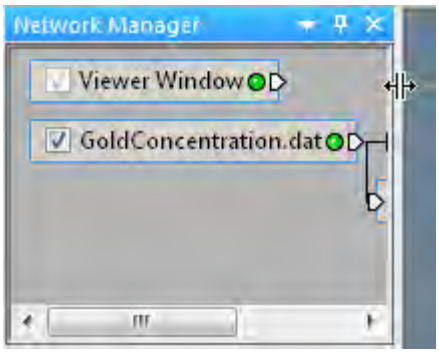
The style of the tab can be changed by clicking the **Tools | Options** and setting the *Tab style*. Select a new tab style from the *Tab style* drop-down list.

No Tabs

Tabs can be turned off by clicking the **Tools | Options** and setting the *Tab style*. Select *None* from the *Tab style* drop-down list.

Resize Window

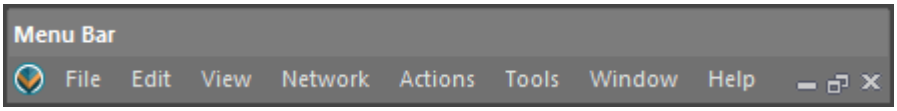
Change the size of a window by moving the mouse to the edge of the window. The cursor changes to the horizontal () or vertical () resizing cursor. Click and drag the cursor to change the window size or shape.



*The ⇹ on the right side of this **Network Manager** indicates the window can be resized by dragging the cursor to the desired size.*

Menu Bar

The menu bar contains the menu commands used to run **Voxler**. The menu bar displays in a docked view by default; however, it can also be displayed as a floating window. See *Changing the Window Layout* for information on displaying the menu bar as either docked or floating.



*The menu bar contains the **Voxler** menu commands.*

Title Bar

The title bar is the top portion of the **Voxler** application window or the top portion of a dialog. The file that is currently open in the **Viewer** window is listed in the program title bar. In a dialog, the dialog name is listed in the dialog title bar. Drag a window or dialog by its title bar to reposition it. Double-click the title bar to maximize or restore the window. When a document contains unsaved changes, an asterisk (*) appears next to its name in the title bar. The asterisk disappears once the unsaved changes have been saved.



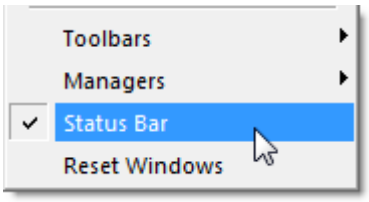
The title bar contains the window or dialog name.

Status Bar

The status bar is displayed at the bottom of the **Voxler** window. The status bar displays additional information about selected items.

Show or Hide the Status Bar

Click the **View | Status Bar** command to show or hide the status bar at the bottom of the **Voxler** window. A check mark appears next to **Status Bar** in the **View** menu when the status bar is visible. Click the **Status Bar** menu item to remove the check mark and hide the status bar.



*Click the **View | Status Bar** command to show or hide the status bar. A check mark indicates the status bar is visible.*

Status Bar Sections

The status bar is divided into three sections: left, middle, and right.

The left side of the status bar displays a short description of the selected menu item. The left section is also used for help messages and progress text.

The middle section is used for a progress gauge to show the amount of work accomplished and the amount left to do for various tasks.

The right section displays the window size of the last redraw in the **Viewer** window. During operations in progress, the right section displays the estimated time remaining for tasks. The window size shows the size of the window screen in pixel width by pixel height.



The status bar commonly appears with the help text on the left; no text in the middle (when there is no progress to report); and window size information on the right.

Import Progress

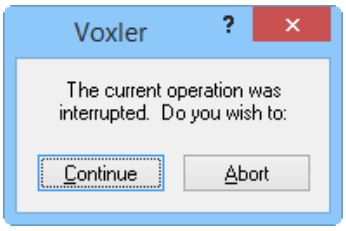
When **Voxler** is importing a data file, the status bar appears with the help message and progress text on the left; a progress gauge in the middle; and the time remaining on the right.



*When **Voxler** is importing a data file, the status bar displays the progress.*

Abort Import

Press the ESC key on the keyboard when the progress gauge is displayed to cancel the file import. A **Voxler** dialog appears with the message *The current operation was interrupted Do you wish to:* Continue or Abort. Click the *Abort* button to cancel the import. Click the *Continue* button to continue the import.

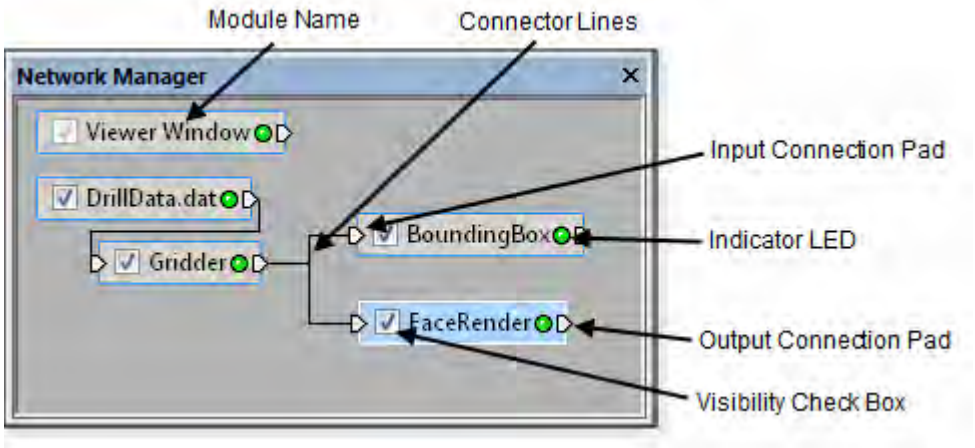


Press the ESC key to abort a file import.

Introduction to Modules

A module is a data set or a process to be applied to a data set or process. Modules are the building blocks from which the final **Voxler** output is constructed. Modules accept data on their input connection pads ▷, modify the data, and pass it along through the output connection pad ▷.

Modules are displayed in the **Network Manager**. You can connect and disconnect modules to create a visualization network representing the flow of data. Modules need to be connected in order to generate an output. The current geometry output of the modules in the **Network Manager** are displayed in a three-dimensional view in the **Viewer** window. If the data is not connected to a graphics output module, nothing is displayed in the **Viewer** window.



*The **Network Manager** displays the visualization network, which includes all loaded modules and their connections.*

View All Modules

All modules are accessed using the Network menu commands. Alternatively, all modules are accessed in the **Module Manager** when the *Show all modules* button is selected .

View Applicable Modules

When a module is selected in the **Network Manager** and the *Show all modules* button is not selected in the **Module Manager**, available modules that can be connected to the selected module output port are displayed in the **Module Manager**. Alternatively, right-click a module in the **Network Manager** to display only the applicable modules in the context menu.

On the module description pages, there are **Inputs** and **Outputs** sections to discuss the type of input and output modules that each module are compatible.

Import

Click **File | Import** to open the **Import** dialog. The type of data determines what type of module can be attached to it. **Voxler** supports several different data types. See the **File Format Chart** for a detailed list of supported file formats.

Viewer Window

The **Viewer Window** is a unique module that is automatically created when a new instance of **Voxler** is generated. The *Viewer Window* module appears in the **Network Manager**. The *Viewer Window* module cannot be deleted. The purpose of the *Viewer Window* module is to control the properties of the **Viewer** window for the current instance of **Voxler**. To change the **Viewer** window properties for future instances of **Voxler**, choose the **Tools | Options** command and adjust properties on the **Colors** page of the **Options** dialog.

Module Types

There are four types of modules: *computational*, *data source*, *general*, and *graphics output.* Each module type is discussed below.

Computational Modules

Computational modules alter the data by changing the data type, filtering, creating a gradient, gridding, performing mathematical transformations, merging, resampling, slicing, creating a subset, or transforming coordinates. Click on one of the following computational modules for detailed information on using the module and module properties.

ChangeType

DuplicateFilter

ExclusionFilter

ExtractPoints

Filter

Gradient

Gridder

Math

Merge

Resample

Slice

Subset

Transform

Data Source Modules

Data source modules serve as a source of raw data. The data can be imported or created from mathematical functions. Click on one of the following data source modules for detailed information on using the module and module properties.

Import (*Data Source*, *Point Source*, *Lattice Source*, *Geometry Source*)

FunctionLattice

TestLattice

WellData (combines multiple *Data Source* modules into a single output)

General Modules

General modules display module information and provide custom lighting in the **Viewer** window. Click on one of the following general modules for detailed information on using the module and module properties.

Info

Light

Graphics Output

Graphics output modules create graphics in the **Viewer** window. Typically, these modules require data input. Click on one of the following graphics output modules for detailed information on using the module and module properties.

Annotation

Axes

BoundingBox

ClipPlane

Contours

FaceRender

HeightField

Isosurface

ObliqueImage

OrthoImage

ScatterPlot

StreamLines

Text

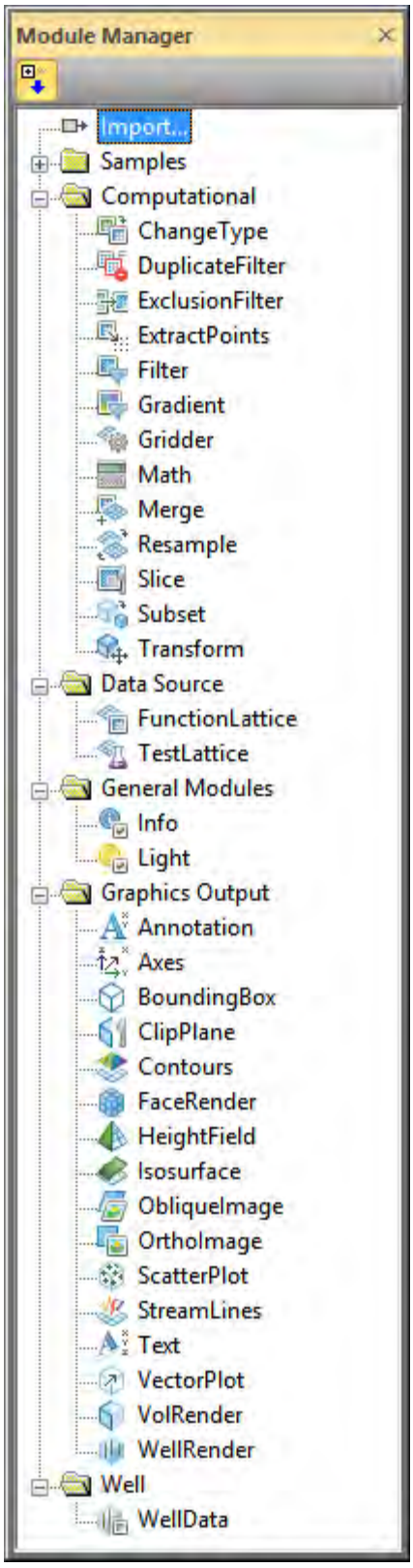
VectorPlot

VolRender

WellRender

The Module Manager

The **Module Manager** displays a list of available modules and other commands. You can add the modules to the **Network Manager** by double-clicking on the module in the **Module Manager** or by clicking on the module name and dragging it into the **Network Manager**. The item is added to the **Network Manager**. Depending on how the module is added, the module may appear in the **Network Manager** already connected to other modules. The **Module Manager** is a quick display of the Network Menu commands. This can be useful to see all of the modules or only the modules compatible with the currently selected module.



The **Module Manager** consists of five sections, displayed as folders: *Import*, *Samples*, *Computational*, *Data Source*, *General Modules*, and *Graphics Output* sections. Depending on which module is currently selected and your settings, you may not have all of these sections available.

Showing All Modules

The toolbar at the top of the **Module Manager** window contains a *Show All Modules* button . If the button is depressed, all modules are listed. If the button is not depressed, only those modules that are compatible with the currently selected module are displayed.

Click the *Show All Modules* button  at the top of the **Module Manager** to expand all folders and list all sample .VOXB files and modules in the tree view list. The button is depressed when this option is enabled. When this option is enabled, all modules are shown in the **Module Manager**. If a module is selected in the **Network Manager** and a compatible module is added, it is automatically connected to the selected module. If the module being added is not compatible with the currently selected module in the **Network Manager**, the new module is added to the **Network Manager** without any connections to existing modules.

If the  button is not depressed, only those modules that are compatible with the currently selected module are displayed. If you double-click on a module, it is automatically connected automatically to the selected module. If no modules are selected in the **Viewer** window, only the *Import*, *Samples*, *Data Source*, *Light*, *Annotation*, and *Text* modules are displayed.

Import

Double-click *Import* to load any file that **Voxler** can import in the **Network Manager**. Select any data file and click the *Open* button to add the file as a new module in the **Network Manager**.

Samples

Several simple sample files are displayed in the *Samples* folder. Double-click an item in the *Samples* folder to open the corresponding .VOXB file in **Voxler**. This is another way of choosing the **File | Open** menu command. If you have an existing **Voxler** project open, you will be prompted to save changes if you open a new sample project. Click the *Yes* button to save the changes to the existing .VOXB file before opening the sample project. Click the *No* button to discard the changes to your existing project and then open the sample project. Click the *Cancel* button to keep the current file open and not open the example project.

Add Modules to the Network Manager

Double-click a module in the *Graphics Output*, *Computational*, *Data Source*, *General Modules*, or *Well* folders in the **Module Manager** to add that module type to the **Network Manager**. If a module is selected in the **Network Manager** and the module in the **Module Manager** is compatible, the two are connected. Otherwise, the module is loaded in the **Network Manager** without connections to any existing modules. You can also click and drag a module from the **Module Manager** to the **Network Manager** to add it without making a connection to the selected module.

Expand and Collapse Folders

The ⊞and ⊟ buttons indicate the folder can be expanded or collapsed to show or hide additional information. This applies to any situation where you see a ⊞and ⊟ in **Voxler**.

There are several ways to expand a folder and see all the items in the folder:

- Click the ⊞ button to the left of the folder name.
- Select the item and press the + key on the numeric keypad.
- Press the right ARROW key on your keyboard.
- Double-click the item.

There are several ways to collapse the folder and view just the folder name:

- Click the ⊟ button to the left of the folder name.
- Select the item and press the - key on the numeric keypad.
- Press the left ARROW key.
- Double-click the item.

Voxler remembers the expanded or collapsed state of the list within a given session and stores the state of the list when you close the program.

Using Keyboard Commands

When working in the **Module Manager**,

- Press the up and down arrow keys to move up and down in the list.
- Press the right arrow key to expand collapsed sections or the left arrow key to collapse the expanded sections.
- Press the PAGE UP key to go to the item currently displayed at the top of the list without scrolling.
- Press the PAGE DOWN key to go to the item currently displayed at the bottom of the list without scrolling.

- Press the HOME key to go to the first item in the list.
- Press the END key to go to the last item in the list.
- To go to a specific line item, press the first letter of the name of the item on your keyboard.
- Click in the **Module Manager** or press ALT+F6 until the **Module Manager** is highlighted. The **Module Manager** must have the focus in order for keyboard commands to apply to its contents.

Show All Modules

Click the *Show All Modules* button  at the top of the Module Manager window to list all modules in the tree control.

Double-click a module in this mode to add it to the Network Manager without connecting it to another module. If the *Show All Modules* button  is not clicked, only those modules that are compatible with the currently selected module are displayed. Double-click one of those modules to automatically connect it to the selected module.

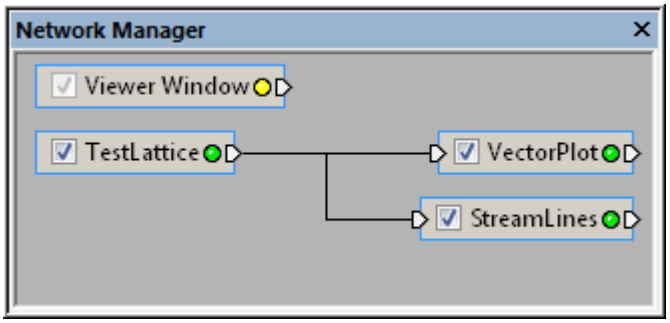
The Visualization Network

Voxler uses a collection of modules and their connections to create a visualization network to represent data and process one or more modules in the **Network Manager**.

Modules

Modules are the building blocks from which the final output is constructed. They consist of input data sets and processes to be applied to the data sets. Modules accept data on their input ports, modify the data, and pass it along through the output ports. The modules link together in an infinite number of ways to form a pipeline that passes the processed data from one module to the next. The final output from the network is usually a graphical representation of the data. This architecture is commonly called a "data flow" model.

Modules have inputs, outputs, and properties. The inputs and outputs are the data types described in the *Data Types* page. Modules are displayed in the Network Manager. The properties of a selected module appear in the Property Manager. When a module's input or properties have changed, it automatically updates in the **Viewer** window. After the module has updated and recomputed its outputs, the framework ensures that all downstream modules are updated as well.



*Modules are display in the **Network Manager**. In this example, the VectorPlot and StreamLines modules are attached to the TestLattice data set.*

Data Sets

Multiple data sets can be loaded into a single **Voxler** project. Each data set can be connected to any number of visualization or computational modules to produce a single scene. Data sets are automatically loaded into source modules. The scene can be interactively manipulated in the three-dimensional **Viewer** window and saved to a file or printed.

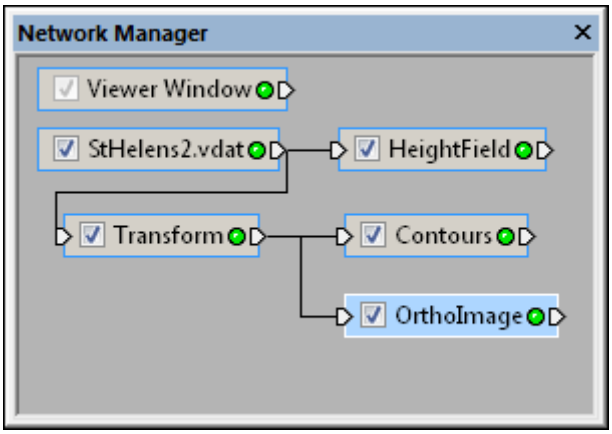
Data Flow Model

Voxler's data flow model allows for incredible flexibility and power. Here are some of the specific advantages:

- The modules can be dynamically connected and edited to see the effect of different parameters on the data. For example, a point set can be loaded, connected to a *Gridder* module, and output to an isosurface. The gridding parameters can be altered to see the new output in the *Isosurface* without having to save grid files. The *Isosurface* automatically recognizes that the grid has changed and automatically updates.
- Multiple modules can be connected to a single input module in order to "stack" the effects. For example, a *Contours* module, *Isosurface* module, and *ClipPlane* module could all be connected to a single lattice module. The output from all three modules is correctly combined in the final scene.
- Network updates execute on a separate work thread. This allows the user interface to stay responsive even while the network is being updated. If a changing property threatens to invalidate the currently updating network, **Voxler** detects and aborts the update in progress. A new update to include the changed properties is then requested.
- There is no need for intermediate files, i.e., grid files. The data are loaded once and need not be resaved in an intermediate form for later reloading.
- Execution through the network can be optimized so only those modules that actually need to execute are triggered. Paths that do not change are cached and do not contribute to the overall execution time. If data or parameters are altered in the middle of the module network, only the downstream modules need to re-execute.
- The **Network Manager** allows the user to easily select modules in the two-dimensional interface.

The Network Manager

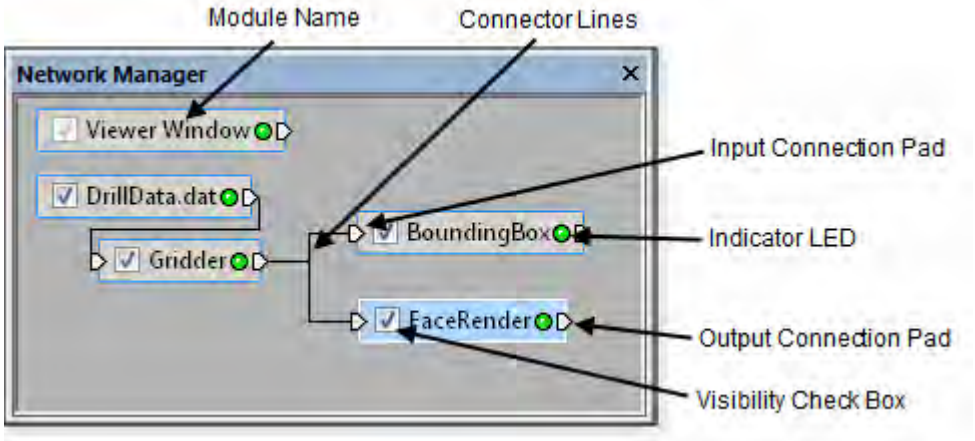
Voxler uses a **Network Manager** to show a graphical representation (also referred to as a visualization network) of data and processes performed in the project. All data, modules, and processing paths for the current project are visible in the **Network Manager**. Modules are connected to perform a desired task. A module is a data set or process to be applied to a data set. Modules accept data on their input connection pads, modify the data, and pass it along through the output connection pads. The final output from the pipeline is usually a graphical representation of data, such as a HeightField or OrthoImage.



*The **Network Manager** displays the visualization network, which includes all loaded modules and their connections.*

Working in the Network Manager

A small rectangle, that can be selected and dragged with the mouse, represents each module in the **Network Manager**. The module rectangles have the following components:



*The sections of the **Network Manager** are marked in the above diagram and described in the table below.*

	Visibility Check Box	The visibility check box indicates whether a module's output is visible in the **Viewer** window. Check the box to display a module and all "downstream" (connected) modules. Uncheck the box to hide a module and all downstream modules. A gray check mark indicates that a module is disabled because of a hidden upstream module.
(Name)	Module Name	Each module is named with the loaded data file name or by the function performed by the module. You can change the name with the **Edit | Rename** command. Alternatively, right-click the module and select **Rename** or press F2 on the keyboard.
	Indicator LED	The indicator LED is a small round "light" showing module status. • Green : the module is up to date • Yellow : the module has been modified and needs to be updated • Red : the module is in an error state
	Connection Pads	An input connection pad is located on the left side of the module. An output connection pad is located on the right side of the module. The presence of connection pads indicates that a module may be connected to other modules. Only modules with the appropriate type of data may be connected.
	Connector Lines	Connector lines are drawn between connected modules. Lines or pipes may be displayed. See the **Tools | Options** dialog to change the display of connector lines.

Select and Deselect Modules

To move a module, save a module, or change any of the module properties, select it by clicking the module rectangle in the **Network Manager**. The selected module is highlighted and its properties are displayed in the **Property Manager**. Note that only a single module may be selected at a time. Click in the **Network Manager** outside all modules to deselect all modules.

Context Menus

Right-click a module to display a context menu containing various commands that can be applied to the module. The commands include:

- A list of modules that may be connected to the selected module. This is the fastest and easiest way to build a network. Simply right-click an existing module and select a module to connect.
- A **Connect** command, which allows you to connect existing modules interactively. The command name changes depending on the selected module type. See *Connecting Modules* for more information about this procedure.
- A **Save Data** command, which allows you to save the module's data output.
- A **Copy** command, which copies all properties of the selected module. This is useful for creating an exact duplicate of an existing module.

- A **Rename** command, which allows you to rename the selected module to any other name.
- A **Delete** command, which removes the selected module from the **Network Manager**. Deleting a module removes the links to other modules.

Right-click in the **Network Manager** without a module selected to display a different context menu that includes commands to import a new file, create standalone modules, or paste a copied module. Standalone modules, such as *Annotation*, do not require an input connection. When right-clicking to add modules, the module appears in the location where you right-clicked.

When an icon is dragged outside the limits of the **Network Manager**, scroll bars are added to allow scrolling to the portions of the network that are not visible.

Keyboard Commands

The following keyboard commands are available in the **Network Manager**:

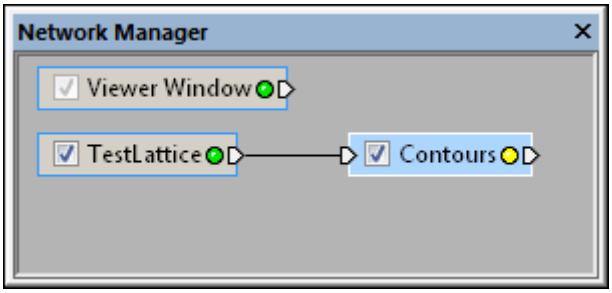
- The DELETE key deletes the selected module if the **Network Manager** is active.
- The TAB key cycles through the modules in the order they were added.
- SHIFT+TAB cycles through the modules in the reverse order.
- The ARROW keys move the selected module.

Viewer Window Module

The **Viewer** window module is automatically created by **Voxler** and cannot be deleted or made invisible in the **Network Manager**. This module is used to control various settings that affect the entire scene, such as background color.

Update Network

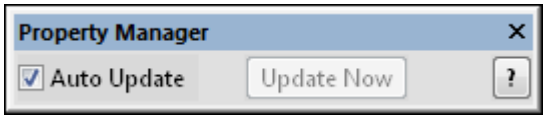
Click the **Network | Update Network** command or press the F9 key on the keyboard to refresh the network if it is out of date. A network is out of date when one of the modules in the **Network Manager** displays a yellow indicator LED. Some modules require action in the **Property Manager** before an update can occur. One example is the *Gridder* module, which requires clicking the *Begin Gridding* button in the **Property Manager** in order to update. Until the properties have been updated, the network does not update, even if the **Network | Update Network** command is used. A green indicator LED indicates the module is up to date.



The yellow indicator LED on the Contours module indicates that the module is out of date. The green indicator LED on the TestLattice module indicates the module is up to date.

Auto Update

The **Network | Update Network** command is useful if the *Auto Update* option in the **Property Manager** is disabled and changes have been made to one or more modules.



*Check the Auto Update box in the **Property Manager** to automatically update the network as changes are made.*

When the *Auto Update* box is not checked, changes made to the modules do not update the network until the **Network | Update Network** menu command is chosen, the *Update Now* button in the **Property Manager** is clicked, or the F9 key is pressed.

This command is disabled (grayed out on the **Network** menu) when the network is up to date.

Connect

There are several ways to connect or disconnect two modules. After you have selected a module:

- Right-click on a module in the **Network Manager** and select **Connect** from the context menu
- Click on the connection pad of a module in the **Network Manager**
- Click the **Network | Connect** command

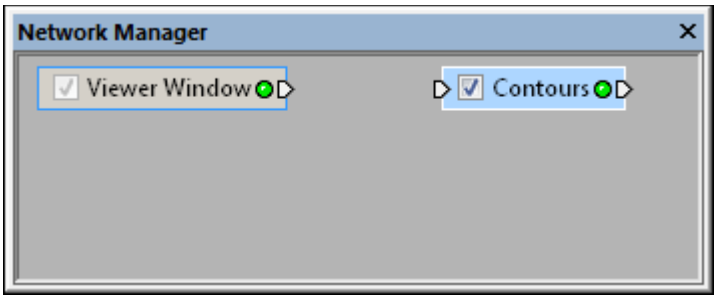
The procedure is the same whether you are connecting or disconnecting two modules. Once a **Connect** command is initiated, the cursor snaps to the **Network Manager** and **Voxler** enters graphical connect mode. Move the mouse until the blue connection line touches the compatible module you want to connect to or disconnect from and the connection line turns yellow. Click the mouse on the module to make or break the connection. The blue connection line turns yellow if the modules are compatible.

The **Connect** command menu text differs depending on which module is selected.

Example - Connecting a Module

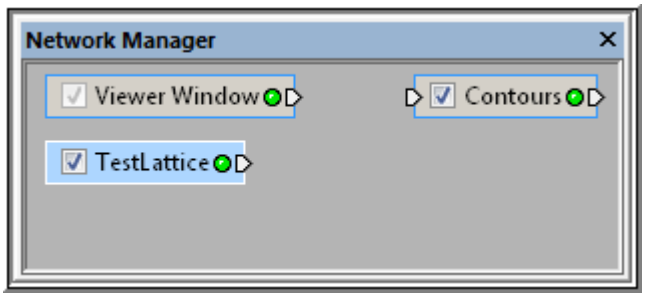
Start with a new window. If a **Voxler** file is already open, click the **File | New** command to open a new blank window.

In the **Module Manager**, click the  button, so all modules are visible. Expand the *Graphics Output* folder. Double-click on the *Contours* module in the **Module Manager** to load the module in the **Network Manager**.



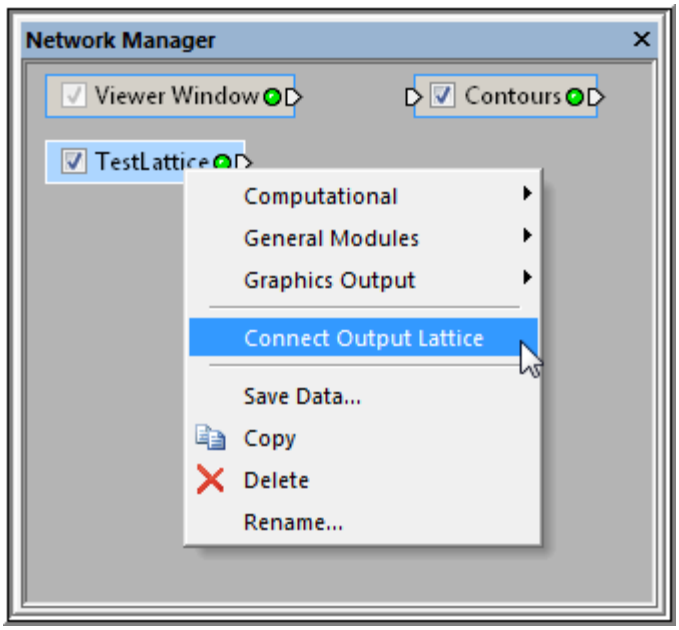
*A module can be added to the **Network Manager** without being connected to any data.*

In the **Module Manager**, expand the *Data Source* folder. Double-click on the *TestLattice* module to load a test lattice. The test lattice is automatically selected in the **Network Manager**.



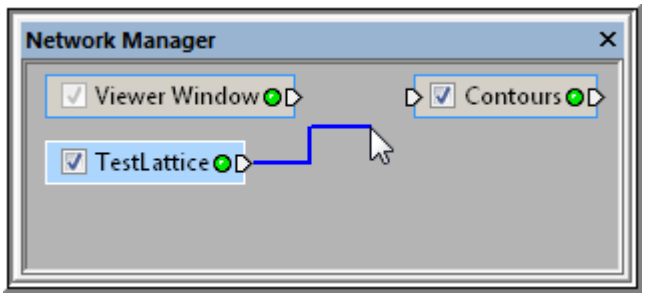
*Additional modules can be added to the **Network Manager** without being connected.*

Right-click on the *TestLattice* module and choose **Connect Output Lattice**.



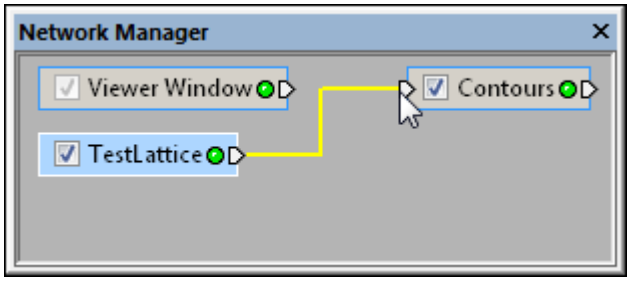
*Choose the **Network | Connect Output Lattice** command to connect the Contours module to the TestLattice module.*

The blue connection line appears in the **Network Manager**.



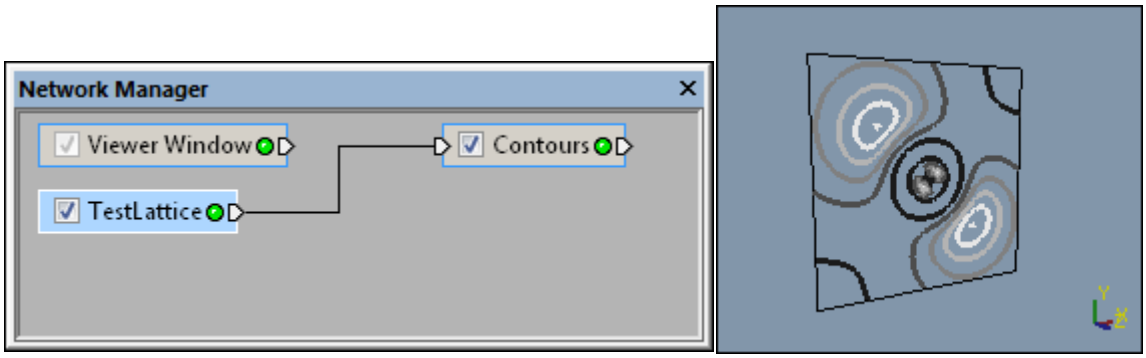
*When the **Network | Connect Output Lattice** command is selected, a blue connection line appears.*

Move the cursor to the *Contours* module and the blue line changes to yellow.



The connection line changes to yellow when it is over a module that can be connected to the selected module.

When the connection line changes to yellow, left-click to make the connection. The connection line becomes a black line and the contour map displaying the TestLattice data appears in the **Viewer** window.



*The contour map is displayed in the **Viewer** window when the Contours module is connected to the TestLattice module.*

To disconnect the two modules, right-click on the *Contours* module and choose the **Connect Input Lattice (TestLattice)** item.

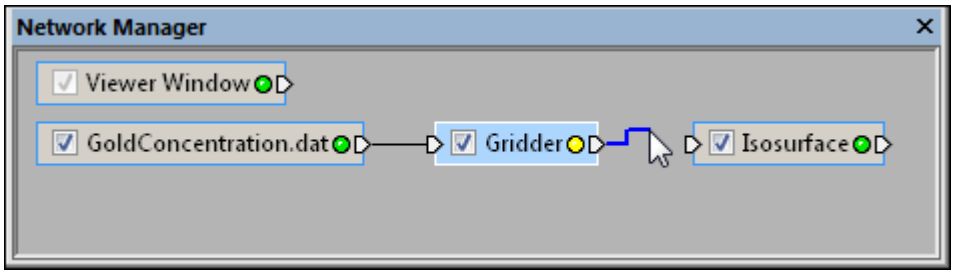
A blue line appears between the *Contours* module and the cursor. Move the cursor over the *TestLattice* module and the connection line turns yellow. Left-click on the *TestLattice* module in the **Network Manager** to break the connection.

Connecting Modules

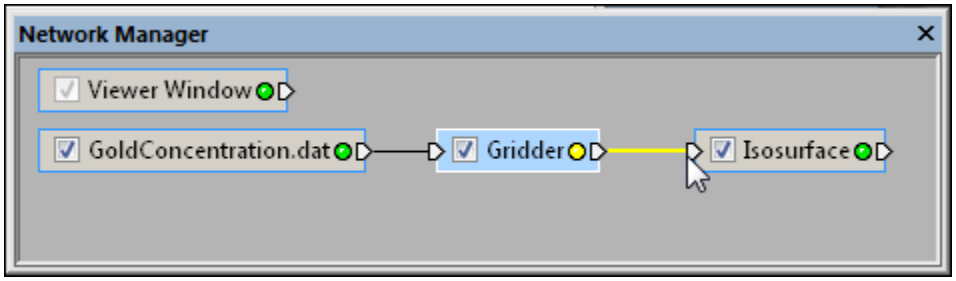
Module connections display the flow of data from one module to another.

Connect Modules

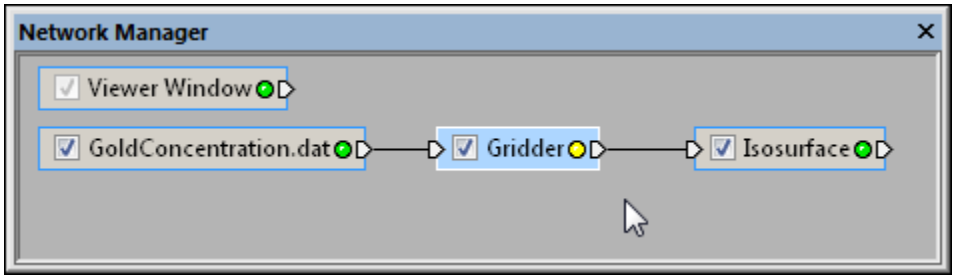
Connect a module to other compatible modules by dragging a connection line from a connection pad ▷ on one module to another. The line is initially blue, but turns yellow when the cursor touches a compatible module. Release the mouse button to complete the connection. The line turns black when the connection is complete.



An unconnected line appears blue.
In this example, the line between the Gridder and Isosurface modules is not yet connected, but has been dragged from the Gridder node.



A connected (but unreleased) line appears yellow.
In this example, the line between the Gridder and Isosurface modules is connecting the two modules, but the mouse button has not been released.



A connected (and released) line appears black.
In this example, the line between the GoldConcentration.dat, the Gridder, and the Isosurface modules are all connected.

Alternatively, you can click on a connection pad, release the mouse button, and move the mouse without dragging. A blue connection line is drawn. The connection line turns yellow when the cursor is over a compatible module. Click the mouse button a second time to connect the two modules.

Another method to connect modules is to select one module in the **Network Manager**. Choose the **Network | Connect** command to connect either an input or output module to the selected module.

Move the mouse until the connection line turns yellow. Click on the module and the two modules are connected.

Disconnect Modules

To disconnect a module from the network, follow the directions above to make the same connection a second time. The disconnected module remains in the **Network Manager** and retains all other connections.

Arranging Modules

Modules can be rearranged in the **Network Manager**. You may need to move modules to see some connections clearly. Click on a module and drag it to a new location to move it or use the ARROW keys for fine adjustment.

Move an Existing Connection

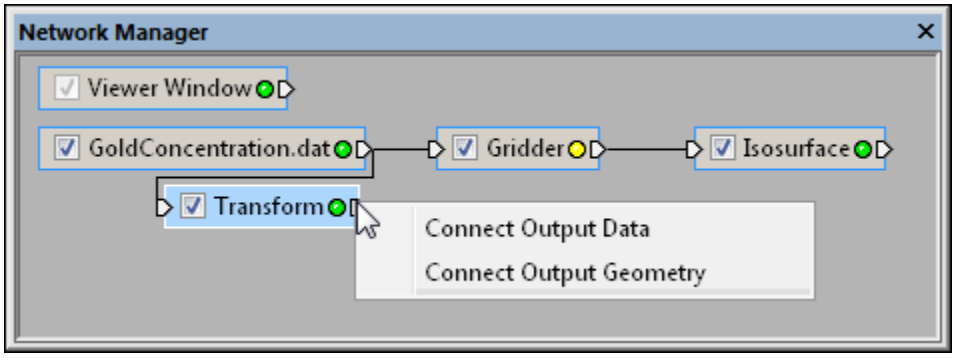
Move an existing connection by left-clicking on a connection line, holding down the left mouse button, and dragging the line to a new connection pad.

Cancel In-Progress Connection

Press the ESC key or click in an empty portion of the **Network Manager** to cancel a connection in progress.

Multiple Input or Output Connections

Some modules accept more than one input connection or provide more than one output connection. In these cases, a context menu is displayed when you click on the connection pad. Select the connection you want and proceed as previously described.



*The Transform module provides output connections for data and geometry. Select **Connect Output Data** to connect to the Isosurface module.*

Tips on Working with Modules

Modules can be connected and edited to see the effect of different parameters on the data. For example, a point set can be loaded, connected to a *Gridder* module, and displayed as an *Isosurface* module. You can experiment with the gridding parameters and immediately see the new output in the isosurface. The *Isosurface* module automatically recognizes that the lattice has changed and updates itself.

Multiple modules can be connected to a single module output connection to stack effects in the **Viewer** window. For example, *Contours*, *Isosurface*, and *OrthoImage* modules could all be connected to a single lattice module. The output from all three modules is correctly combined in the scene.

Change to the module location, visibility, and connection status can be undone using the **Edit | Undo** command.

The style of the connection lines can be customized using the **Tools | Options** command.

Mismatch Error

When attaching a module that is not compatible, the **Voxler Error** dialog appears.



*The **Voxler Error** dialog appears when a non compatible module is attached to another module.*

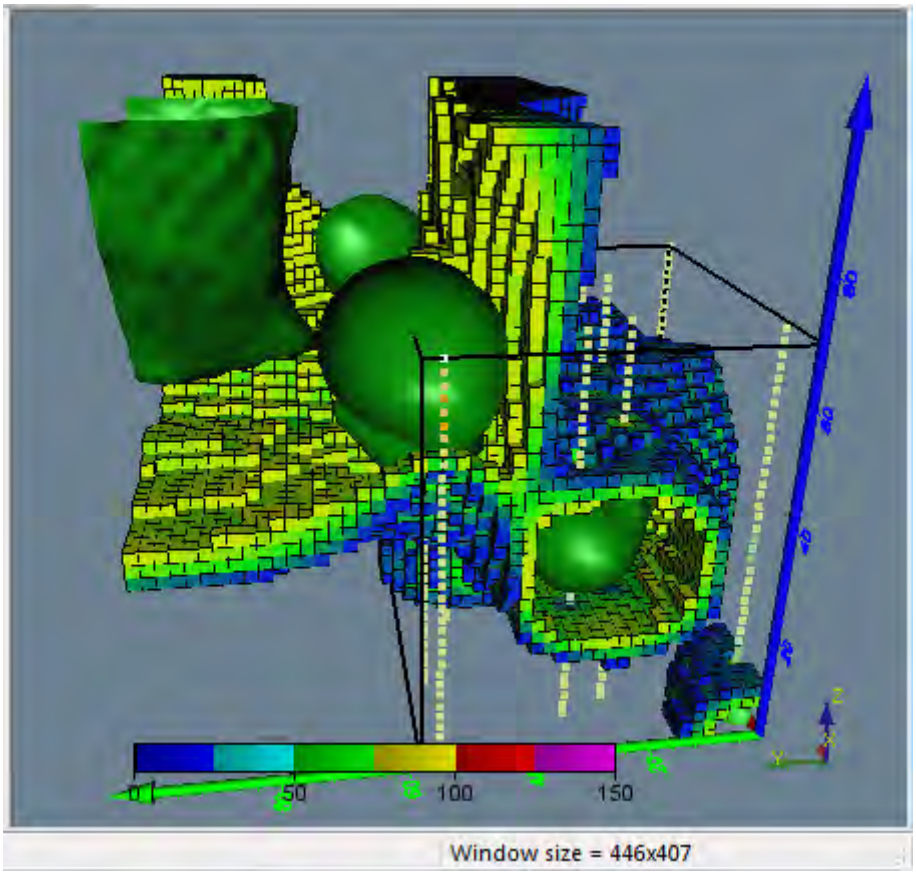
Click the *OK* button to remove the newly added incompatible module from the network.

The Viewer Window

The **Viewer** window displays the three-dimensional graphical output produced by the visualization network. All visible items in the **Network Manager** are displayed in the **Viewer** window.

The **Viewer** window behavior is controlled by several settings from the toolbars or the View menu. A virtual camera is used to view, rotate, zoom, and pan the scene. A world axis triad, located in the lower right corner, displays the global coordinate system orientation.

The image output of the currently selected module displays in the center of the window.



*The output geometry is displayed in the **Viewer** window.*
The world axis triad is displayed in the lower right corner.

Camera

In **Voxler**, the camera is a metaphor for how we view the scene in the **Viewer** window. A scene is the collection of all graphical output. The camera is a virtual device used to view the scene. The camera has a point of view that is controlled by panning, zooming, rotating, and lighting. The camera position and the center target of the scene can be changed with the **View | Camera Properties** command.

Home Position

The camera remembers a home position and orientation. Click the button on the toolbar, press the HOME key on the keyboard, or click the **View | Home** command to return the camera to its last stored home position.

Set a new home position by holding down the CTRL key and clicking the  button on the toolbar, by choosing the **View | Set Home** command, or by right-clicking in the **Viewer** window and selecting **Set Home** from the context menu. The home position is automatically set when a large change in the **Viewer** window occurs due to a significant change in network output when the *Enable AutoZoom* command is checked on the **General** page of the **Options** dialog accessed with the **Tools | Options** command.

Projection

The camera can display graphics in perspective or orthographic projections. Specify the projection with the **View | Projection** command. The projection affects how the three-dimensional scene is drawn in the Viewer window.

The most distinguishing characteristic of perspective projection is foreshortening: the farther an object is from the camera, the smaller it appears in the final image. Perspective projection emulates the human eye so scenes appear more realistic or lifelike—larger when viewed closely, smaller when viewed from a distance.

Orthographic projection produces a parallel projection with no distortion for distance. As a result, it is sometimes difficult to determine how far an object is from you when viewing it in orthographic projection. This view is useful, however, when you need to measure distances or angles, or exactly align objects in three-dimensional space.

Headlight

The **Viewer** window includes a built-in headlight at the camera position pointing in the same direction as the camera. This allows all geometry to be seen even if no explicit lights have been added to the network. The headlight is turned on or off with the **View | Headlight** command. If the headlight is turned off, the scene is dark unless one or more explicit *Light* modules are added.

Draw Style

Choose the method that overlapping objects are drawn in the **Viewer** window with the **View | Still Draw Style** options. These commands are useful if a particular display mode takes too long to render and you want to temporarily display graphics in a faster drawing format such as *Wireframe*, *Low Resolution*, or *Bounding Box*. By choosing a faster drawing format, the scene will redraw quicker, but some elements may appear incorrect.

A similar command, **View | Animating Draw Style**, applies to the graphics when the camera position is being changed, such as when the graphic is spinning or while changing the zoom level. If a faster drawing format, such as *Wireframe* or *Bounding Box* is used, the rotating objects may appear incorrect.

Transparency Mode

There are several transparency modes available through **View | Transparency Type**. Different modes work better for various types of geometry. Some modes provide faster rendering while others give better quality. Occasionally, you will need to experiment with transparency modes to find the best display.

If you find that your transparent graphics are partially opaque at certain orientations, try selecting **View | Transparency Type | Sorted object, sorted triangle blend** or **View | Transparency**

Type | Sorted Object Blend. These methods provide a more accurate transparency mode, but are also slower and more memory-intensive than others.

If you find that you are running out of memory or rendering is taking a very long time, choose the **View | Transparency Type | Blend** command. This method usually provides a reasonable tradeoff between accuracy and speed.

See *Transparency Type* for advantages and disadvantages of each transparency type.

AutoZoom

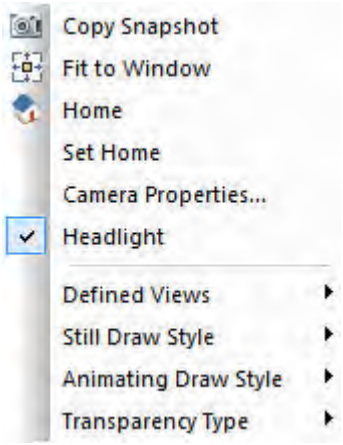
Voxler attempts to automatically keep the zoom to the full extents of the geometry in the **Viewer** window when it detects a large change to the extents of the geometry. **AutoZoom** does not occur when geometry output is turned on or off using the check box on the module icon in the **Network Manager**. The *Enable AutoZoom* command can be turned on or off in the **Tools | Options** dialog.

Customize Mouse Buttons

Choose the **Tools | Options** command to open the **Options** dialog. The **Mouse** page allows you to customize the mouse button assignments for the pan, zoom, rotation, and context menu control.

Common Commands

Right-click in the **Viewer** window to display a context menu of common commands, including **Copy Snapshot**, **Fit to Window**, **Home**, **Set Home**, **Camera Properties**, and **Headlight**. You may also choose a **Defined View**, **Still** or **Animating Draw Style**, or **Transparency Type**.



*Right-click in the **Viewer** window to change viewing options.*

Trackball

The **Viewer** window is in trackball mode by default. If the **Viewer** window is not in trackball mode, choose the **View | Trackball** command.

Trackball mode allows you to click in the **Viewer** window and rotate or spin the graphics with a virtual trackball. The trackball can be thought of as a transparent sphere covering the entire scene. To rotate a graphic, click in the **Viewer** window and drag the mouse. To spin the graphic, click in the **Viewer** window, drag the mouse, and release the mouse button while dragging. To stop the

spin, click anywhere in the **Viewer** window. Record the spinning graphic as an .AVI file with the **Actions | Capture Video** command.

For convenience, the following modifiers are allowed while in trackball mode:

- Hold down SHIFT while dragging to pan the camera
- Hold down CTRL while dragging up or down to zoom in or out

Rotate Object

To rotate objects in the **Viewer** window, left-click, hold the mouse button down, and drag the mouse, release the button to spin the image.

Stop Rotation

In a rotating window, hold the mouse stationary and click to stop the object from spinning.

Prevent Rotation

To prevent rotation when dragging the mouse, hold the mouse stationary prior to releasing the left mouse button.

Record Rotation

To record the rotation in the **Viewer** window, use the **Actions | Capture Video** command.

Pan Window

The **Viewer** window can be panned. This allows you to move the graphics in the **Viewer** window without changing the level of magnification or rotation. This is useful when you would like to look at a different portion of the geometry at the current scale. To use this feature, choose the **View | Pan** command or click the  button. Hold down the left mouse button and move the cursor around the window to pan the scene.

Fit to Window

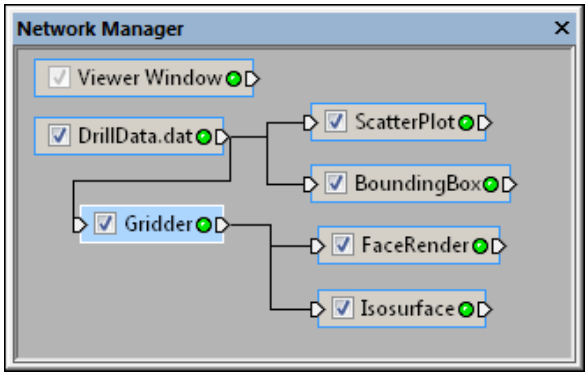
If the center of rotation for the objects in the **Viewer** window is changed and you need to reset the rotation, choose the **View | Fit to Window** or the  button to reset the rotation center to the center of the objects in the **Viewer** window. Occasionally, the geometry is no longer visible in the **Viewer** window, perhaps because new geometry has been added or the existing coordinates were modified. The **View | Fit to Window** allows you to zoom in to fit all the geometry into the **Viewer** window. After the **Fit to Window** command is used, the camera is moved until the scene fills the window extents.

Zoom

The **Zoom Realtime** command zooms in and out as the mouse is dragged up and down or when the mouse wheel is moved. To use this feature, choose the **View | Zoom Realtime** command. Hold down the left mouse button and then drag it up or down in the **Viewer** window to zoom in or out. As you drag the mouse up, the screen is zoomed in. As you drag the mouse down, the screen is zoomed out. You can also use the mouse wheel to zoom in or out. Move the wheel away from you to zoom in and toward you to zoom out.

The Property Manager

The **Property Manager** displays the properties of the module currently selected in the **Network Manager**. A property is a setting or parameter used by the module to control its behavior. Each property is displayed in a list within the window with one property per line. Properties are split onto multiple tabs. Left-click on a module in the **Network Manager** to display its properties in the **Property Manager**.



*The Gridder module is selected in the **Network Manager**. The selected module is highlighted in blue.*



*The properties of the selected module are displayed in the **Property Manager**.*

Property Manager Tabs

The **Property Manager** displays module properties, which are settings or parameters used by a module to control its behavior. The window itself consists of several tabs with similar properties, on a single tab.

Title Bar

A title bar is displayed at the top of the window. It consists of the window name, the button, and the button. See the *Working With Voxler Windows* topic for more information about how to use these buttons.

Context-Sensitive Help

Click the ? button to obtain information about the module currently displayed in the **Property Manager**. The **Voxler** help window is displayed with the relevant help page.

Applying Module Property Changes with Auto Update and Update Now

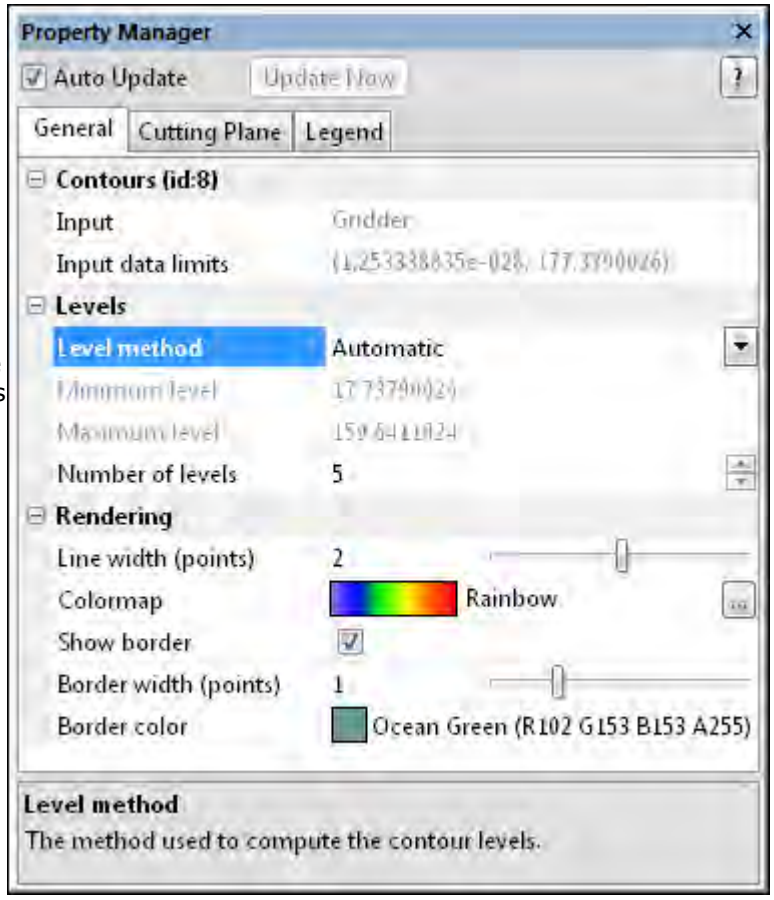
The *Auto Update* check box and *Update Now* button below the title bar allow you to choose whether to update the **Viewer** window automatically or manually. The *Auto Update* check box is checked, by

default. If a process takes a long time or you want to make a large number of changes to different modules that take a long time to redraw, then uncheck the *Auto Update* box. Make all changes to all modules. After all changes have been made, press the *Update Now* button to update the network manually. When the *Auto Update* box is checked, the module properties automatically update after you change an object. Uncheck the *Auto Update* box at the top of the **Property Manager** to disable this feature and make multiple changes without updating the **Viewer** window.

The 

button manually updates the network and any modified modules when *Auto Update* is unchecked. The *Update Now* button is enabled whenever there are pending changes to the network. Click the *Update Now* button to manually update the module properties in the **Viewer** window. Alternatively, choose the **Network | Update Network** command or press the F9 key on the keyboard to update the **Viewer** window with all changes.

Properties



The main component of the **Property Manager** is a list of properties used by the selected module, separated by tabs. This list has two columns: the left column contains the property name. The right column contains the controls used to change the property. Click on the property control in the right column to change the property's value. Drag the vertical line between the left and right column to adjust the column width. If a module's properties contain subsections, a ⊞ or ⊟ is located to the left of the name. Click on ⊞ or ⊟ to expand or collapse the list. For example, a *Contours* module contains three tabs: **General, Cutting Plane,** and **Legend**. The **General** tab contains three sections: *Contours*, *Levels*, and *Rendering*. Additional properties, such as the *Level method*, can be changed by clicking on the tab and opening these sections.

A short description of the selected property displays at the bottom of the window. If this area is not visible, click the **Tools | Options** command and check the box next to *Show property help* on the **General** page of the **Options** dialog.

Each property is displayed in a list within the window. Use the scroll wheel on the mouse to scroll through the list.

Expand and Collapse Features

Features with multiple options appear with a ⊞ or ⊟ to the left of the name. To expand a group, click on the ⊞ icon, select the item and press the + key on the numeric keypad, press the right arrow key on the keyboard, or double-click the item. To collapse the group, click on the ⊟ icon, press the - key on the numeric keypad; press the left arrow key; or double-click the item. For

example, the expanded *Rendering* section in the above image contains four options: *Line width (points)*, *Colormap*, *Show border*, *Border width (points)*, and *Border color*.

Resize

To change the size of either column of the **Property Manager**, position the cursor over the center vertical line to change the cursor into a two-headed arrow . Drag the arrow left or right to resize the viewable area on either size of the line.

To resize the help area at the bottom of the **Property Manager**, position the cursor just above the help area and drag the cursor  up or down to create more or less viewable help area.

Help

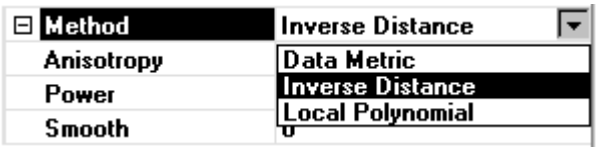
A simple help section is available at the bottom of the **Property Manager** for help on the selected property. The help area is turned on or off with the **Tools | Options** command. The horizontal dividing line at the top of this section can be dragged up or down. Click the  button at the top of the **Property Manager** to display more detailed information in the online help file about the module currently displayed in the **Property Manager**.

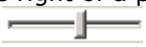
Property Manager Controls

Voxler provides several different types of controls in the right column of the **Property Manager** that allow you to customize the behavior of a selected module. These controls include buttons, check boxes, drop down lists, edit boxes, sliders, and spin buttons. The focus is automatically set to the object when it is clicked or selected via the keyboard TAB or arrow keys. The focus is indicated by a dotted rectangle on the control or by a change in color. Some property list items only provide information. These items are disabled (grayed) to indicate they cannot be changed. Occasionally, some properties may not be valid due to other selections or data restrictions. These options are disabled as well.

The following controls are available:

Button	Click the button with the mouse or press the SPACEBAR when the button has the focus to perform the action indicated by the button text. Click the  button to obtain more information about a property. The resulting action varies depending on the module, but can open a dialog for loading a file into an existing data input module, allow you to edit multi-line text, display a **Colormap Editor** or other actions.
Check Box	A check box toggles the state between one of two possible outcomes. Click the check box with the mouse or press the SPACEBAR when the box has the focus to change the state of the check box.
Color	The color control displays a sample and the name of the current color. If the color cannot be found in the predefined colors list, then the name takes the form "r, g, b" (the amounts of red, green, and blue, 0 to 255 each). To change the color, click the color to the right of the *Color* property to open the color palette.

Drop Down List	A drop down list of options displays when the corresponding property value is selected. To display the drop down list, click the value with the mouse or press the SPACEBAR when the control is selected. To select an item in the drop down list, click the item in the list or use the up or down arrow keys on your keyboard to highlight the item and press ENTER.  *An unselected list item with a text value does not display a drop down arrow.*  *A selected list item with a text value does display a drop down arrow.*  *When a list item with a text value is selected, click the drop down arrow to display the list of options.* Use the keyboard keys to perform the following actions: • The F4, SPACEBAR, or up and down arrow keys activate a drop down list for the selected property value. • The ESC key closes a drop down list. • The up, down, left, and right arrow keys scroll through a drop down list. • The HOME key moves focus to the first item in a list. • The END key moves focus to the last item in a list. • The PAGE UP or PAGE DOWN key moves up or down several items at a time in a drop down list. • ENTER selects an item from a drop down list. • Any letter on the keyboard can be used to select the first list entry starting with that letter. Press the letter again to move to the next item beginning with that letter.
Edit Boxes	Edit boxes allow the input of text or numeric values. Edit boxes are commonly linked to an associated slider  or spin button  (see below) when entering a number, or may include the  button for multiline text or other functions. Drag the borders of the **Multiline Text** dialog to resize it. Press the ESC key to cancel your changes and restore the original text.

Slider	A slider appears to the right of a property that requires a numeric entry. Drag the slider  left or right to decrease or increase the number. Click anywhere on the slider to immediately move the slider to that position. When the slider has the focus, the arrow, PAGE UP, PAGE DOWN, HOME, and END keys can be used to change the value. The slider can be very short or not visible if the column width is small. Increase the column width by dragging the left or right edge of the column. The slider is set to reasonable limits, but **Voxler** accepts a larger range of values from the edit field for some properties. In such a case, enter the number directly in the edit field.
Spin Button	A spin button  may appear to the right of a property that requires a number. Use the up and down arrows to increase or decrease the value in the edit control. When the value has reached the maximum or minimum allowed value, it will either stop changing or wrap around to the beginning or end of the range. The spin button is set to reasonable limits, but there is sometimes a need to exceed those limits. If necessary, enter the desired value directly into the edit boxes.
Static (Grayed-Out) Text	Static text is informational and cannot be edited, as indicated by the grayed-out color of the text.

Keyboard Commands

Use the following keys when working in the **Properties** window when the property name is selected:

- The UP ARROW key selects the previous property.
- The DOWN ARROW key selects the next property.
- The LEFT ARROW key selects the previous property or collapses the list of a parent property.
- The RIGHT ARROW key selects the next property or expands the list of a parent property.
- The PAGE UP key moves the selection up by the height of the window, scrolling if necessary.
- The PAGE DOWN key moves the selection down by the height of the window, scrolling if necessary.
- The HOME key selects the first property.
- The END key selects the last property.
- The TAB key selects the next property that accepts keyboard input (similar to a dialog). It skips collapsed items as indicated by the (+) button to the left of the item. Use the arrow keys to navigate to a collapsed item.
- SHIFT+TAB selects the previous property that accepts keyboard input (similar to a dialog).

New Features

This is an overview of some of **Voxler 4's** new features.

Graphical Module Features

- Additional predefined colormaps
- *HeightField* module improvements, including:
    - Overlay a vector data file.
    - **Surfer** blanking value support.
- *ScatterPlot* module improvements, including:
    - Use 3D Cone, 3D Cube, 3D Cylinder, and 3D Sphere symbols.
    - Specify the symbol color with RGB color columns in the data file.
    - Points can be classed/binned.
    - Control label density.
- Export *Isosurfaces* to XYZC data files.
- *FaceRender*: pan through volume with a slice that is a single voxel thick.

User Friendly

- Streamlined data import process.
- Open multiple projects and worksheets in one **Voxler** application window.
- The *Input component* property in the **Property Manager** for graphical, computational, and data source modules presents a list of available components.
- Header row information is displayed in the **Property Manager** to make data column selection easier.
- More *User interface style* appearance options.
- New Welcome to Voxler dialog.

Data Features

- Import tabular data files to *Data Source* modules.
- Specify which logs are included in the *WellRender* module in the *Data Source* properties.
- Create and edit data files in the worksheet window.
- Save data files in multiple formats: BLN, BNA, CSV, DAT, SLK, TXT, XLS, XLSX
- Hot editing: changes made in the worksheet window to linked worksheets are automatically visible in the viewer window.
- Assign and convert coordinate systems in the worksheet.
- Hide or show wells from the **Property Manager Wells** page for the *WellData* module.
- Unicode character support.
- Import LASer LiDAR Binary data files.

Import and Export Improvements

- Import and export 3D DXF files that include 3D entities.
- Export *HeightFields*, *Isosurfaces*, and *Contours* to 3D DXF files.
- Export PNG, TIF, and GIF image files with application background transparency.
- Import Google Earth KML/KMZ files.

- Improved line width handling when importing AutoCAD DXF files.
- Import 3D BLN files.
- Import NASA SRTM Grid Data HGT files.
- Import Adobe Acrobat PDF Vector and Raster files.
- Import and export JPEG-2000 image files.
- Import ERDAS Imagine IMG image and grid files.
- Import compressed/zipped SHP Esri Shapefiles.
- Import TIF image files with YCbCr color encoding.
- Import macro-enabled Excel XLSM files.
- Import LASer LiDAR Binary data files.
- Import compressed/zipped Raster and Vector DDF files.
- Import GRIB Weather Data Grid files.
- Import and export NC NetCDF Lattice files.
- Import ECW and SID files without reduced resolution.
- Extract a subregion of ECW and SID files for import.
- Import PLY files with color and material properties.
- Import compressed Geosoft .GRD .GGF grid files.

Automation Updates

- Pass command line arguments to a script.
- Specify multiple component, label, or log columns for a Data Source module.

Chapter 2 - Tutorial

Tutorial Introduction

The tutorial is designed to introduce you to some of **Voxler's** basic features. After you have completed the tutorial, you should be able to begin to use **Voxler** with your own data. We strongly encourage completion of the tutorial before proceeding with **Voxler**. The lessons should be completed in order; however, they do not need to be completed in one session. The tutorial should take approximately one hour to complete.

In this tutorial, you will import data, link the data to modules, change properties, and save information. To open the tutorial, choose the **Help | Tutorial** command.

The following is an overview of the lessons included in the tutorial.

Starting Voxler shows how to start **Voxler** or open a new visualization network.

Lesson 1 - *Loading Data* shows how to load data.

Lesson 2 - *Creating Graphics Output Modules* shows how to create a scatter plot of the data and how to add a bounding box.

Lesson 3 - *Changing Properties* shows how to change module properties and rotate the view.

Lesson 4 - *Editing Linked Data in the Worksheet* shows how to view a *Data Source* module's data in the worksheet window and how to edit the data.

Lesson 5 - *Using Computational Modules* shows how to create a uniform lattice from the point set, display the lattice as an isosurface, and filter data.

Lesson 6 - *Connecting Multiple Modules* shows how to connect multiple output modules to a single input module.

Lesson 7 - *Saving Information* shows how to save graphics, data, and a network.

Lesson 8 - *Importing and Displaying Wells* shows how to import well data, display the well data as a wellrender module, and modify the well render module properties.

The lessons should be completed in order; however, they do not need to be completed in one session.

Advanced Tutorials

Advanced Tutorial Lessons are available to demonstration additional features of **Voxler**. These tutorials are intended for beginner to advanced users.

A Note About the Documentation

Various font styles are used throughout the **Voxler** quick start guide and online help file. **Bold** text indicates windows, menu commands, dialog names, and page names. *Italic* text indicates items within a dialog or window such as modules, group box names, options, and field names. For example, the **Save Geometry Data** dialog contains a *Save in* drop down list. Bold and italic text may occasionally be used for emphasis.

Also, menu commands appear as **View | Fit to Window**. This means "click the **View** menu at the top of the program window, then click **Fit to Window** within the **View** menu list."

Using the Tutorial with the Demo Version

Some **Voxler** features are disabled in the demo version, which means that some steps in the tutorial cannot be completed by users running the demo version. This is noted in the tutorial and users are prompted to proceed to the next step.

Starting Voxler

To begin a **Voxler** session:

1. Click on the Windows Start button.
2. Navigate to **Programs | Golden Software Voxler 4** (in Windows XP) or **All Programs | Golden Software Voxler 4** (in Windows Vista and Windows 7) and click *Voxler 4*.
3. **Voxler 4** starts with a new blank **Viewer** window. The first time you open **Voxler** you are prompted for a serial number. Your serial number is located on the CD cover or in the email download directions, depending on how you purchased **Voxler**.

If **Voxler** is already open, select the **File | New** command or click the  button to open an empty visualization network before continuing with the tutorial.

Lesson 1 - Loading Data

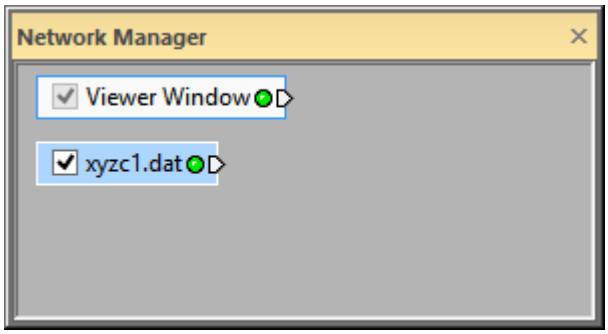
Many data types are used in **Voxler**. For a detailed list of supported formats, refer to the *File Format Chart*. The three main categories of data are point sets, lattices, and geometry. Point sets contain X, Y, and Z data along with optional components and label columns. Lattices can be one-, two-, or three-dimensional data arrays. Geometry files include lines, triangles, and other shapes.

You can load data in **Voxler** by:

- clicking the **File | Import** command, or
- right-clicking in the **Network Manager** and selecting **Import**,
- double-clicking the *Import* item in the **Module Manager**, or
- dragging a file from the computer to the **Network Manager** window. Release the file on the **Network Manager**.

To load a data file:

1. Click the **File | Import** command. The **Import** dialog opens.
2. In the **Import** dialog, change the *Look in* field to the *Samples* directory, located inside the main **Voxler** folder. By default, this folder is located at C:\Program Files\Golden Software\Voxler 4\Samples. Click on the *xyzc1.dat* file and click the *Open* button.
3. The file is loaded into the project, and the **Network Manager** has a new *Data Source* module titled *xyzc1.dat*.



*After a data file is loaded, it appears as a module in the **Network Manager**.*

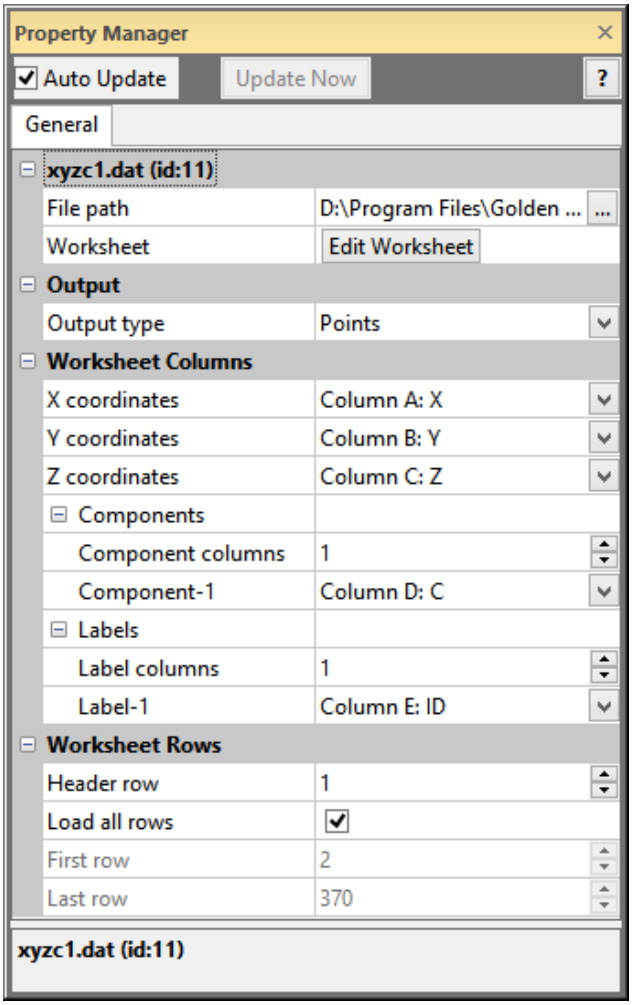
4. We must make sure the data file columns are correctly defined before using the data file to add graphics output or computational modules. Click on the *xyzc1.dat* module to view its properties in the **Property Manager**.
5. Click the ⊞ next to *Output* if the section is not already open and make sure the *Output type* is set to *Points*. If it is set to *Wells*, click *Wells* and select *Points* from the list.
6. Click the ⊞ next to *Worksheet Columns* to view the column specification properties. Next verify the data columns are specified correctly.

- X coordinates is set to Column A: X
- Y coordinates is set to Column B: Y
- Z coordinates is set to Column C: Z

7. Click the ⊞ next to *Components* and *Labels* to view the two sections, if they are not already open. Verify the following settings are correct:

- Component columns is set to 1
- Component-1 is set to Column D: C
- Label columns is set to 1
- Label-1 is set to Column E: ID

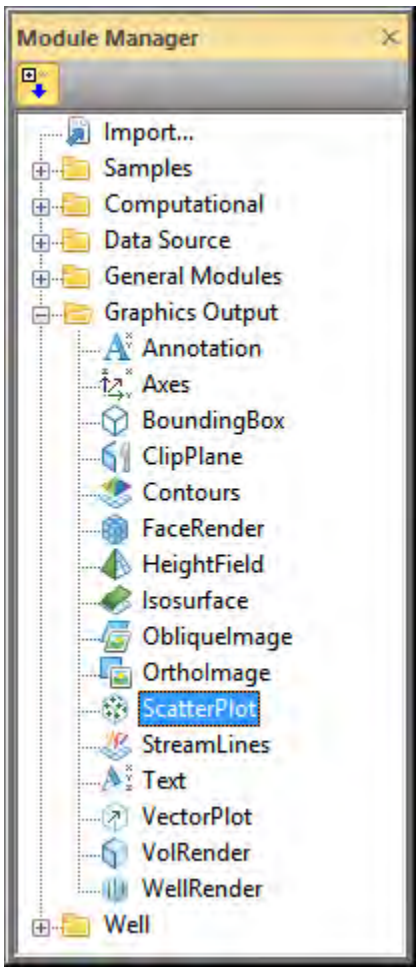
Now the data module is ready to be connected to a graphics output module.



The columns for the xyzc1.dat module are specified as above.

Lesson 2 - Creating Graphics Output Modules

Once the data are loaded, a data module appears in the **Network Manager**. The **Viewer** window is blank because no graphical modules have been added to the data module. In this lesson, we add a graphic.



Select ScatterPlot in the Graphics Output folder to create a scatter plot of the data.

2.1 - Creating a Scatter Plot - Tutorial

A scatter plot is a model of point data within a volume of space, optionally with colors representing data values.

To create a scatter plot:

1. Click on the *xyzc1.dat* module in the **Network Manager** to select it. The selected module is highlighted.
2. In the **Module Manager**, double-click on the *ScatterPlot* module name in the *Graphics Output* section of the list. Alternatively, right-click on *xyzc1.dat* in the **Network Manager** and select **Graphics Output | ScatterPlot** from the context menu.

A scatter plot is displayed in the **Viewer** window and the *ScatterPlot* module appears connected to the *xyzc1.dat* module in the **Network Manager**.

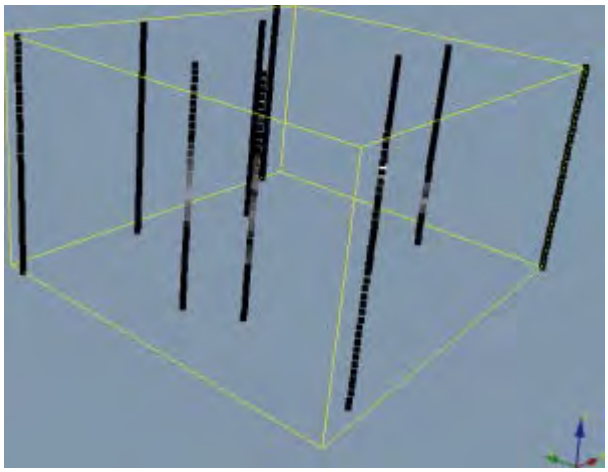
In the **Module Manager**, only the modules that can be used with the data type are listed if the *Show All Modules* button is not selected. The button is selected if it looks like this  and not selected when it looks like this . In this example, we used point data, so the main graphics outputs are axes, bounding box, scatter plots, and vector plots. Other graphics, such as isosurfaces, need lattices as inputs so they are not listed when the button is not selected.

2.2 - Adding a Bounding Box - Tutorial

Next we will add a bounding box around the input data. A bounding box draws a three-dimensional box around the input data extents.

To draw a bounding box:

1. In the **Network Manager**, click the *xyzc1.dat* module.
2. In the **Module Manager**, double-click *BoundingBox* under *Graphics Output.*

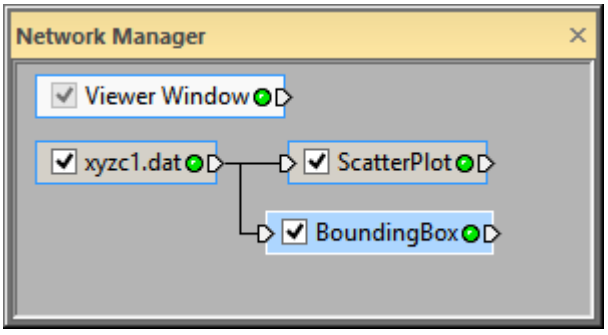


Your scatter plot should look similar to this after you add the bounding box.

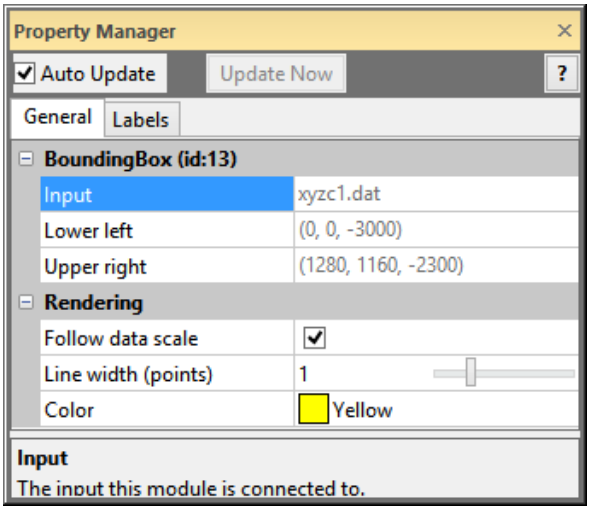
Alternatively, you could have right-clicked the *xyzc1.dat* module in the **Network Manager** and selected **Graphics Output | BoundingBox** from the context menu.

Lesson 3 - Changing Properties

Once modules have been created, their properties can be changed in the **Property Manager**. Click on a module in the **Network Manager** to select it and display the module's properties in the **Property Manager**. Note that some items in a module's property list are informational only and cannot be changed. These items appear gray in the list.



*Select a module in the **Network Manager**.*



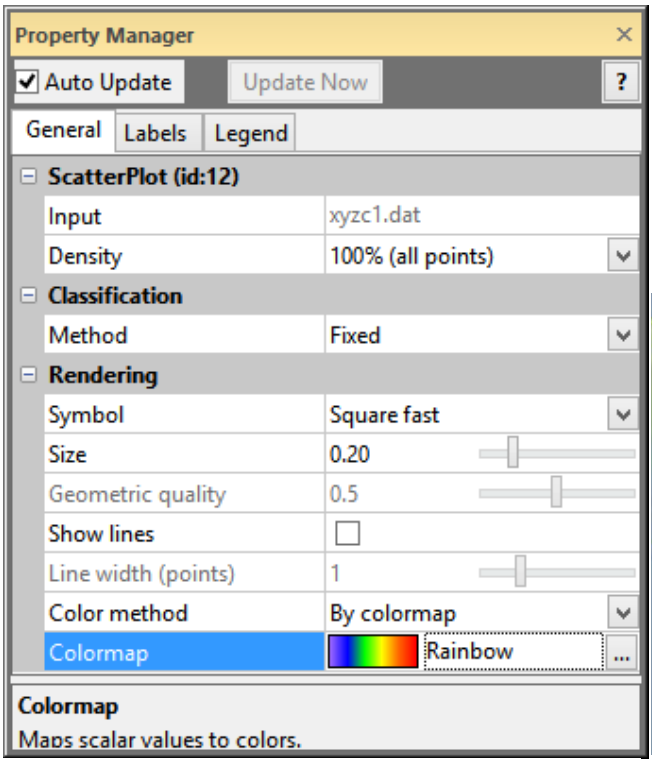
*The selected module's properties are displayed in the **Property Manager**.*

3.1 - Changing Symbol Colors - Tutorial

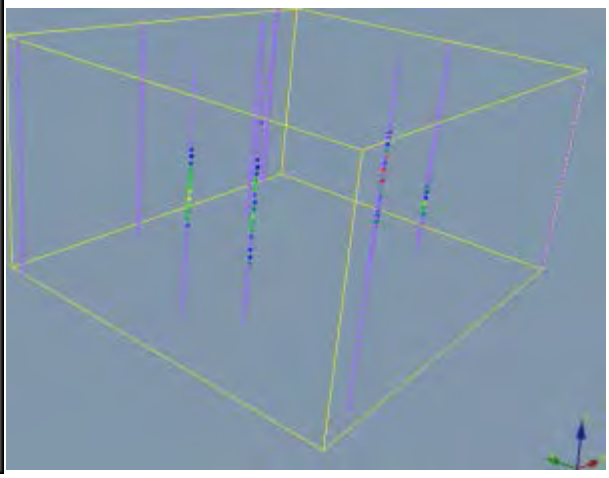
One property that can be changed for a scatter plot is the symbol color.

To change the symbol colors:

1. Click on the *ScatterPlot* module in the **Network Manager**. The *ScatterPlot* module properties open in the **Property Manager**.
2. In the **Property Manager**, click on the **General** tab.
3. Scroll down to the *Colormap* option. Click the sample color spectrum (*GrayScale*) to the left of the ... button to open the option list. Click *Rainbow* to change the colors. The colors are mapped to the data variable C, as selected when the data were loaded in the *Loading Data* lesson.



Click the color spectrum next to Colormap *and select* Rainbow *from the list to change the symbol colors.*



The scatter plot is updated with the colors in the selected Colormap.

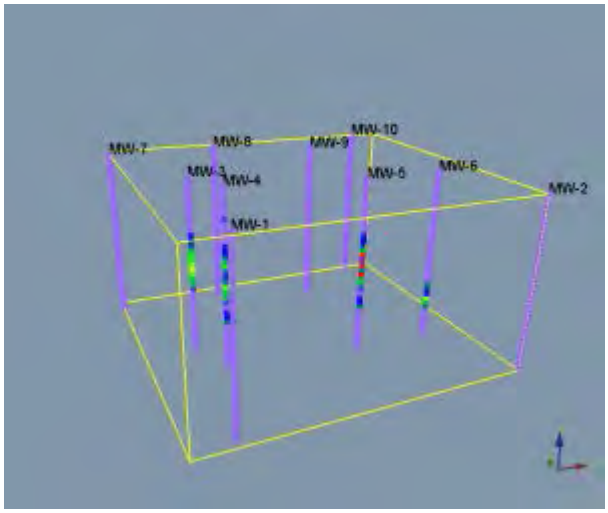
3.2 - Displaying Labels - Tutorial

Labels can be displayed on the scatter plot from the X, Y, Z, XYZ, or specified label columns. Let's add the label from the label column so that the data can be identified.

To add labels:

1. Click on the *ScatterPlot* module in the **Network Manager**.
2. In the **Property Manager**, click on the **Labels** tab.
3. Check the box next to *Show labels* to turn on the display of labels for the module.

4. Change the *Label field* to the desired column. In this case, let's select *Column E: ID*. The labels are added next to the points that contain information in the label column.
5. Click the ⊞ next to *Font* to open the font properties for the labels.
6. To increase the size of the labels, click and drag the  next to *Size (points)* until the desired size is shown in the **Viewer** window.



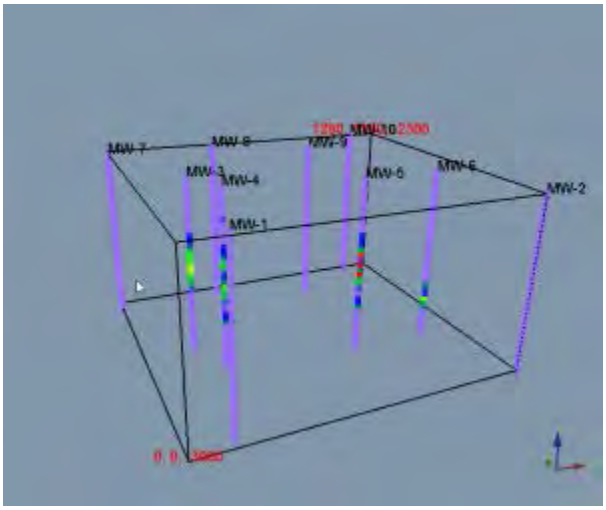
Labels are displayed next to scatter plot points.

3.3 - Changing the Bounding Box Properties - Tutorial

Changing the bounding box properties is similar to changing the symbol colors.

To change the bounding box line thickness and color:

1. Click on *BoundingBox* in the **Network Manager.**
2. In the **Property Manager**, click on the **General** tab.
3. Click *Yellow* next to the *Color* property. The color palette opens.
4. Click on the color black to change the bounding box color to black.
5. Highlight the value 1 next to *Line width (points)* and type 1.5. Press ENTER on the keyboard and the line thickness updates.



The bounding box properties can be changed using the settings in the ***Property Manager****.*

To add labels:

1. Click on the **Labels** tab.
2. Check the box next to *Show labels* to display labels at the maximum and minimum values for the bounding box.
3. To change the label color, click the black color box next to *Color* and select another color, such as red.

3.4 - Rotating Graphics - Tutorial

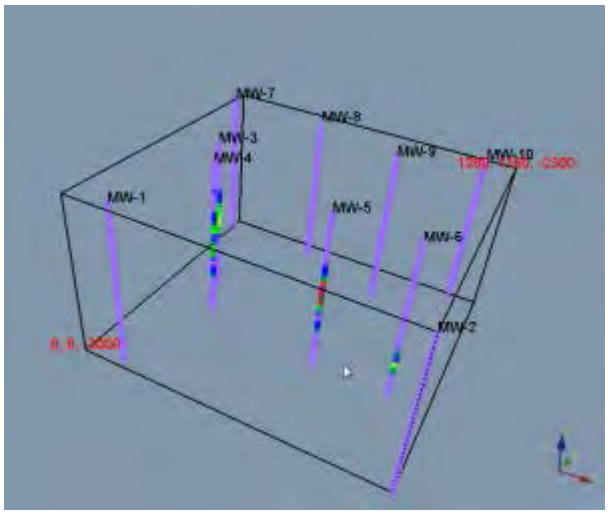
The **Viewer** window contents can be rotated and animated (spinning). Currently, we are viewing the scatter plot from the lower left side. We can rotate the scatter plot to see the symbols and labels more clearly.

The view is rotated by clicking on the **Viewer** window, holding down the left mouse button, and dragging the mouse. If you release the mouse button while the mouse is still moving, the **Viewer** window will enter spin mode.

Experiment with different rotations. If you spin the graphic, you can stop the spin by clicking anywhere in the **Viewer** window.

A world axis triad is displayed in the lower right corner of the Viewer window. The axis triad is informational only. The triad is a depiction of the X, Y, and Z directions that shows the Viewer window camera orientation. This is useful when rotating graphics to see how the graphics have been rotated in space.

The rotation of the **Viewer** window can be recorded with the **Actions | Capture Video** command.



This shows the scatter plot after it has been rotated.

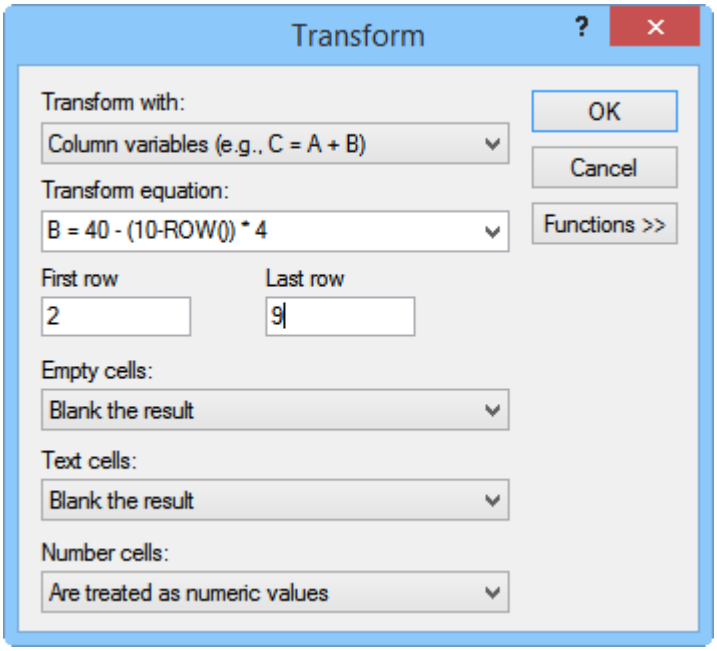
Lesson 4 - Editing Linked Data in the Worksheet

The data held in a *Data Source* module can be edited in the worksheet window. Changes made in the worksheet window are immediately visible in the **Viewer** window. In this lesson we will make a few changes in the worksheet and see how it affects the downstream modules.

To view the *xyz1.dat* module data in the worksheet:

1. Click on the *xyc1.dat* module in the Network Manager.
2. Click the *Edit Worksheet* button in the *Worksheet* field of the **Property Manager**. The module data is opened in a new worksheet window. Notice the document tab displays "Linked to:", the module name, and the module ID. This indicates the worksheet is displaying the module data and not the xyzc1.dat data file.

3. Click on cell E2 containing the "MW-1" label. The bold cell border indicates that cell E2 is now the active cell.
4. Type a new label name, for example type *BH-10* and press ENTER.
5. Click the **Data | Transform** command. The **Transform** dialog opens.
6. Change the Transform with setting to Column variables (e.g. C = A + B).
7. In the Transform equation field, type B = 40 - (10-ROW()) * 4.
8. Type *2* in the *First row* field and type *9* in the *Last row* field.
9. Click *OK* in the **Transform** dialog.
10. Now click on the *Voxler1* project tab located above the *Active Cell Location* box.



*A large number of data manipulations can be performed with the **Transform** command.*

Notice that the point formerly labeled "MW-1" is now labeled "BH-10" (or whatever you chose to name it). You can also see how changing the Y value for the first 8 points changed the scatterplot. Note that we changed the data for the *xyzc1.dat* module. The xyzc1.dat sample file is unchanged. To save a copy of changes made to linked data, click the **File | Save Copy As** command while viewing the data in the worksheet.

Lesson 5 - Using Computational Modules

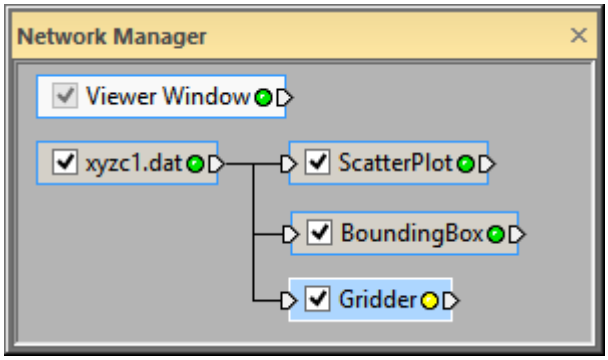
Computational modules use data inputs to grid point sets, filter data, merge data, and perform other tasks. Several visually interesting graphics output modules require lattices, including contours, height fields, isosurfaces, oblique images, orthogonal images, streamlines, and volrenders. The XYZC1.DAT file contains scattered point data. You can use the *Gridder* module to create a lattice from the scattered point data. After the point set is converted to a lattice, we can display it as an isosurface, volrender, or contours module. Filtering can be applied to see the effect on the network and graphics output modules.

5.1 - Gridding Data - Tutorial

To create a lattice from xyzc1.dat:

1. In the **Network Manager**, click the *xyzc1.dat* module.

2. In the **Module Manager**, double-click the *Gridder* module in the *Computational* section. The *Gridder* module is loaded into the network. Alternatively, you can right-click on the *xyzc1.dat* module in the **Network Manager** and select **Computational | Gridder**.
3. In the **Network Manager**, the *Gridder* module displays with a yellow indicator LED, indicating that additional input is required. In this case, we need to initiate gridding in the **Property Manager**. With the *Gridder* module selected in the **Network Manager**, click on the **General** tab in the **Property Manager**.
4. Click the *Begin Gridding* button in the **Property Manager** to begin the gridding process. The *Gridder* module indicator LED changes to green when the gridding is complete.



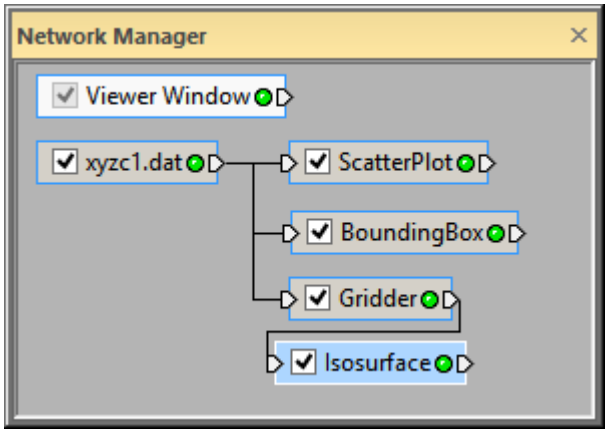
The indicator LED is a small circle to the right of the module name. The Gridder module indicator LED initially appears yellow, indicating you must make a properties change.

5.2 - Creating an Isosurface - Tutorial

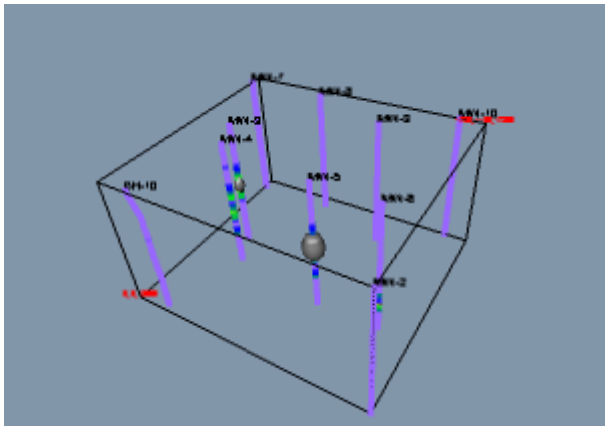
The *Gridder* module interpolated the scattered point data onto a uniform lattice. To display the lattice in the **Viewer** window, the *Gridder* module needs to be connected to a graphics output module. An isosurface, a surface of constant value in three dimensions, is one way a lattice can be displayed.

To create an isosurface:

1. Click the *Gridder* module in the **Network Manager** to select it.
2. Double-click the *Isosurface* module in the **Module Manager**. Alternatively, right-click on the *Gridder* module and select **Graphics Output | Isosurface**. The *Isosurface* module appears connected to the *Gridder* module in the **Network Manager** and an isosurface displays in the **Viewer** window.



Add an Isosurface module to the Gridder module.



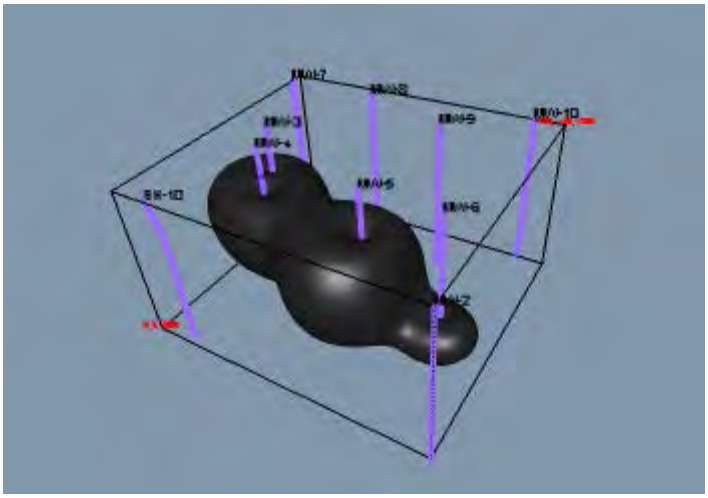
*An Isosurface with default properties is displayed in the **Viewer** window.*

5.3 - Changing the Isosurface Properties - Tutorial

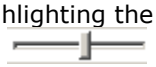
We can experiment with the isovalue (constant value) to change the isosurface appearance.

To change the isosurface properties:

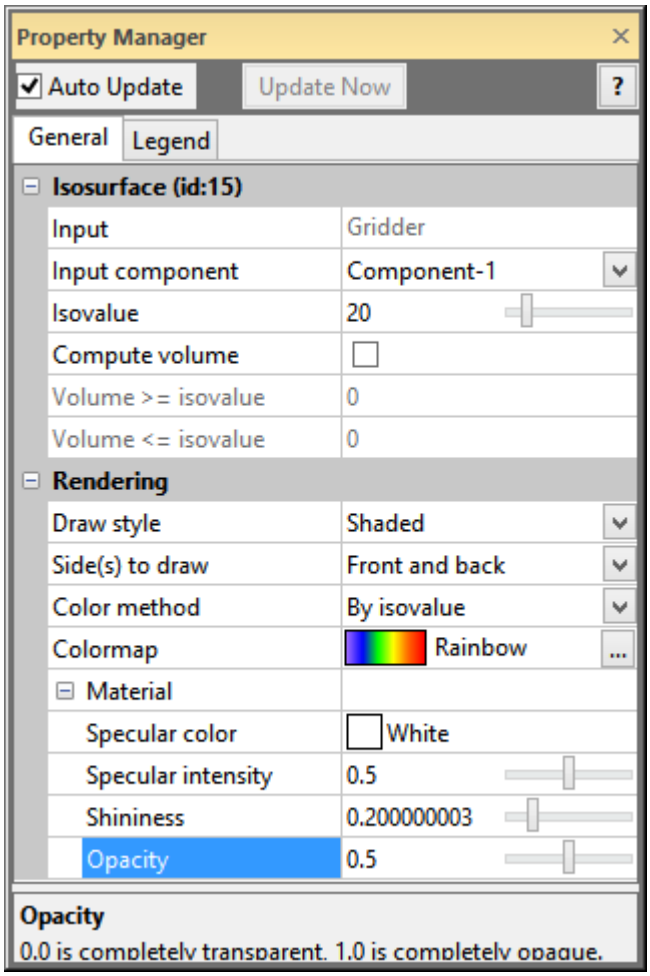
1. Click on the *Isosurface* module in the **Network Manager** to select it.
2. In the **Property Manager**, click on the **General** tab.
3. Change the value next to *Isovalue* by double-clicking on the default value, typing 20, and pressing ENTER on your keyboard. Alternatively, the  next to the number can be moved to change the *Isovalue*. A new isosurface is calculated and immediately displayed in the **Viewer** window.



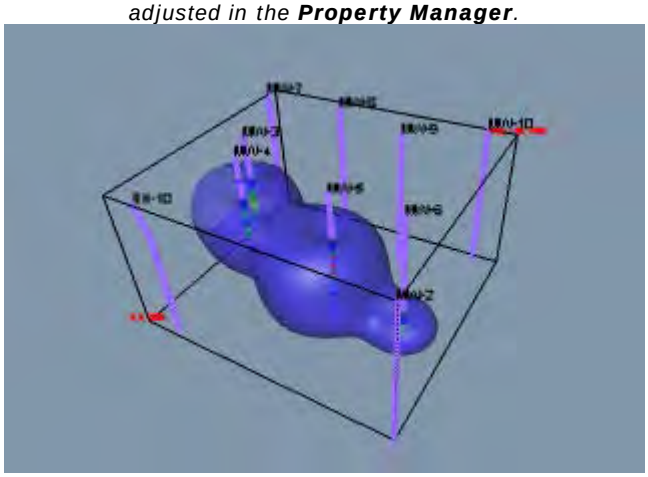
An Isosurface displays constant value in three dimensions.

3. In the **Property Manager**, click the *GrayScale* color spectrum next to *Colormap* and select *Rainbow*.
4. In the **Property Manager**, open the *Material* section by clicking the ⊞ next to *Material*.
5. Change the *Opacity* value by highlighting the existing value, typing 0.5, and pressing ENTER on your keyboard or moving the  until the value is 0.5. Changing the *Opacity* to a

lower value allows the isosurface to be partially transparent. The lower the value, the more transparent the isosurface.



*The Colormap and Opacity settings are adjusted in the **Property Manager**.*



An Isosurface with a custom Colormap and Opacity properties.

5.4 - A Note About Transparency - Tutorial

An *Opacity* value of 0.0 is fully transparent. An *Opacity* value of 1.0 is fully opaque. Transparency can be very time consuming to get absolutely correct. As such, **Voxler** contains several different algorithms that trade off speed against correctness. See the **View | Transparency Type** options if the transparency does not look correct for your particular data. Usually the **Sorted Object, Sorted Triangle Add** and **Sorted Object, Sorted Triangle Blend** methods result in good output but these methods are significantly slower than the other methods. A quick method that produces good results in many cases is **Blend** or **Delayed Blend**.

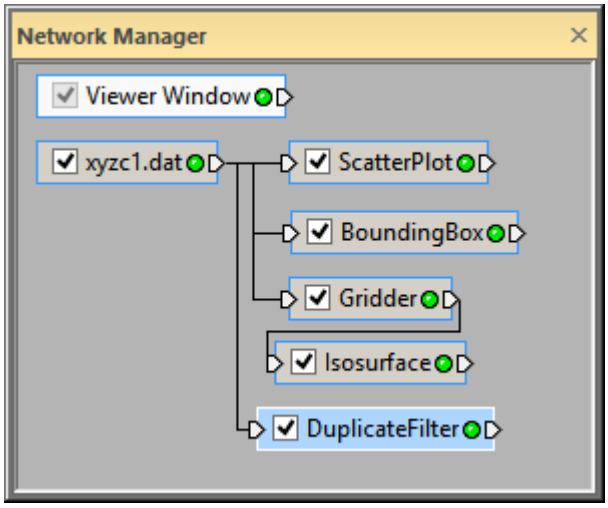
5.5 - Filtering Data - Tutorial

You can add computational modules between the data file module and the *Gridder* module to change the isosurface. There are many data filtering options in **Voxler**. Filtering modifies the data stream, which affects all downstream modules. Typically, the "downstream" modules are automatically changed when "upstream" modules are altered. The *Gridder* module is one exception due to the time required to grid the data.

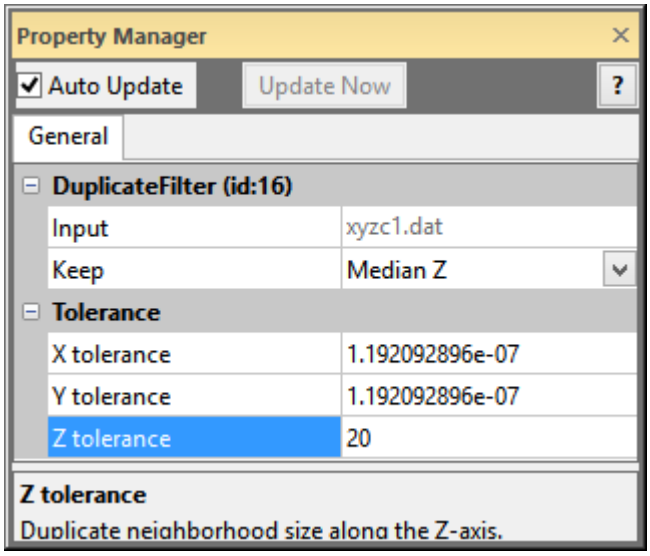
As an example of filtering our data, assume the data contains points that are very close together and we would like to combine these duplicate points into a single representative value.

To average these duplicate points:

1. Click the *xyzc1.dat* module in the **Network Manager**.
2. In the **Module Manager** *Computational* section, double-click the *DuplicateFilter* module to add it to the **Network Manager**. Alternatively, right-click on the *xyzc1.dat* module and select **Computational | DuplicateFilter**.
3. Click on the *DuplicateFilter* module in the **Network Manager** to select it.
4. In the **Property Manager**, change the *Keep* option to *Median Z*.
5. In the **Property Manager**, enter 20 for the *Z Tolerance*.



*Select the DuplicateFilter module in the **Network Manager**.*

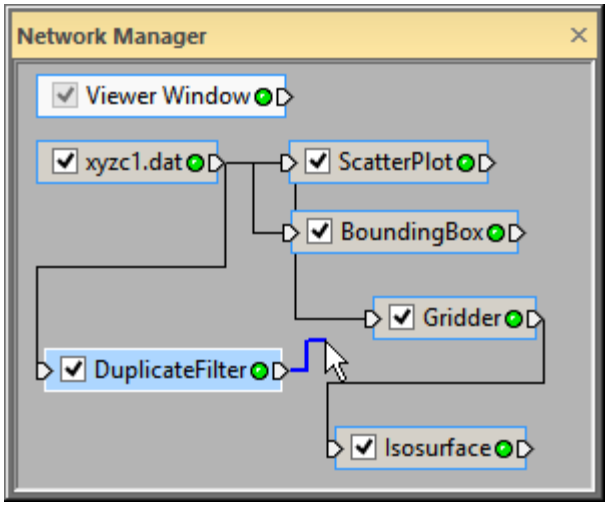


*Adjust the module properties in the **Property Manager**.*

Since there are no output modules currently connected to the *DuplicateFilter* module, there are no visible changes in the **Viewer** window. We can make changes by connecting the *DuplicateFilter* module to the *Gridder* module.

To connect the *DuplicateFilter* module:

1. First, move the *DuplicateFilter* module to the left side of the **Network Manager** so the connections are easily seen. Click on the *DuplicateFilter* module icon and drag it to the left side of the **Network Manager**.
2. Click on the output connection pad ▷ on the right side of the *DuplicateFilter* module in the **Network Manager**.
3. In the **Network Manager**, click the input connection pad ▷ on the left side of the *Gridder* module to connect the two modules. The connection line changes from blue to yellow when the cursor is over a module to which it can be connected. The connection line color changes to black when the connection is completed.

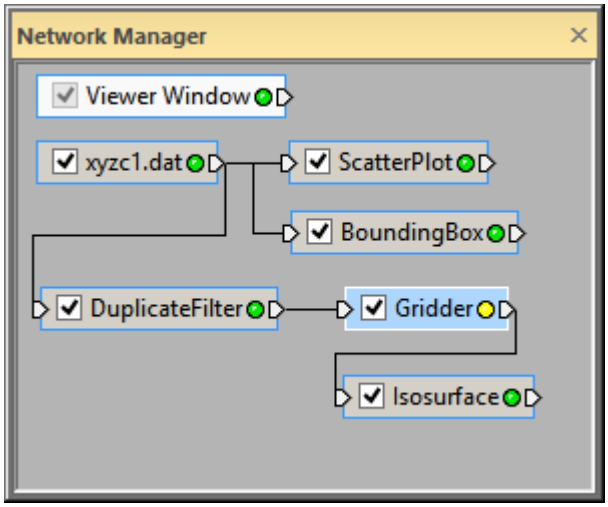


Click on the DuplicateFilter output connection pad and move the cursor to the Gridder module to connect the modules.

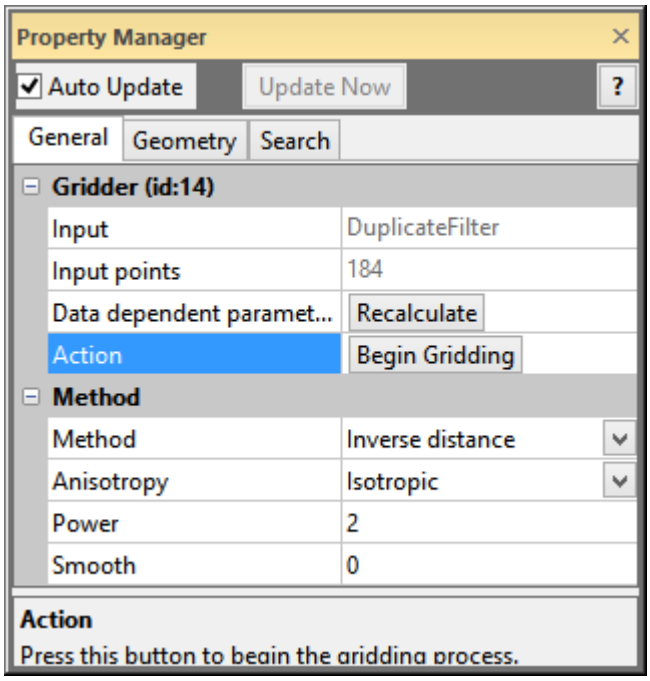
Since the *Gridder* module accepts only one input, connecting the *DuplicateFilter* module causes the *Gridder* module to automatically disconnect from the *xyzc1.dat* module. In addition, the *Gridder* module indicator LED turns yellow indicating that additional attention is required. Once the gridding is complete, the *Isosurface* module automatically updates and the new graphical output is sent to the **Viewer** window.

To update the *Gridder* and *Isosurface* modules:

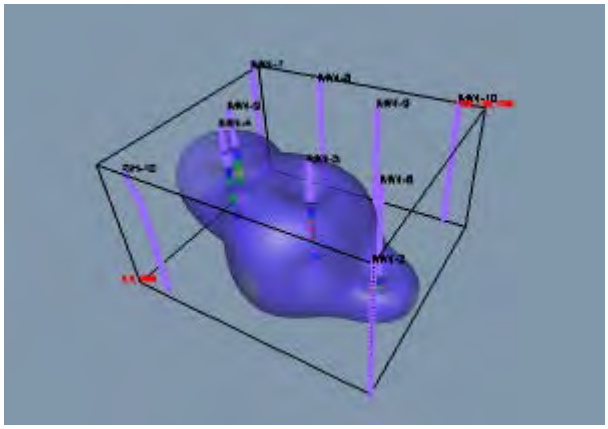
1. In the **Network Manager**, click the *Gridder* module to view its properties in the **Property Manager**.
2. In the **Property Manager**, click on the **General** tab.
3. Click the *Recalculate* button next to *Data dependent parameters*. This recalculates the lattice limits and other parameters to use the new input coming in from the *DuplicateFilter* module.
4. Click the *Begin Gridding* button in the **Property Manager**. The progress gauge displays the gridding progress and the *Gridder* module indicator LED turns green when the gridding is complete. The *Isosurface* module automatically updates with the new information and the results display in the **Viewer** window.



*Select the Gridder module in the **Network Manager** to select it.*



Adjust the Gridder properties and re-grid the data to account for the new DuplicateFilter module.



After the data are regridded, the Isosurface automatically updates to reflect the changes since it is "downstream" from the Gridder module.

Lesson 6 - Connecting Multiple Modules

Modules can have multiple connections. For example, the output from a *Gridder* module can be connected to several graphics output modules to show multiple aspects of the data in one graphic.

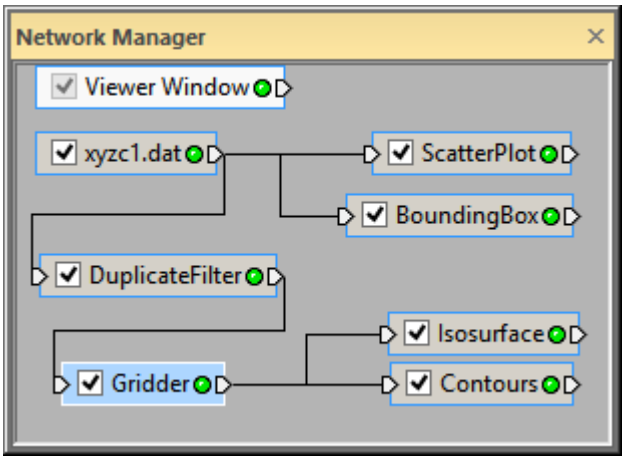
6.1 - Adding Contours - Tutorial

To add another graphics output module to the *Gridder* module:

1. Click the *Gridder* module in the **Network Manager**.
2. In the **Module Manager**, double-click the *Contours* module in the *Graphics Output* section to connect it to the *Gridder* module. Alternatively, right-click on the *Gridder* module and select

Graphics Output | Contours. The **Network Manager** now contains a connected *Contours* module and the **Viewer** window displays contours with the default settings.

3. To visualize the connections better, click on the *Gridder* module and drag it under the *DuplicateFilter* module.



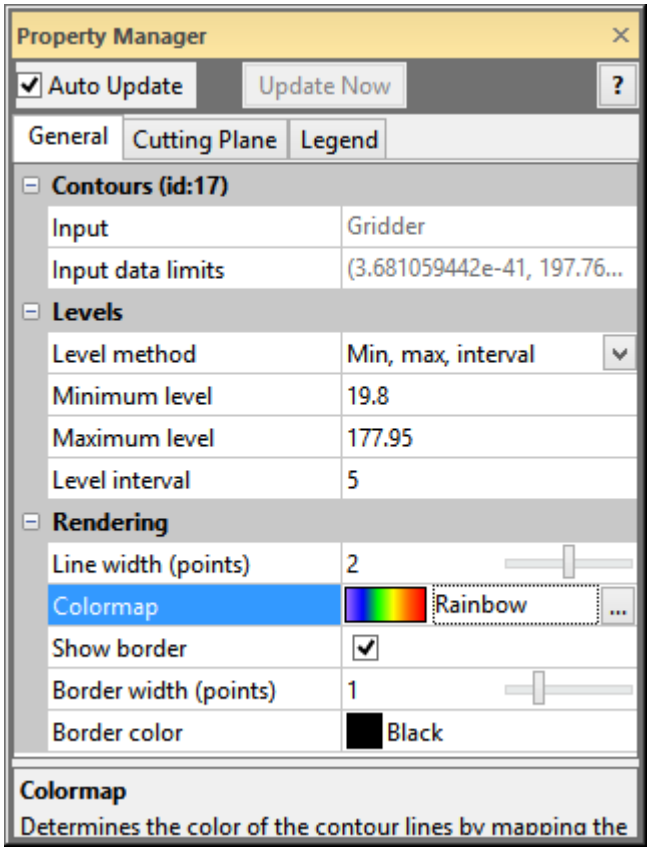
Click on the Gridder module and drag it to a new location to visualize the connections better.

Sometimes, the default settings are not exactly what we want to show in the **Viewer** window.

To change the contour properties:

1. Click the *Contours* module in the **Network Manager.**
2. In the **Property Manager**, click on the **General** tab.
3. Click *Automatic* next to *Level method* and choose *Min, max, interval*. This property section allows you to set the minimum and maximum contour values, and the contour interval (number of units between contour lines).
4. In the **Property Manager**, double-click on the default *Level interval* value, type 5, and press ENTER.
5. In the **Property Manager**, locate the property named *Colormap*. Click the sample color spectrum (*GrayScale*) to the left of the ... button to open the drop down list. Click *Rainbow* to change the colors of the contour lines.

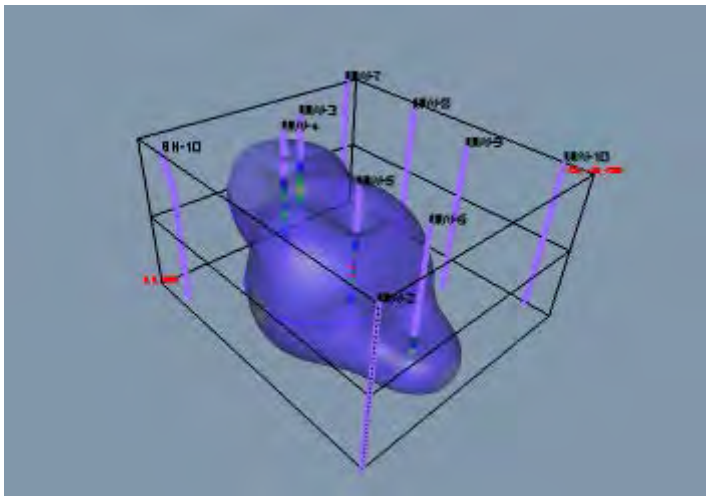
With each contour property change, the contours update automatically in the **Viewer** window.



*Adjust the Contours properties in the **Property Manager**.*

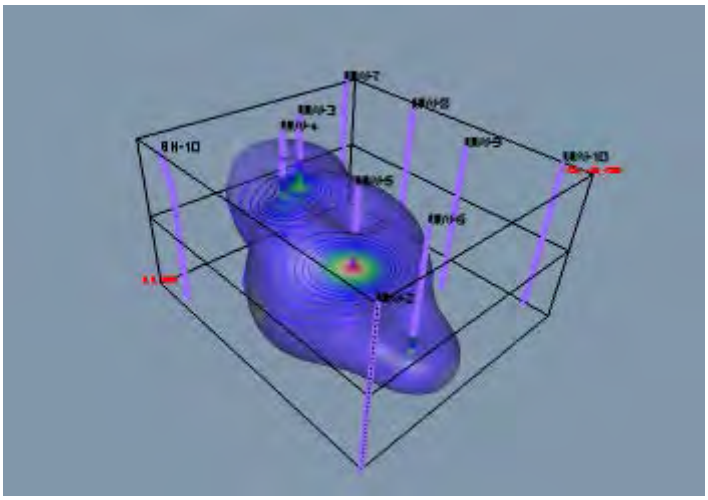
6.2 - Changing the Transparency - Tutorial

As discussed in *A Note About Transparency*, the transparency settings may need adjusting to fit the needs of your project. All of the contours may not be visible in the **Viewer** window depending on the transparency settings.



The default Blend transparency type does not create the ideal output.

To change the transparency mode to delayed blend choose the **View | Transparency Type | Delayed Blend** command. Alternatively, right-click in the **Viewer** window and select **Transparency Type | Delayed blend** from the context menu. Experiment with the transparency options to see how it affects the scene.



The Delayed Blend as the Transparency Type is a better choice for this project and displays the contour lines through the transparent isosurface.

Lesson 7 - Saving Information

There are several ways to save **Voxler** information:

- Select **File | Save** to save the data set and all of its associated modules as a **Voxler** project file.
- Select **File | Save** in the worksheet window to save the worksheet in a data file format.
- Select **File | Save Data** to save a selected module's data.
- Select **File | Save Copy As** while viewing linked data in the worksheet to save changes made to the linked module's data.
- Select **File | Export** to export graphic files such as bitmaps.
- Select **Actions | Copy Snapshot** to copy the **Viewer** window view to the clipboard.
- Select **Actions | Capture Video** to capture the screen rotation and save to an .AVI.

If you are using the demo version of **Voxler** you will not be able to use the save or export commands, so please skip to *Lesson 8*.

7.1 - Saving a Project - Tutorial

Click the **File | Save As** command to save the project as a .VOXB file. **Voxler** project files contain all of the source data, modules, connections, and graphics.

To save a **Voxler** project file:

1. Click the **File | Save As** command. The **Save As** dialog opens.
2. Type *TUTORIAL* into the *File name* field. Note there is only one option in the *Save as type* list, *Voxler Project Files (*.voxb).*

3. Click the *Save* button and the dialog closes. The project is saved so that it can be reused in **Voxler**. The project file format includes all data, including the raw source modules, and everything else needed to reload the project in the future.

If you are using the demo version of **Voxler** you will not be able to use the **File | Save** or **File | Save As** commands.

7.2 - Saving Data - Tutorial

Choose the **File | Save Data** command to save the output data from the currently selected module. You may also select the module whose output you want to save, right-click the module and select **Save Data**. Data can be saved to many different formats, depending on the type of module selected.

To save the selected module's output data:

1. Click the *DuplicateFilter* module in the **Network Manager.**
2. Click the **File | Save Data** command. Alternatively, right-click on the module and select **Save Data**. The **Export** dialog appears.
3. Type tutorial duplicate filter into the File name field.
4. Select DAT Golden Software Data (*.dat) in the Save as type box.
5. Click the *Save* button.
6. In the **Data Export Options** dialog, accept the defaults and click the *OK* button. The data are saved as a point set in the specified location.

If you are using the demo version of Voxler you will not be able to use the **File | Save Data** command.

7.3 - Saving a Graphic - Tutorial

Click the **File | Export** command to export the entire scene. This is a graphics-only export, so no data are saved for this operation. **Voxler** is designed to export the visible portion of the plot when using the **File | Export** command. It uses the monitor to define the limits of the exported image. To export a larger or smaller portion of the display, zoom in or out prior to export.

To save a graphic:

1. Click the **File | Export** command. The **Export** dialog opens.
2. In the **Export** dialog, type *TUTORIAL GRAPHIC* into the *File name* field.
3. Keep BMP Windows Bitmap (*.bmp) in the Save as type field.
4. Click the *Save* button. The **Export Options** dialog opens.
5. In the **Export Options** dialog, leave the default selections and click the *OK* button. The image is saved as a bitmap .BMP file in the specified location.

If you are using the demo version of Voxler you will not be able to use the **File | Export** commands.

7.4 - Copying a Snapshot - Tutorial

Choose the **Actions | Copy Snapshot** command to copy a raster version of the current view of objects in the **Viewer** window to the clipboard. The size of the copied image is the same as it appears in the current **Viewer** window.

To paste the raster graphics to another program, switch to the other program and choose the **Edit | Paste** command or press CTRL+V.

If you are using the demo version of **Voxler** you will not be able to use the **Actions | Copy Snapshot** command.

7.5 - Capturing a Video - Tutorial

Saving the file to a **Voxler** .VOXB file using the **File | Save** command is a great way of sharing models. Any other **Voxler** user can open the .VOXB file. The .VOXB file will appear exactly as you see it. The other person can rotate the graphic to the desired orientation and experiment with any settings. If the person does not have **Voxler**, they can download the demo to experiment with the model themselves.

If the person simply wants to view the model, capturing and sending them a video is a good way to share the information contained in the **Voxler** model. You can rotate the view as you create the video, you can zoom in and out, or turn modules on or off, to present the information in the video in the way you prefer.

To capture a video:

1. Adjust the **Viewer** window to the desired size for the video by clicking the window Restore button. Alternatively you can adjust the size of the **Voxler** application window. The video capture will use the size of the current **Viewer** window. The size of the window will affect the file size of the final .AVI file.
2. Click the **Actions | Capture Video** command. The **Capture Video** dialog opens.
3. Change the *Path* to specify a location where you want to save the captured video. The default path location is *C:\Users\<user name>\Documents\VoxlerVideo.avi*.
4. Adjust the *Frame rate* to 15, which is a value that produces good results.
5. Adjust the *Quality* to 80% with the slider.
6. Click the *Start Capture* button to begin the video capture. The *Estimated time (sec)*, *Estimated file size (MB)*, and *Estimated frames* information is dynamically displayed during capture.
7. Rotate or zoom the **Viewer** window if you wish.
8. Click the *Stop Capture* button to end the video capture. The *Estimated time (sec)*, *Estimated file size (MB)*, and *Estimated frames* information is statically displayed when the video has successfully completed.

If you are using the demo version of Voxler you will not be able to use the **Actions | Capture Video** command.

Lesson 8 - Importing and Displaying Wells

Voxler can import well data from numerous sources, including from LAS files. Typically the well data (logs) will be imported from one file and the physical location of the well (collars) will be imported from another file. Often, trajectory data indicating the direction of the well trace will also be imported, when the well is not assumed to be vertical.

To start this project in a new empty window, click the **File | New | Project** command. If you have not already done so, you can save the previous tutorial information.

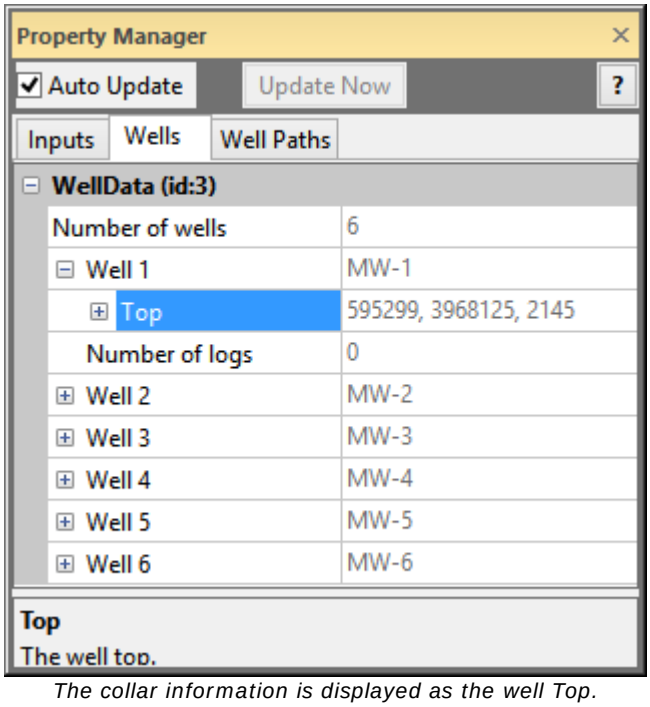
8.1 - Importing Well Collar Data

For this example, well collars, trajectory data, and log data are on three tabs of an Excel spreadsheet. Each tab is imported separately, with the following steps.

1. Click the **File | Import** command.
2. In the **Import** dialog, select the well collar file. For this example, select the *SampleWellData 2.xlsx* file from the Samples directory. By default, the Samples directory is located at C:\Program Files\Golden Software\Voxler 4\Samples.
3. Click *Open*.
4. In the **XLSX Import Options** dialog, select the *Collars* table and click *OK*. A *Data Source* module is added to the **Network Manager**.
5. Click on the *SampleWellData 2.xlsx - Collars* module to view its properties in the **Property Manager**.
6. In the Property Manager
    a. In the *Output* section set the *Output type* to *Wells*. Note that the column properties change from point data properties to well data properties.
    b. In the *Well Columns* section, set the *Sheet type* to *Collars* by clicking the current selection (*All*) and selecting *Collars* from the list.
7. Double-click on *WellData* in the *Well* folder of the **Module Manager**, or right-click on the *SampleWellData 2.xlsx - Collars* module and select **Well | WellData**. A *WellData* module is added and automatically connected to the collars data module.

In the **Property Manager** for the *SampleWellData 2.xlsx - Collars* module, you can see that the data columns were automatically specified correctly once the *Sheet type* was changed to *Collars*. **Voxler** assigns data columns based on column order for both well and point data. Refer to the *Data Source Module* topic in the online help for more information about automatic column assignment.

In the **Property Manager**, the **Wells** page shows the six wells that were imported. If you click on the ⊞ next to any well, you can see the *Top* information that was imported for that well.



The collar information is displayed as the well Top.

8.2 - Importing Trajectory Data

At this point, all that has been imported is the collar, or top location, of the six wells. The well trajectory is the trace of the well. The trace defines the way the location of the well moves as it gets deeper. To import the well trajectories, follow the steps in this section.

1. Click the **File | Import** command.
2. In the **Import** dialog, select the well file. Select the *SampleWellData 2.xlsx* again and click *Open*.
3. In the **XLSX Import Options** dialog, select the *Trajectories* table and click *OK*.
4. In Property Manager
    a. Check to make sure the *Output type* is set to *Wells*. The *Output type* selection is remembered from the last import.
    b. Change the *Sheet type* property to *Directional Survey*. Notice that a different set of properties is visible in the **Property Manager** than for the *Collars* selection in the previous section. Again the columns are specified correctly because of the order of the data in the worksheet.
5. Click the output connection pad ▷ on the *SampleWellData 2.xlsx - Trajectories* module. Next click the input connection pad ▷ on the *WellData* module.
6. In the context menu, select **Connect Input worksheet B (not connected)** to finish connecting the two modules.

The information is imported, and the actual X, Y, and Z values for the path of the well are automatically calculated by the program.

8.3 - Displaying the Wells

At this point, we can display the wells by clicking the **Network | Graphics Output | WellRender** command. Alternatively right-click the *WellData* module in the **Network Manager** and select **Graphics Output | WellRender**. The well traces are then displayed.



*The initial well traces are displayed in the **Viewer** window.*

8.4 - Importing Log Data

The well appearance can be altered by importing additional data.

1. Click the **File | Import** command.
2. In the **Import** dialog, select the well file. Select the *SampleWellData 2.xlsx* again and click *Open*.
3. In the **XLSX Import Options** dialog, select the *Samples* table and click *OK*.
4. In the Property Manager
    a. Again verify that the *Output type* is set to *Wells*.
    b. Change the *Sheet type* property to *From / To Logs*.

   c. Change the *Log columns* property value to *2* by typing the value in the field or clicking the button.
   d. Click on the current selection in the *Log-2* field and select *Column E: MnO* from the list.
5. Click the output connection pad on the *SampleWellData 2.xlsx - Samples* module. Next click the input connection pad on the *WellData* module.
6. In the context menu, select **Connect Input worksheet** C (not connected) to finish connecting the two modules.

8.5 - Displaying Log Data on the Wells

Once the log data is imported, the well appearance can be modified to display the logs using these steps:

1. Click on the *WellRender* module in the **Network Manager**.
2. Click on the Interval Data tab in the Property Manager.
3. Check the box next to *Show intervals* to add variable width log information.
4. Change the *Interval log* to *Column E: MnO* to use the MnO log *To Depth* and *From Depth* as the interval definition.
5. Change the *Color method* to *By log* so the colors of the intervals vary with log data.
6. Set the Color log to Column E: MnO.
7. Change the *Colormap* to *Rainbow* to display different colors along the length. The colors are determined by mapping the values in the MnO log to the colors in the *Rainbow* colormap.
8. Set the Size method to By log.
9. Set the *Size log* to *Column D: TiO2*. The width of the log is now determined by the data in the TiO2 log.



The wells are now displayed as tubes of variable width, based on the data in the TiO2 log.

Advanced Tutorials

By completing Lesson 1 through 7 of the tutorial, you now have a basic understanding of **Voxler**.

The remaining tutorial lessons are optional lessons that are highly recommended. Beginner users through advanced users can benefit from the additional tutorials.

Lesson 9 - *Working with a HeightField module*

Lesson 10 - *Working with a Math Module*

HeightFields - Advanced Suggestions

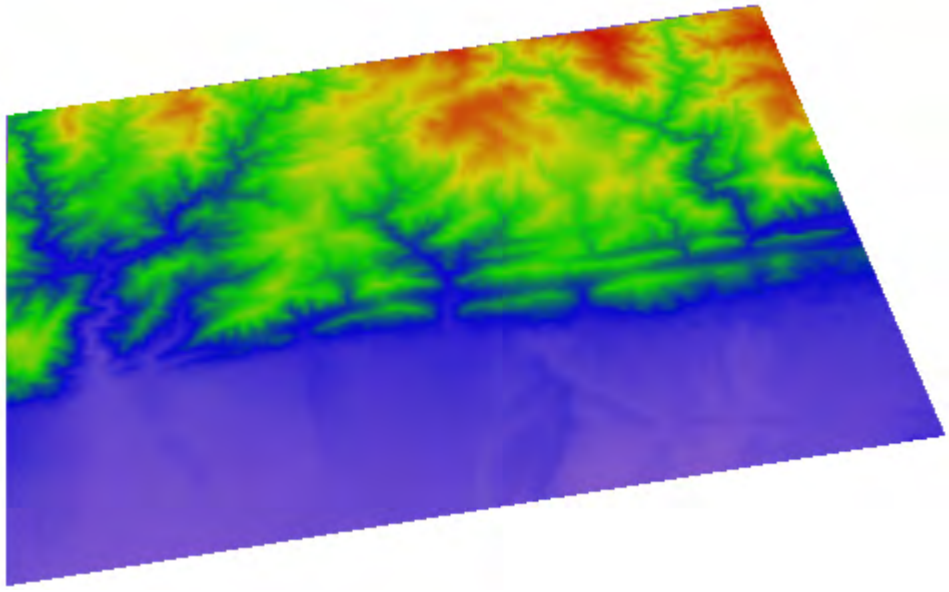
This tutorial provides suggestions when using the *HeightField module*.

Two import methods are discussed in this advanced tutorial. You will import the same grid file using both methods.

Method 1

In the first method, you will load a Surfer grid file .GRD with a flat Z value where the gridded Z value is imported as the C value.

1. Click the **File | New** command to open a new **Viewer** window.
2. Click the **File | Import** command.
3. In the **Import** dialog, select the sample grid file *BoulderColorado.grd* from the **Voxler 4** Samples folder (located at: C:\Program Files\Golden Software\Voxler 4\Samples) and click the *Open* button.
4. In the **Lattice Import Options** dialog, select the *Import as uniform lattice (default)* option and click *OK*. Selecting this option brings the grid file into **Voxler** so that the Z values are imported as zeros and the actual Z values are assigned to the C component value, allowing for XYZC data to be imported into **Voxler**.
5. The data module *BoulderColorado.grd* is displayed in the Network Manager. Right-click on the data module and select **Graphics Output | HeightField**.
6. Select the *BoulderColorado.grd* module in the **Network Manager** to display the properties in the **Property Manager**. Click the View Data button to see the data that was imported into **Voxler**.
7. In the **Lattice View** dialog, click the *Show: Z Coordinate* on the left side. The Z values are all zeros. Click the *Show: Component 1* on the left side. The C values are the grid file Z values. In order to properly display the elevation values, a scale factor has been added in the *HeightField* by default. Click the X button to close the **Lattice View** dialog.
8. Select the *HeightField* module in the **Network Manager** to display the properties in the **Property Manager**. Notice that the scale is set to 0.25. Change the *Scale* to 0 and the grid file is displayed as a flat surface, because the Z values are actually zeros.
9. Change the Colormap to Rainbow.

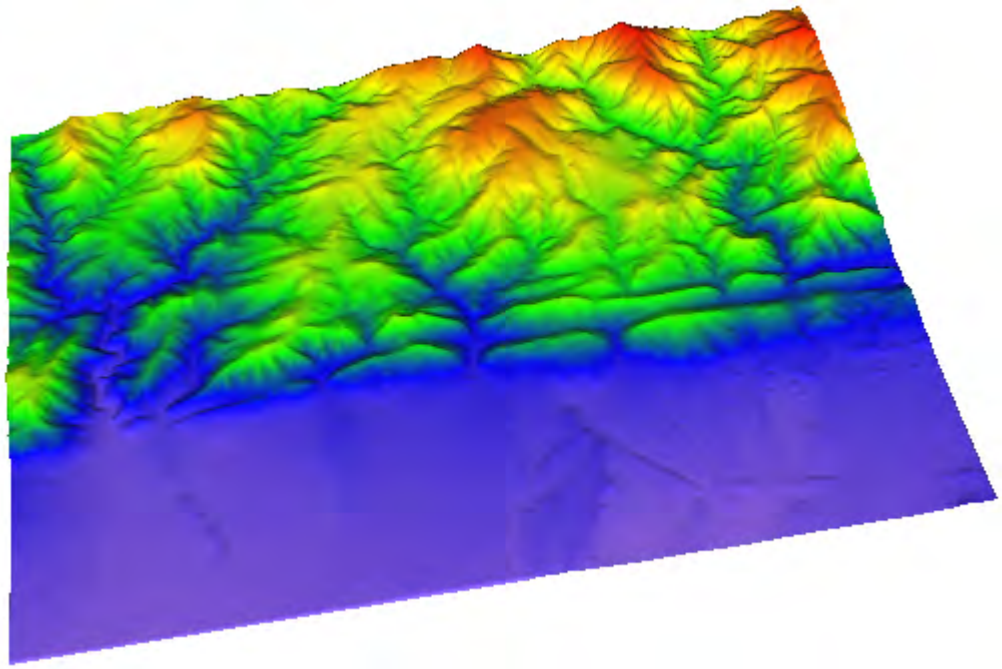


Using the Import as Uniform Lattice (default) option imports the grid file Z values as zeros and the Z values as the component. This is easy to see when the Scale Factor is set to zero.

Method 2

In the second method, you will import a Surfer grid file .GRD at correct height, add a *HeightField* module, and manually adjust the *HeightField* properties.

1. Click the **File | Import** command.
2. In the **Import** dialog, select the sample grid file *BoulderColorado.grd* from the **Voxler 4** Samples folder (located at: C:\Program Files\Golden Software\Voxler 4\Samples) and click the *Open* button.
3. In the **Lattice Import Options** dialog, select the *Import as curvilinear lattice* option. Check the *Use component minimum* option. Click the *OK* button.
4. The data module *BoulderColorado.grd 2* is displayed in the **Network Manager**. Right-click on the data module and select **Graphics Output | HeightField**.
5. Select the *BoulderColorado.grd* module in the **Network Manager** to display the properties in the **Property Manager**. Click the View Data button to see the data that was imported into **Voxler**.
6. In the **Lattice View** dialog, click the *Show: Z Coordinate* on the left side. The Z values are all the Z data values from the grid file. Click the *Show: Component 1* on the left side. The C values are the grid file Z values. This allows the Z values to be properly display the elevation values with no scale factor. Click the X button to close the **Lattice View** dialog.
7. Click on the *HeightField 2* module in the **Network Manager**. In the **Property Manager,** change the *Colormap* to *Rainbow.*



Using the Import as Curvilinear Lattice option imports the grid file Z values both as Z values and as the component. This is easy to see when the Scale is set to zero.

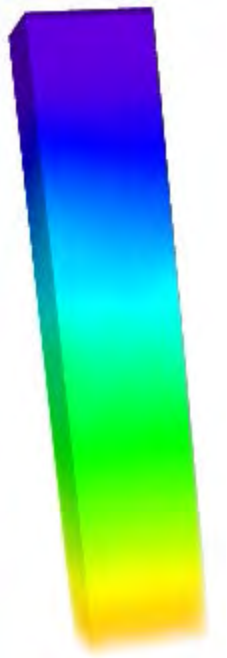
Math Module - Advanced Suggestions

This tutorial provides suggestions when using the *Math module*.

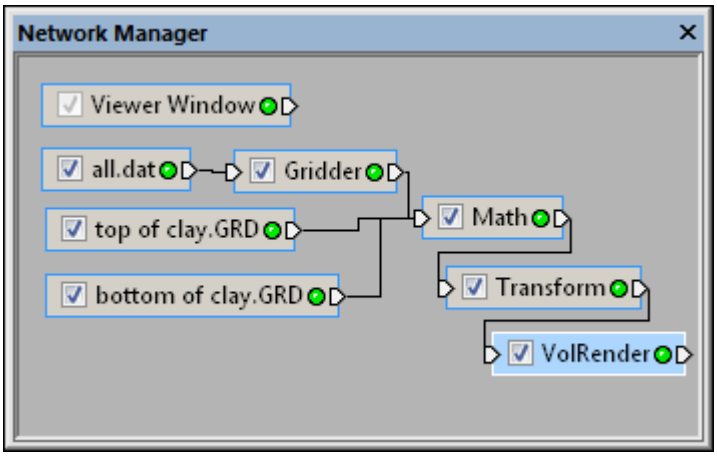
Working with the *Math* module to fill between two *HeightField modules:*

1. Create the top and bottom of the surface grid files in **Surfer** or another program.
2. In **Voxler**, click the **File | Import** command. Select the top surface grid file. Click the *OK* button. The data module is loaded into the **Network Manager**.

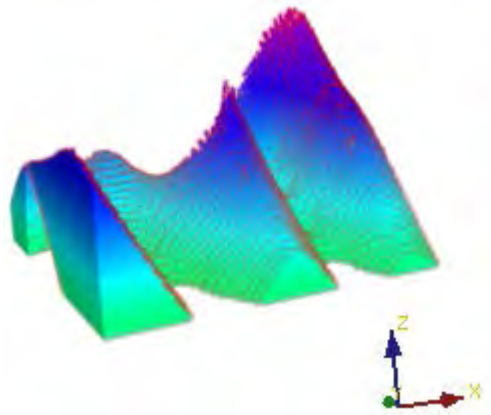
3. Click the **File | Import** command. Select the bottom surface grid file. Click the *OK* button. The data module is loaded into the **Network Manager**.
4. Click the **File | Import** command. Select data file that was used to grid the top surface. Set the X and Y columns to the correct columns. For the Z and C columns, set the value to the top of the surface.
5. Select the data file in the **Network Manager**. Right-click on the data module and select **Computational | Gridder** to add a *Gridder* module to grid the data.
6. Select the *Gridder* module in the **Network Manager**. In the **Property Manager**, click the *Begin Gridding* button.
7. Right click on the *Gridder* module in the **Network Manager** and select **Computational | Math** to add a *Math* module to the gridded data.
8. Select the top of surface grid module in the **Network Manager**. Right-click and select **Connect output lattice**. Drag the connection line to the *Math* module and select **Connect Input Lattice B**.
9. Select the bottom of surface grid module in the **Network Manager**. Right-click and select **Connect output lattice**. Drag the connection line to the *Math* module and select **Connect Input Lattice C**.
10. Select the *Math* module in the **Network Manager**. In the **Property Manager**, change the *Expression[1]* to IF Z > B OR Z < C, 0, A.
11. If necessary, attach a *Transform* to the *Math* module using the **Network | Computational | Transform** command. This may be necessary to scale the Z height.
12. Connect a *VolRender* to the *Transform* module, if it exists, or to the *Math* module using the **Network | Graphics Output | VolRender** command. Change any properties of the volrender.



This example shows all of the data displayed as a VolRender module.



*The **Network Manager** contains the original data file, the two grid files, the Gridder, Math, Transform, and VolRender modules. Using the Math module allows you to display fill between two surfaces (as seen below).*



The resulting output is a VolRender that displays the fill between two clay surfaces.

Tutorial Completion

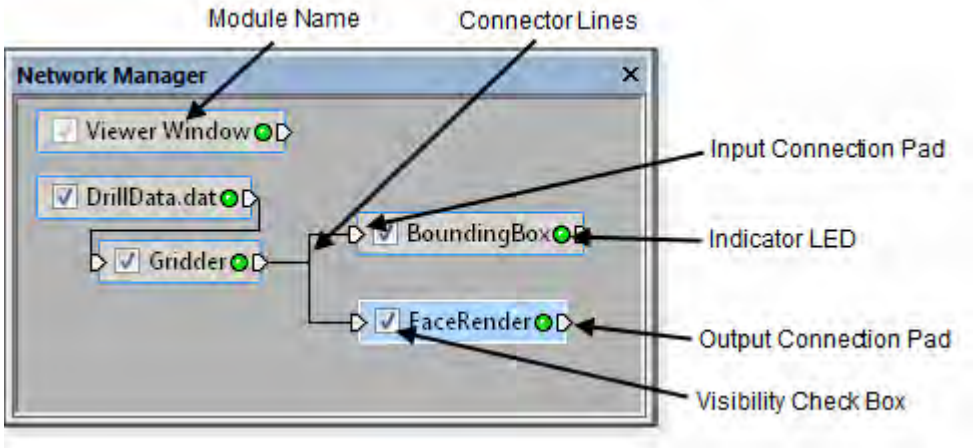
Congratulations, you have completed the **Voxler** tutorial! Training videos are available on the Golden Software website.

Chapter 3 - General Modules

Introduction to Modules

A module is a data set or a process to be applied to a data set or process. Modules are the building blocks from which the final **Voxler** output is constructed. Modules accept data on their input connection pads ▷, modify the data, and pass it along through the output connection pad ▷.

Modules are displayed in the **Network Manager**. You can connect and disconnect modules to create a visualization network representing the flow of data. Modules need to be connected in order to generate an output. The current geometry output of the modules in the **Network Manager** are displayed in a three-dimensional view in the **Viewer** window. If the data is not connected to a graphics output module, nothing is displayed in the **Viewer** window.



*The **Network Manager** displays the visualization network, which includes all loaded modules and their connections.*

View All Modules

All modules are accessed using the Network menu commands. Alternatively, all modules are accessed in the **Module Manager** when the *Show all modules* button is selected .

View Applicable Modules

When a module is selected in the **Network Manager** and the *Show all modules* button  is not selected in the **Module Manager**, available modules that can be connected to the selected module output port are displayed in the **Module Manager**. Alternatively, right-click a module in the **Network Manager** to display only the applicable modules in the context menu.

On the module description pages, there are **Inputs** and **Outputs** sections to discuss the type of input and output modules that each module are compatible.

Import

Click **File | Import** to open the **Import** dialog. The type of data determines what type of module can be attached to it. **Voxler** supports several different data types. See the *File Format Chart* for a detailed list of supported file formats.

Viewer Window

The **Viewer Window** is a unique module that is automatically created when a new instance of **Voxler** is generated. The *Viewer Window* module appears in the **Network Manager**. The *Viewer Window* module cannot be deleted. The purpose of the *Viewer Window* module is to control the properties of the **Viewer** window for the current instance of **Voxler**. To change the **Viewer** window properties for future instances of **Voxler**, choose the **Tools | Options** command and adjust properties on the **Colors** page of the **Options** dialog.

Module Types

There are four types of modules: *computational*, *data source*, *general*, and *graphics output.* Each module type is discussed below.

Computational Modules

Computational modules alter the data by changing the data type, filtering, creating a gradient, gridding, performing mathematical transformations, merging, resampling, slicing, creating a subset, or transforming coordinates. Click on one of the following computational modules for detailed information on using the module and module properties.

ChangeType

DuplicateFilter

ExclusionFilter

ExtractPoints

Filter

Gradient

Gridder

Math

Merge

Resample

Slice

Subset

Transform

Data Source Modules

Data source modules serve as a source of raw data. The data can be imported or created from mathematical functions. Click on one of the following data source modules for detailed information on using the module and module properties.

Import (*Data Source*, *Point Source*, *Lattice Source*, *Geometry Source*)

FunctionLattice

TestLattice

WellData (combines multiple *Data Source* modules into a single output)

General Modules

General modules display module information and provide custom lighting in the **Viewer** window. Click on one of the following general modules for detailed information on using the module and module properties.

Info

Light

Graphics Output

Graphics output modules create graphics in the **Viewer** window. Typically, these modules require data input. Click on one of the following graphics output modules for detailed information on using the module and module properties.

Annotation

Axes

BoundingBox

ClipPlane

Contours

FaceRender

HeightField

Isosurface

ObliqueImage

OrthoImage

ScatterPlot

StreamLines

Text

VectorPlot

VolRender

WellRender

General Modules

Choose the **Network | General Modules** command to add a general module to the scene. General modules display module information and provide custom lighting in the **Viewer** window.

General modules include the following:

- *Info* displays information about the connected module
- *Light* adds a light source to the scene

Info

The **Network | General Modules | Info** command adds an *Info* module.

The *Info* module displays various information about the connected module, such as *data limits*, *number of components*, and *component type*. The information displayed in the properties for the *Info* module varies depending on the type of module to which it is attached.

Inputs

An Info module can be attached to almost any type of input data, such as a lattice, point set, well data, or geometry.

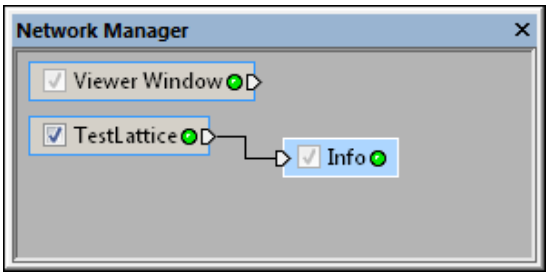
Two-dimensional graphics, e.g., legends and screen text, do not appear in the geometry counts.

Outputs

Info modules do not connect to any other output modules.

Properties

The *Info* module properties are described below. The options displayed vary depending on the type of data set or module to which the *Info* module is connected. These properties cannot be changed. They are shown for information purposes, only. Select the *Info* module in the **Network Manager**. The properties are displayed in the **Property Manager**.



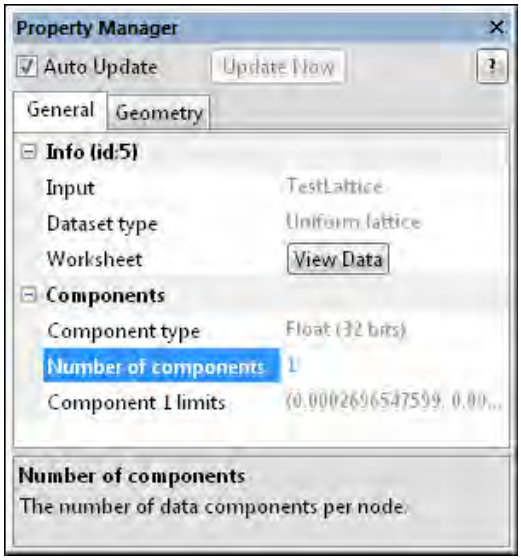
*Select the Info module in the **Network Manager** to display its properties in the **Property Manager**.*

The *Info* module contains the following tabs in the **Property Manager:**

General

Geometry

General Options



*This is an example of the information displayed in the **Property Manager** about the TestLattice module.*

Input

The *Input* property shows the source to which the module is connected. This option cannot be changed in the **Property Manager**, but can be changed in the **Network Manager** by changing the module input.

Dataset Type

The *Dataset type* item displays the data set type to which this module is connected.

Worksheet

Click the *View Data* button to display the data in a worksheet view. The data is displayed in the **Lattice View** dialog or the **Data View** dialog, depending on the type of data set loaded.

Any number of **Lattice View** or **Data View** dialogs can be open at a time. The data in the window do not automatically update if the input changes, as these views are a static snapshot of the data at the time they were opened. To update the data in the view window, close the view window. Select the *Info* module, and press the *View Data* button again. These views are not saved to disk with the network.

The following information is updated if the input is changed.

Component Type

The *Component type* item details the type of data storage associated with each data component for a lattice or point set.

Number of Components

The *Number of components* item details the number of data components per node for a lattice or point set.

Component Limits

The numbered *Component limits* items display the limits for each of the data components for a lattice or point set.

Number of Points

The *Number of points* item displays the number of points in the point set or geometry.

Coordinate Type

The *Coordinate type* details the type of data storage associated with the X, Y, and Z directions. This property cannot be changed.

Number of Lines

The *Number of lines* item displays the number of lines in the point set or geometry.

Number of Triangles

The *Number of triangles* item displays the number of triangles in the geometry.

Number of Text

The *Number of text* item displays the number of text strings in the geometry.

Number of Images

The *Number of images* item displays the number of images in the geometry.

Number of Wells

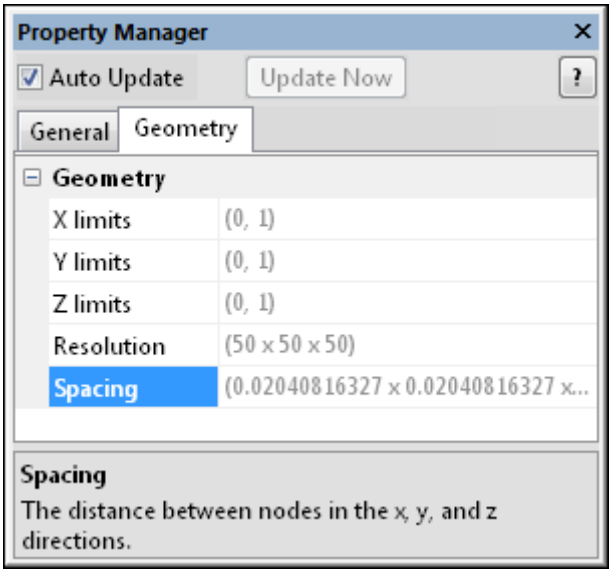
The *Number of wells* item displays the number of wells in the well data.

Well Information

Each well is listed separately. The name of the well is displayed next to *Well #*. The *Top*, *Number of logs*, *Log name*, *MD range*, and *Data range* are listed for each well.

Info Module - Geometry Page

The *Info* module **Geometry** page displays the extents of the data, grid, or geometry module. To open the **Geometry** page, click on the *Info* module in the **Network Manager**. In the **Property Manager**, click on the **Geometry** tab.



This is an example of the information displayed in the Property Manager on the Geometry tab for a Info module.

X Limits

The *X limits* item displays the limits in the X direction in the point set, geometry, or lattice.

Y Limits

The *Y limits* item displays the limits in the Y direction in the point set, geometry, or lattice.

Z Limits

The *Z limits* item displays the limits in the Z direction in the point set, geometry, or lattice.

Resolution

The *Resolution* item displays the number of nodes in the x, y, and z directions for a lattice.

Spacing

The *Spacing* item displays the distance between nodes in the x, y, and z directions for a lattice.

Light

The **Network | General Modules | Light** command adds a *Light* module to the **Viewer** window.

The *Light* module creates a new directional, point, or spot light and adds it to the scene. Lights are cumulative. Every time a new light is added, the scene becomes a little brighter. You can add approximately seven lights to the scene.

The headlight is automatically attached to the camera, which can make it difficult to see the affect of added lights. To view a scene with only light modules, uncheck the **View | Headlight** command to turn off the global light.

Lights may be positioned and oriented numerically, or by enabling the dragger in the **Property Manager**.

Inputs

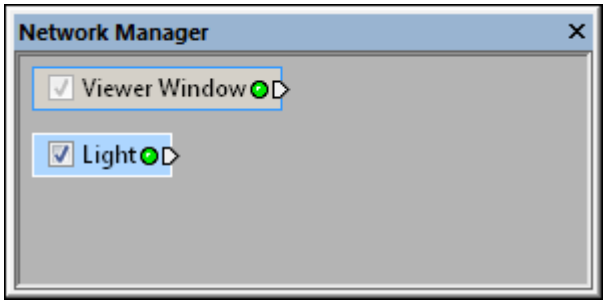
The *Light* module is added to the scene, not an individual module. The *Light* module is not connected to any module on the input side.

Outputs

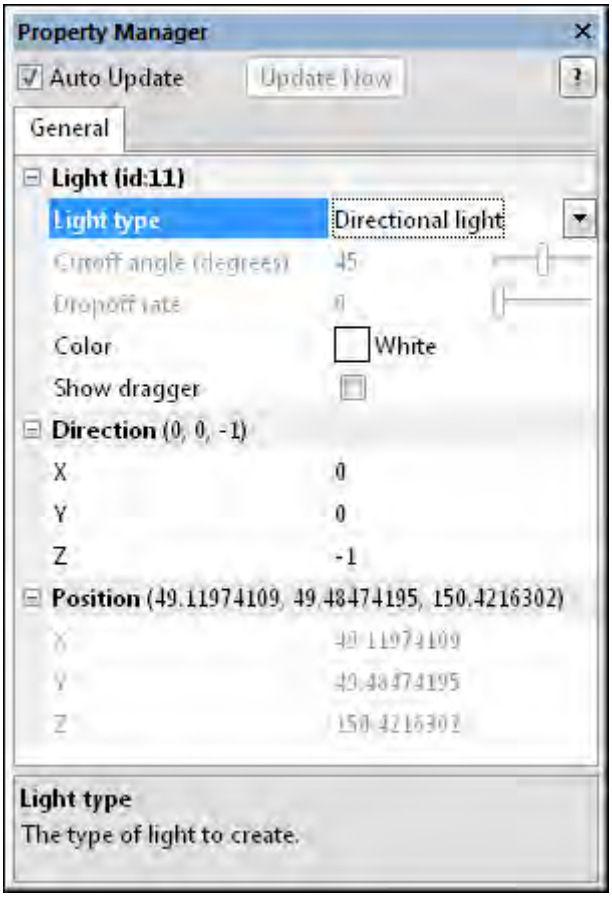
The *Info* and *ClipPlane* modules can be connected to a *Light m*odule on the output side. Adding the *Info* module provides information about the light. Adding a *ClipPlane* module limits the light to only certain locations in the scene.

Properties

The *Light* module properties are described below. Click on the *Light* module in the **Network Manager**. The properties are displayed in the **Property Manager**.



*Select the Light module in the **Network Manager** to display its properties in the **Property Manager**.*



*Change the light properties in the **Property Manager**.*

Light Type

To change the *Light type*, click on the existing light type option and select the desired option from the list. Available options are *Directional light*, *Point light*, or *Spot light* to brighten the image and create special effects. Each type of light has different properties and displays differently in the **Viewer** window.

For a *Directional light*, the *Direction X*, *Y*, and *Z* properties specify a direction vector pointing in the direction the light rays travel. Directional lights do not have a position; only the direction is relevant.

For a *Point light*, the *Position X*, *Y*, and *Z* specify the position of the light source in map coordinates. The light radiates outward from the source symmetrically in all directions like a star.



A *Spot light* has both a *Direction* and a *Position*. Like a theatrical spotlight on a stage, the spot light illumination is a cone of light diverging from the light's *Position*. The axis of the cone is oriented according to the specified *Direction*. The location of the spot light is defined by the *Position*.



Color

The *Color* option changes the color of the light source. This can affect the color of the modules being displayed in the **Viewer** window. To change the color, click on the existing color and select a new color from the color palette.

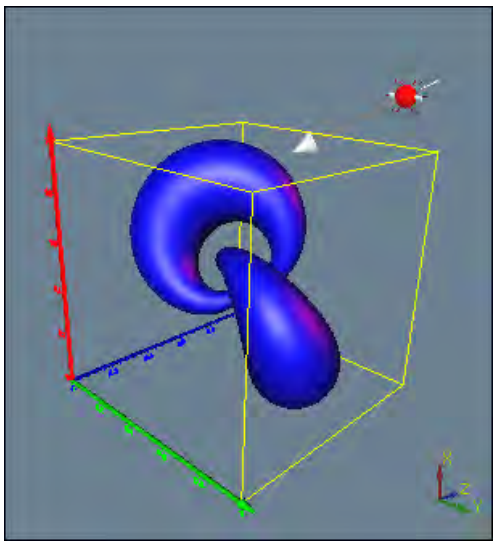
Show Dragger

Check the box next to the *Show dragger* command to show the light dragger so it can be interactively adjusted in the **Viewer** window. The light dragger is scaled to be a reasonable size according to the existing geometry at the time the dragger is turned on. If new geometry is added or removed, the dragger size may become too large or small relative to the new geometry. In this case, it may be necessary to turn the dragger off and on in order for **Voxler** to display the dragger at a proportionate size to the new geometry.

The *Directional light* dragger can be moved or reoriented; however, the actual position of the dragger is irrelevant. Drag either end of the arrow to reorient the light.

The *Point light* dragger can only be moved. Drag the spherical portion to move the dragger in the plane. Drag the rectangular bar to move the dragger perpendicularly to the plane.

The *Spot light* dragger can be moved, oriented, and have its cone angle adjusted. Drag either end of the arrow to reorient the direction of the light. Move the spherical portion to adjust the position of the light. Drag the rectangular bar to move the dragger perpendicularly to the plane. Drag the cone to adjust the *Cutoff angle (degrees)*. See *Dragger* properties for more details.



With the Show Dragger option on, the light source can easily be moved and adjusted. This example shows a blue Directional Light.

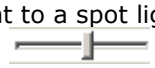
Direction

For *Directional light* and *Spot light* types of light, enter the *X*, *Y*, and *Z* vector values in the *Direction* section. The direction values can be manually adjusted in the **Property Manager**, or automatically adjusted using the *Dragger*. The *Direction* values are in vector units. A value of 0 will have no light in that direction. Values in the *X*, *Y*, and *Z* boxes are combined to get the vector direction. To change the value, highlight the existing value and type a new number.

Position

For *Point light* and *Spot light* types of light, enter the *X*, *Y*, and *Z* light location in the *Position* section. The position values can be manually adjusted in the **Property Manager**, or automatically adjusted using the *Dragger*. The *Position* values are entered in map units. To change the value, highlight the existing value and type a new number.

Cutoff Angle

For a *Spot light*, you can enter a value into the *Cutoff angle (degrees)* box. The cutoff angle box can be manually adjusted in the **Property Manager**, or automatically adjusted using the *Dragger* by altering the cone angle. The *Cutoff angle (degrees)* defines how light diverges from the spot light source. Values range between approximately 0 and 90. A value of zero would be equivalent to a directional light. A value of 90 would be equivalent to a spot light. To change the value, highlight the existing value and type a new value or drag the  to the desired level.

Dropoff Rate

For a *Spot light*, you can enter a value into the *Dropoff rate*. The dropoff rate is the rate at which the light intensity drops off from the primary direction. The dropoff rate is a value between 0 and 1. To change the value, highlight the existing value and type a new value or drag the  to the desired level.

Viewer Window Properties

The *Viewer Window* is a module in the **Network Manager** that displays information about the current properties of the **Viewer** window and contains various properties that affect the entire scene, such as background color. The *Viewer Window* is automatically created. New *Viewer Windows* cannot be added to an existing network. Existing *Viewer Window* modules cannot be deleted from an existing network. The *Viewer Window* is displayed in the **Network Manager**; it is not listed in the Module Manager since it always exists and cannot be deleted.

The *Viewer Window* only controls the options for the current **Viewer** window. To change the default **Viewer** window settings, click the **Tools | Options** command and click on the **Colors** tab. The *New viewer window background* controls the color of future **Viewer** windows.

Input: none - A single *Viewer Window* is automatically created and always present

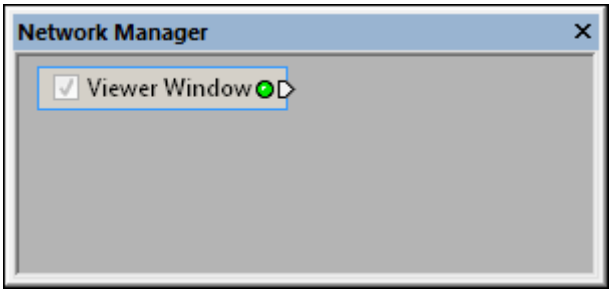
Output: *Info*

Changing the Default Viewer Window Properties

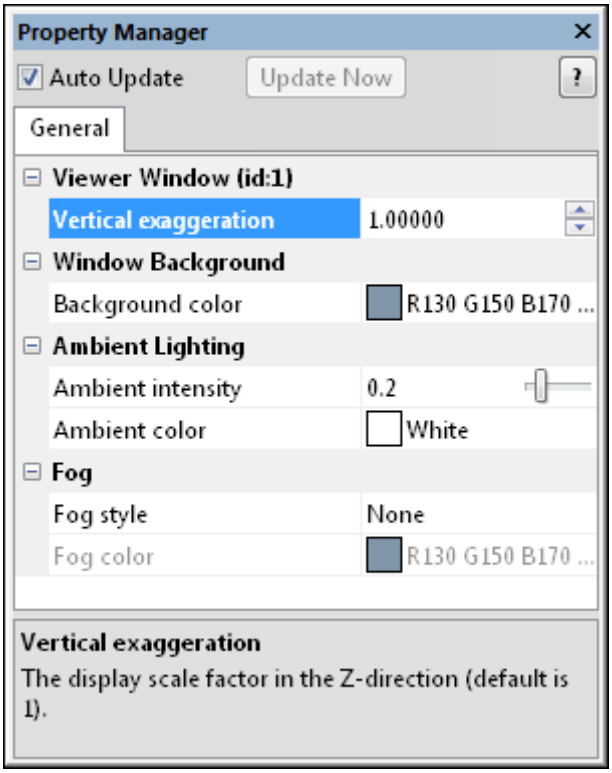
To change the default **Viewer** window properties that will remain for future **Voxler** instances, click the **Tools | Options** command to open the **Options** dialog. The **Colors** page contains options to customize the *Network manager background*, *New viewer window background*, *Viewer wireframe color*, *Viewer axis triad X*, *Viewer axis triad Y*, *Viewer axis triad Z*, and *Viewer axis triad text*. Select one of the options and then select a new color from the color drop-down list.

Editing the Viewer Window Properties

Select a module in the **Network Manager** to display the module properties in the **Property Manager**. The property options are discussed below.



*Select the Viewer Window module in the **Network Manager** to display the module properties in the **Property Manager**.*



*Change the appearance of the **Viewer** window by modifying the properties in the **Property Manager**.*

Vertical Exaggeration

The *Vertical exaggeration* option sets the scale for the Z direction for the entire project. The *Vertical exaggeration* sets the ratio of the X and Y scale over the Z scale. For the vertical exaggeration to be calculated and reported correctly, the X, Y, and Z units should all be the same units. For instance, all of the values should be in feet or all of the values should be in meters.

The *Vertical exaggeration* can range between 0.00001 and 100000. A *Vertical exaggeration* of 1 is considered no vertical exaggeration. This means that the distance covered by one page unit vertically is the same as the distance covered by the same page unit horizontally. When set to 10,

the distance covered by one page unit vertically is ten times as large as the distance covered by the same page unit horizontally. To change the *Vertical exaggeration*, highlight the existing value and type the desired value or click the  button to increase or decrease the value. Press ENTER on the keyboard to make the change.

Background Color

The background color is the color of the **Viewer** window where no objects are shown. The background color can make a difference in how objects appear in the **Viewer** window. The background color can also be exported or copied. To change the color of the **Viewer** window background, click the color next to *Background color*. The color palette opens. Choose an existing color, or click the *Other* button to open the **Colors** dialog. The **Colors** dialog allows you to select additional colors or create a new color.

Ambient Lighting

Ambient lighting is light that comes from all directions, without a single source. Ambient light has been scattered so evenly that its direction is impossible to determine. Ambient light does not cast shadows. All items in the **Viewer** window are equally lit by ambient lighting. To highlight a portion of the **Viewer** window, a *Light* module can be added to the **Network Manager**.

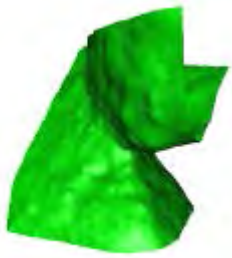
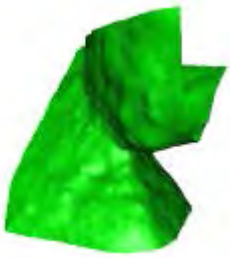
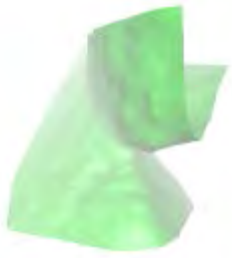
The *Ambient intensity* affects the brightness of the overall scene. Adjust the *Ambient intensity* by highlighting the existing value and typing a new value. Or, click and drag the  to a new value. The *Ambient intensity* value ranges between 0.0 and 1.0. A value of zero represents no light. A value of 1 represents the maximum amount of light. The larger the *Ambient intensity* value, the brighter the objects appear in the **Viewer** window.

The *Ambient color* changes the color of the ambient light. The light color is added to the color of the objects in the **Viewer** window. Click the current color next to *Ambient color* to change the ambient light in the scene. The color palette opens. Choose an existing predefined color or click the *Other* button to open the **Colors** dialog. The **Colors** dialog allows you to select additional colors or create a new color.

Fog Properties

Fog increases opacity as distance from the camera increases. Fog is applied to the entire scene, similar to ambient lighting. Change the *Fog style* to *None*, *Haze*, *Fog*, or *Smoke*, depending on how you want the fog applied to the **Viewer** window. See the chart below for fog style descriptions and visual examples. To change the *Fog style*, click on the existing option and select the desired option from the list.

The fog color is the color of the fog being added to the scene. For a more realistic affect, set the *Fog color* to be the same as the *Background color*. Click the color next to *Fog color* to change the fog color. The color palette opens. Choose an existing predefined color or click the *Other* button to open the **Colors** dialog. The **Colors** dialog allows you to select additional colors or create a new color.

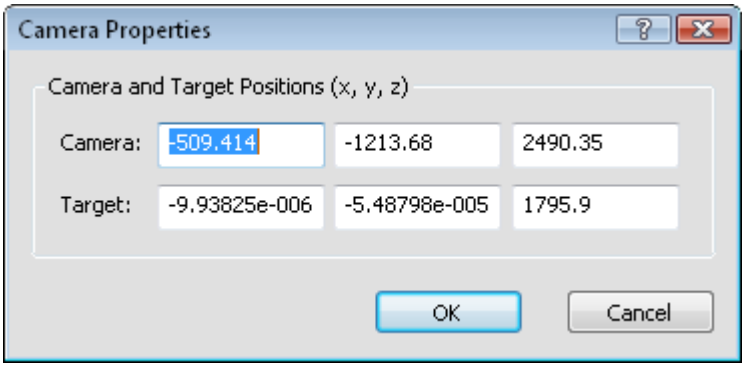
Fog Style	**Fog Style Description**	**Visual Example** **Fog Color** is same as background color = White	**Visual Example** **Fog Color** is different from background, RGB = 109, 129, 145
None	*None* applies no fog affect, setting the visibility of all objects equally independent of the distance from the camera.		
Haze	*Haze* sets the visibility to decrease linearly with distance from the camera.		
Fog	*Fog* sets the visibility to decrease exponentially with distance from the camera.		
Smoke	*Smoke* sets the visibility to decrease exponentially with the square of the distance from the camera, simulating very thick natural foggy conditions.		

Camera Properties

The camera properties control the zoom, rotation, and angle of objects in the **Viewer** window.

Change View with Camera Properties Dialog

Click the **View | Camera Properties** command or  button to change the camera and target positions. The **Camera Properties** dialog displays.



*You can change the Camera and Target Positions (x, y, z) in the **Camera Properties** dialog.*

Camera

The *Camera* position contains the coordinates of the viewer's eye (the camera). Enter a value for each coordinate (in the same units used by your data).

Target

The *Target* position contains the coordinates of the object at which the camera is looking. Enter a value for each coordinate (in the same units used by your data).

OK or Cancel

Click *OK* to save your changes and close the dialog. The view will update to show the new camera and target position. Click *Cancel* to return to the current view without making any changes.

Change View with Mouse

To change the current view with the mouse, click the **View | Trackball** command, position the mouse in the **Viewer** window, hold down the left mouse button, and drag the mouse to the desired position.

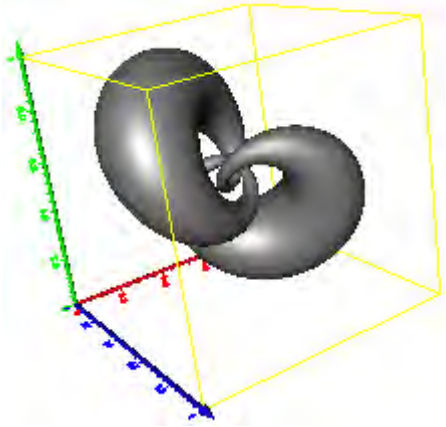
Default Home Position

Choose the **View | Home** command to undo all value changes and return to the default home position. Change the camera properties and choose the **View | Set Home** command to set the current view as the home position.

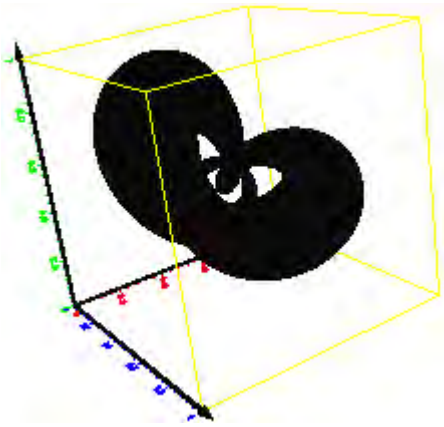
Changing the camera and target positions cannot be undone.

Headlight

Click the **View | Headlight** command or  button to turn the camera headlight on or off. The headlight is a directional light positioned at the camera which shines in the same direction that the camera is pointing. This light is usually left on unless additional light modules have been placed in the scene. Turning off all lights can result in a blank window.



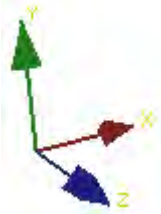
In this example, the headlight is turned on. This is the default setting.



In this example, the headlight is turned off.

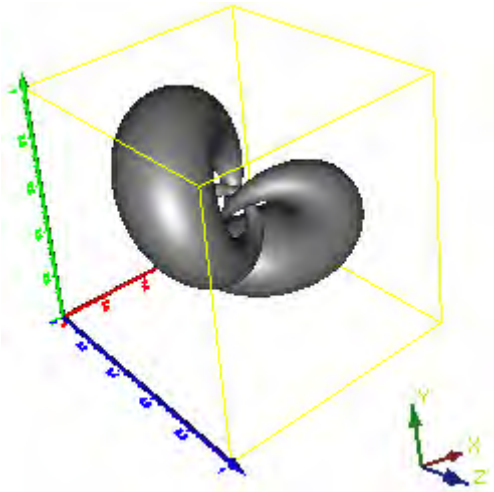
World Axis Triad

Click the **View | World Axis Triad** command or  button to turn the world axis triad on or off. This feature is located in the lower right corner of the **Viewer** window. The axis triad is informational only. Turning it off does not affect the network or geometry in any way.



This is the world axis triad.

The world axis triad is a depiction of the X, Y, and Z directions that shows the **Viewer** window camera orientation. The world axis triad draws the X axis in red, the Y axis in green, and the Z axis in blue.



This is an example of the world axis triad and the geometry it is reflecting.

Customize Axis Colors

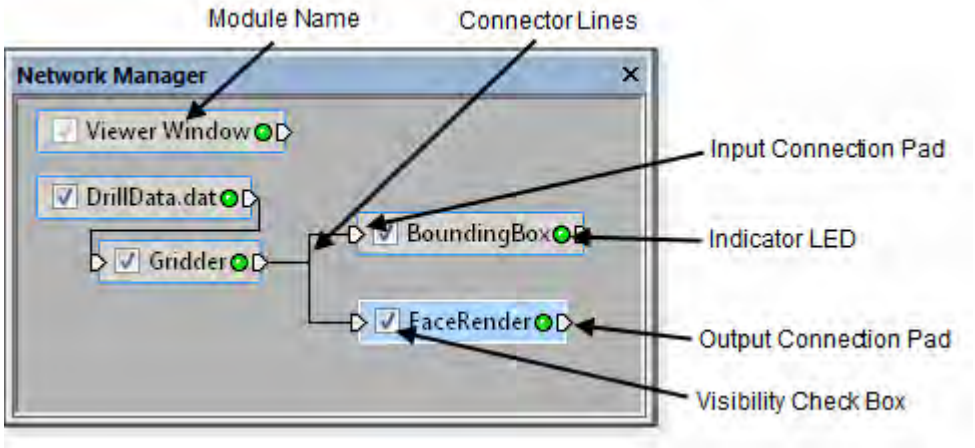
Customize the axis triad colors on the **Colors** page of the **Options** dialog. Choose the **Tools | Options** command to open the **Options** dialog.

Chapter 4 - Data Source Modules

Introduction to Modules

A module is a data set or a process to be applied to a data set or process. Modules are the building blocks from which the final **Voxler** output is constructed. Modules accept data on their input connection pads ▷, modify the data, and pass it along through the output connection pad ▷.

Modules are displayed in the **Network Manager**. You can connect and disconnect modules to create a visualization network representing the flow of data. Modules need to be connected in order to generate an output. The current geometry output of the modules in the **Network Manager** are displayed in a three-dimensional view in the **Viewer** window. If the data is not connected to a graphics output module, nothing is displayed in the **Viewer** window.



*The **Network Manager** displays the visualization network, which includes all loaded modules and their connections.*

View All Modules

All modules are accessed using the Network menu commands. Alternatively, all modules are accessed in the **Module Manager** when the *Show all modules* button is selected .

View Applicable Modules

When a module is selected in the **Network Manager** and the *Show all modules* button  is not selected in the **Module Manager**, available modules that can be connected to the selected module output port are displayed in the **Module Manager**. Alternatively, right-click a module in the **Network Manager** to display only the applicable modules in the context menu.

On the module description pages, there are **Inputs** and **Outputs** sections to discuss the type of input and output modules that each module are compatible.

Import

Click **File | Import** to open the **Import** dialog. The type of data determines what type of module can be attached to it. **Voxler** supports several different data types. See the *File Format Chart* for a detailed list of supported file formats.

Viewer Window

The **Viewer Window** is a unique module that is automatically created when a new instance of **Voxler** is generated. The *Viewer Window* module appears in the **Network Manager**. The *Viewer Window* module cannot be deleted. The purpose of the *Viewer Window* module is to control the properties of the **Viewer** window for the current instance of **Voxler**. To change the **Viewer** window properties for future instances of **Voxler**, choose the **Tools | Options** command and adjust properties on the **Colors** page of the **Options** dialog.

Module Types

There are four types of modules: *computational*, *data source*, *general*, and *graphics output.* Each module type is discussed below.

Computational Modules

Computational modules alter the data by changing the data type, filtering, creating a gradient, gridding, performing mathematical transformations, merging, resampling, slicing, creating a subset, or transforming coordinates. Click on one of the following computational modules for detailed information on using the module and module properties.

ChangeType

DuplicateFilter

ExclusionFilter

ExtractPoints

Filter

Gradient

Gridder

Math

Merge

Resample

Slice

Subset

Transform

Data Source Modules

Data source modules serve as a source of raw data. The data can be imported or created from mathematical functions. Click on one of the following data source modules for detailed information on using the module and module properties.

Import (*Data Source*, *Point Source*, *Lattice Source*, *Geometry Source*)

FunctionLattice

TestLattice

WellData (combines multiple *Data Source* modules into a single output)

General Modules

General modules display module information and provide custom lighting in the **Viewer** window. Click on one of the following general modules for detailed information on using the module and module properties.

Info

Light

Graphics Output

Graphics output modules create graphics in the **Viewer** window. Typically, these modules require data input. Click on one of the following graphics output modules for detailed information on using the module and module properties.

Annotation

Axes

BoundingBox

ClipPlane

Contours

FaceRender

HeightField

Isosurface

ObliqueImage

OrthoImage

ScatterPlot

StreamLines

Text

VectorPlot

VolRender

WellRender

Data Types

In **Voxler**, data are imported and connected to various modules to accomplish a useful task such as displaying scatter plots, isosurfaces, heightfields, and image slices. The type of data determines what type of module can be attached to it. **Voxler** supports several different data types. See the *File Format Chart* for a detailed list of supported file formats.

There are four main types of data: *point sets*, *well data*, *lattices*, and *geometry*. Each are discussed in detail below.

Point Sets

Point sets contain one or more three-dimensional point locations. Each location has an X, Y, and Z coordinate along with optional data components. A point can have any number of components. A component is a data variable associated with each point. Usually, this variable is the data you are trying to visualize, such as concentration.

Point data are collections of XYZ points in space, optionally with associated data values. Occasionally, this is called "XYZC data" where XYZ represent the three-dimensional position and C represents one or more data values at that position.

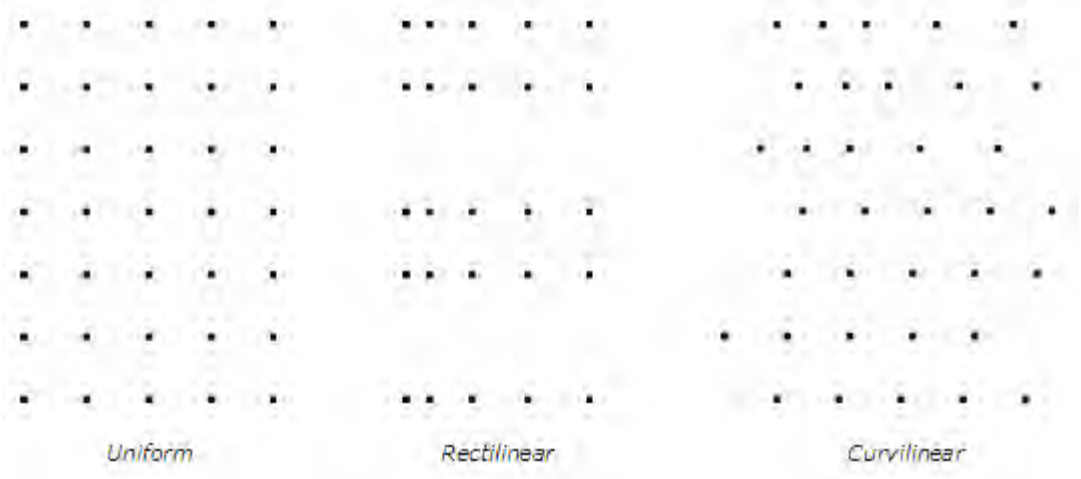
Well Data

The *WellData* module is a container for well data imported into the project. Each well is imported into the *WellData* module separately using the **File | Import** command. A *Well Data* module can have any number of wells, with each well containing information specific to that well. Each well can contain *X*, *Y*, *Z*, *MD* (Measured Depth), *Azimuth*, *Inclination*, and any number of *Log*s. The log is the data variable associated with the downhole location, and is usually the variable to be modeled.

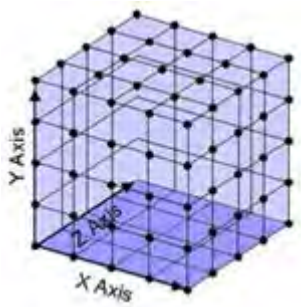
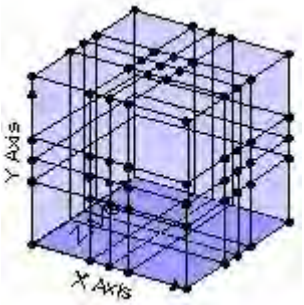
Lattices

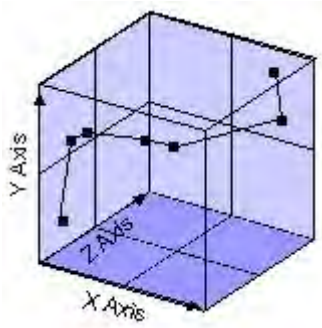
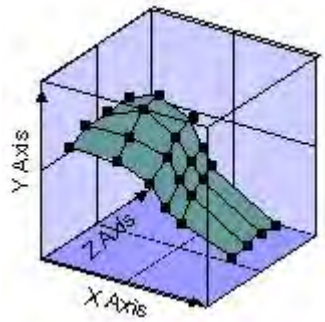
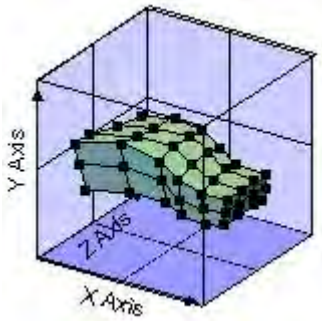
A lattice consists of a one-, two-, or three-dimensional data array. An array is a regular, structured matrix of points. A one-dimensional lattice is a line of data. A two-dimensional lattice is similar to an image or a **Surfer** grid. A three-dimensional lattice defines a three-dimensional volume. Each node (or point) in the lattice can contain one or more components or data values. Lattices are further categorized by the node geometry: uniform, rectilinear, and curvilinear. A single component lattice has a single data value (component) associated with each node. A multiple component lattice has two or more components per node.

Lattices can be further categorized by the geometry of the nodes:



The above diagrams are in two dimensions, but the actual lattice may be one-dimensional, two-dimensional, or three-dimensional.

Uniform	Spacing between nodes is fixed in each direction, though the spacing in each of the three directions may be different. Uniform lattice axes are orthogonal. An image or bitmap is an example two-dimensional uniform lattice. Most medical data sets produced from an MRI or CT machine are three-dimensional uniform lattices.	 This is an example of a uniform lattice.
Rectilinear	Spacing is variable, but the axes are orthogonal (like log-log graph paper). The coordinates are explicitly stored for the lattice planes in the X, Y, and Z directions and must be monotonically increasing.	 This is an example of a rectilinear lattice.

Curvilinear	Spacing is variable, and the lattice axes are non-orthogonal. A curvilinear lattice has regular topology (like all lattices) but irregular geometry. The coordinates of each node are set independently. The lattice may be warped into any configuration in which the cells do not overlap (or self-intersect). Curvilinear lattices are often used in fluid dynamics. Since each node has its own set of three-dimensional coordinates, curvilinear lattices are the most general. They are also the most inefficient to work with, requiring more memory and time.	 This is an example of a one-dimensional curvilinear lattice.  This is an example of a two-dimensional curvilinear lattice.  This is an example of a three-dimensional curvilinear lattice.

Lattices contain a data tuple at each node. A tuple consists of one or more data values (components). The number of components per tuple is fixed for any given lattice. Components are specified with a single primitive data type. These primitive types trade storage requirements for accuracy and range. The size specified in the table is the size to store a single value (in bytes).

Primitive Type	Size	Minimum	Maximum
Signed 8 bits	1	-128	127
Unsigned 8 bits	1	0	255
Signed 16 bits	2	-32768	32767

Unsigned 16 bits	2	0	65535
Signed 32 bits	4	-2147483648	2147483647
Unsigned 32 bits	4	0	4294967295
Signed 64 bits	8	~ -9.223372e18	~ 9.223372e18
Float 32 bits	4	~ -3.402823e+38	~ 3.402823e+38
Double 64 bits	8	~ - 1.797693e+308	~ 1.797693e+308

- Unsigned 8-bit data is often called "byte" data.
- Lattice nodes are referenced by an index tuple (i.j.k.), "i" being the index in the X direction, " j" being the index in the Y direction, and " k" being the index in the Z direction.
- Color data are stored within a lattice as RGBA byte values.
- The data are stored in a single memory block with the X direction varying fastest, followed by the Y direction, then followed by the Z direction. For example, assume a lattice with X, Y, Z dimensions of 2 x 3 x 2, two components per tuple, using BYTE primitives. The data block would have the following layout in memory, using the terminology (ZYXC):

0000, 0001, 0010, 0011,

0100, 0101, 0110, 0111,

0200, 0201, 0210, 0211,

1000, 1001, 1010, 1011,

1100, 1101, 1110, 1111,

1200, 1201, 1210, 1211.

The first byte is at the origin. The data block would be 24 bytes in size.

Geometry

Geometry consists of triangles, texture maps, line segments, and other objects. Geometry is collected at the end of the network and displayed in the **Viewer** window. Geometry can be added from a data file, such as a DXF, SHP, BLN, or IV file, or can be created with a module, such as a *Contours* or *Isosurface*.

Blanked Data

Blanking is fully supported in the point sets and lattices. Blanked data in **Voxler** is not represented by a special "flag" value. This allows the full range of a given primitive type to be used without being misrepresented as a blanked node. For example, if unsigned byte data were being stored in a lattice, values from 0 to 255 could be stored in the lattice without needing to reserve a special blanked value. This is all handled internally by **Voxler**.

Lattice nodes are commonly blanked in the *Gridder module* when the search criteria are not met; in *Filter* edge handling; and when a smaller input lattice is resampled to a larger output lattice (in which nodes outside the input lattice extents are blanked).

Blanked nodes are not displayed in output modules; however, they are written to exported data files if the export format supports them. For example, if a **Surfer** . GRD file is exported, the internal blank nodes within **Voxler** are converted to **Surfer** grid blank values in the output file.

Source Modules

Some source modules— including *FunctionLattice*, *TestLattice*, and *WellData* — can be loaded from the **Module Manager**. Most source modules, however, do not appear directly in the menus; rather, they are automatically created when a data set is loaded with the **File | Import** command.

Source modules serve as a source for the original data and are usually named the same as the imported data file. Source modules load geometry, lattices, and point data into the scene.

Inputs

Source modules have no input. To change the data displayed delete the module from the **Network Manager**. Choose **File | Import** to select a new source module. Using the **File | Import** command can create geometry, points, or a lattice.

Outputs

Depending on the type of data imported, the source module may be connected to *Graphics Output Modules* or *Computational Modules.* An *Info Module* may also be connected to the output node.

Properties

When you import a data file, its corresponding source module appears in the **Network Manager**. Click the module to view its properties in the **Property Manager**.

Reloading a Source Module File

In the **Property Manager**, next to the *File path* selection, click the  button and the **Open** dialog appears. The data file name appears in the *File name* box of the dialog. The directory where the original data file was located is automatically shown. Click the *Open* button to close the dialog and reload the file from the saved copy. Reloading a file is useful if the contents of the file have changed and you want to update the network.

Changing a Source Module File

In the **Property Manager**, next to the *File path* selection, click the  button and the **Open** dialog appears. Select a different file to change the input file completely. You must select a new file that contains the same type of data as the original file. If the same type of data does not exist in the new file, the original file remains loaded in the **Network Manager** and a warning message will occur.

Import

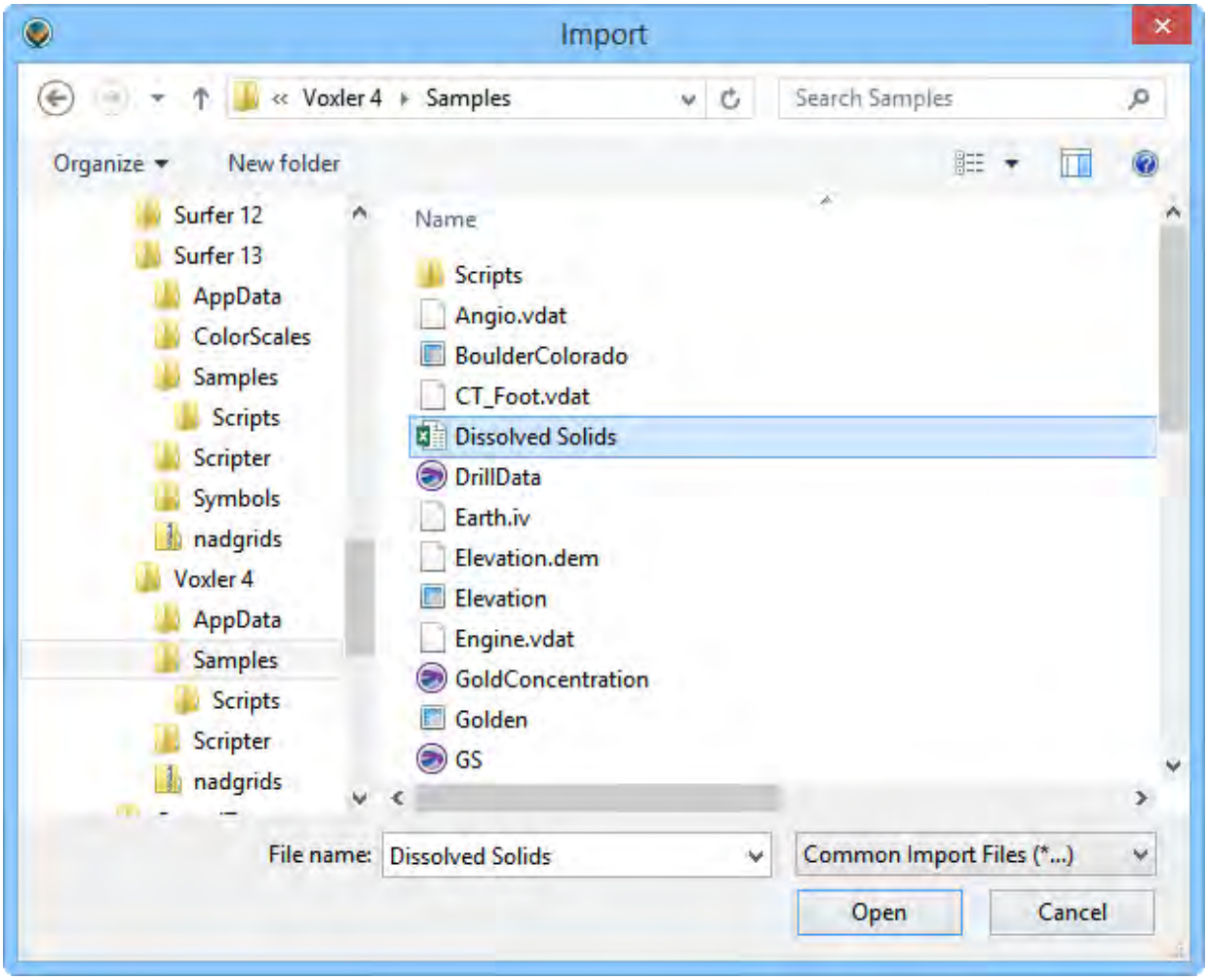
Click the **File | Import** command, click the  button, right-click in the **Network Manager** and choose **Import**, or double-click *Import* in the **Module Manager** to load a data file into a data source module. Depending on the format of the data, one or more format-specific dialogs may appear. The loaded data displays as a new icon in the **Network Manager**. It can then be connected to other modules in the network.

When this command is selected, the **Import** dialog displays, allowing one or more files to be selected and loaded. Each data set from each file is loaded into its own source module. Multiple

data files can be loaded at once by pressing the SHIFT or CTRL keys while selecting files in the dialog.

The Import Dialog

The **File | Import** command opens the **Import** dialog.



*Specify files to import using the **Import** dialog.*

The **Import** dialog has the following options available:

Look In

The *Look in* field shows the current directory. Click the down arrow to see the directory structure; click the folders to change directories.

Creating New Folders and Changing the View

The buttons to the right of the *Look in* field allow you to create new folders and change the view of the file list.

File List

The *File list* displays files in the current directory, which is listed in the *Look in* field. The *File of type* field controls the display of the file list. For example, if *Voxler Project Files (*.voxb)* is listed in the *Files of type* field, only *.VOXB files appear in the files list.

File Name

The *File name* field shows the name of the selected file. Type a path and file name into the box to open a specific file.

Specify a File Type

The *Files of type* field controls the display of the file list. For example, if *Voxler Project Files (*.VOXB)* is listed in the *Files of type* field, only .VOXB files appear in the file list. To see all files in the directory, choose *All Files (*.*)* from the *Files of type* list. Double-click a file to open it, or click the file once and then click the *Open* button.

The *Files of type* field shows the file format to be imported. To change the file format, click the down arrow and select the file type from the list. *All Files (*.*)* displays all files in a directory.

Open or Cancel

Click *Open* to close the **Open** dialog and import the specified file. Click *Cancel* to close the dialog without opening a new file.

Data Import Options Dialog

The **Data Import Options** dialog appears only once, even if more than one file is selected. Any changes made in this dialog are automatically applied to all files. If you need to specify different import options for one or more files, you must load them individually.

File Formats

Voxler automatically attempts to determine the selected file formats. The format of the imported file(s) is determined as follows:

1. If a particular format is specified in the *Files of type* list of the **Import** dialog, that format is used.
2. If that fails, then a match is made by using the file extension.
3. If the format still cannot be determined, the **Select Format** dialog displays and you are prompted to select the format from a list of supported formats.

Data Types

Voxler supports several different types of data:

- Tabular data— such as worksheets, comma-separated variable (CSV), and data files— are loaded into *Data Source* modules. The data must be organized by columns. For example, one column corresponds to the X coordinate, another to the Y coordinate, etc. You can then specify if the data file contains point or well data in the **Property Manager**. The columns that contain the X, Y, and Z coordinates along with the columns containing the data at those coordinates are all also specified in the **Property Manager**.
- Log ASCII Standard .LAS files are loaded into a *WellData* module. Each of the variables in the .LAS file are imported into **Voxler** as a log component.
- Data files that are not in a tabular file format are loaded into a *point source* module.

- Images and bitmaps are loaded as two-dimensional lattices. Grayscale images are converted to 1-byte scalar lattices. Color images are converted to four- component RGBA lattices.
- Native three-dimensional lattice data in various proprietary and public formats are supported, including .RAW format.
- One-dimensional curvilinear lattices are automatically converted to point sets during import.
- Vector formats.

Loading Multiple Images

Image stacks are loaded as a three-dimensional lattice. These are usually a series of planar image slices that are meant to be stacked one on top of another to form a three-dimensional volume. The individual slices are often in a standard two-dimensional format like .TIFF, .JPEG, or .DICOM. This type of data is commonly created by medical imaging equipment.

If all of the images are selected at once and the sizes are identical, **Voxler** attempts to combine them into a single three-dimensional lattice. The slices are combined in alphabetical order according to the file names. The stacking starts at the bottom (Z=1) and proceeds upward. Simultaneous selection is most easily achieved by first clicking the first slice and then holding down the SHIFT key and clicking the last slice. The resulting source module takes the name of the first loaded image. The name of the last file is listed next to *File Path* in the **Property Manager**.

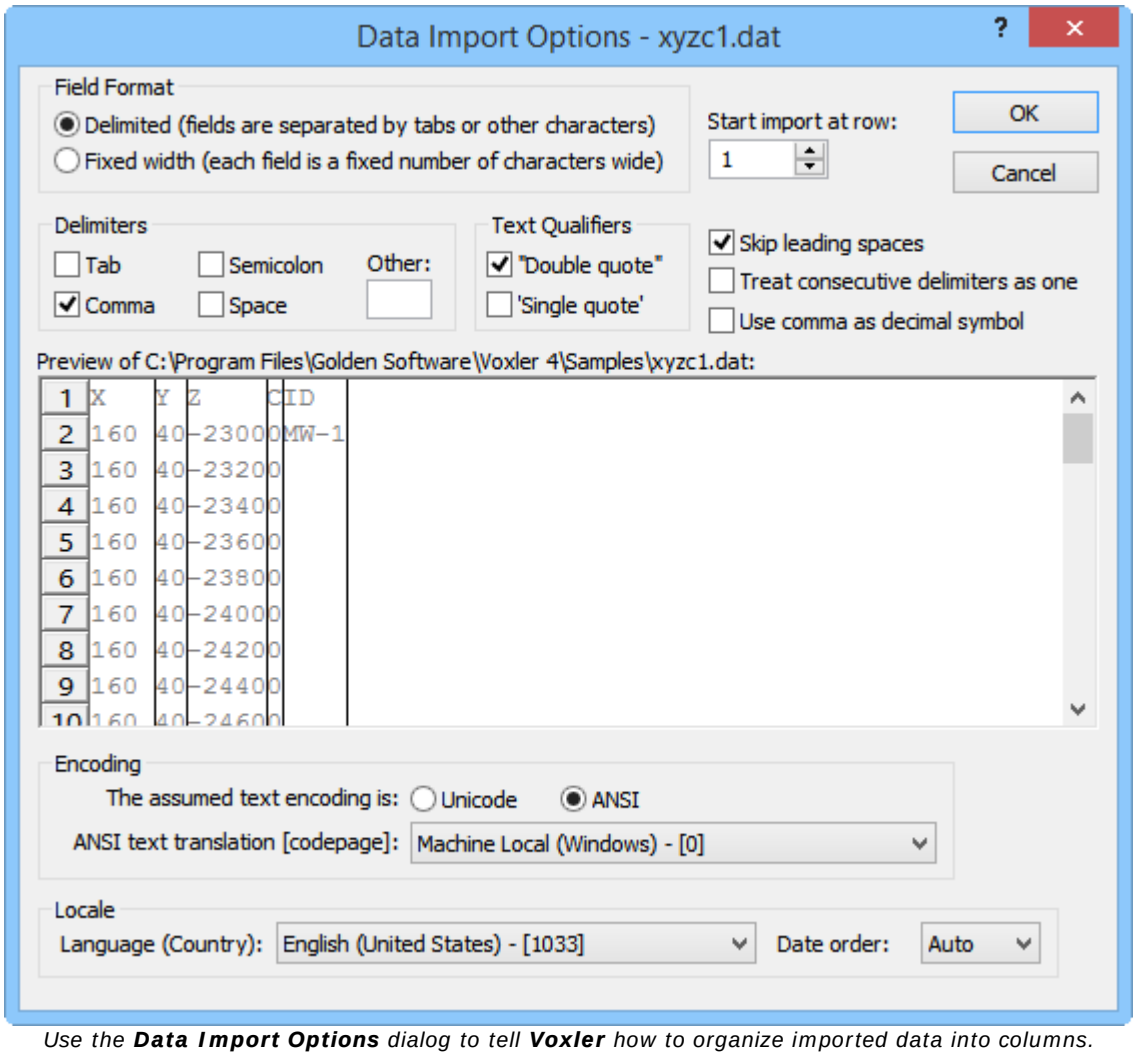
Image Coordinates

When loading image bitmap files with coordinates, the coordinate information comes from a geotiff (internal information) or external files.

Data Import Options Dialog

If a file is in an ASCII text format with an unrecognized file extension, the **Data Import Options** dialog appears when opening the file.

The **Data Import Options** dialog is used to organize tabular data into point databases organized by columns. The dialog appears when the **File | Import** command is used when importing tabular data from delimited text files (i.e. .TXT). The **Data Import Options** dialog is also displayed when using the worksheet **Import** command and when the *Show Import Options* box is checked in the **Paste Special** dialog. These file formats are assumed to have one record per line in which each record contains a fixed number of numeric data fields.



*Use the **Data Import Options** dialog to tell **Voxler** how to organize imported data into columns.*

Field Format

Specify the format of the input fields in the *Field Format* group. The *Field Format* group controls allow you to specify whether the fields in each record are of fixed width or are separated by delimiters. The options are *Delimited* or *Fixed Width*.

Delimited

Choose *Delimited (fields are separated by tabs or other characters)* if the imported data uses *Delimiters* (tab, semicolon, comma, space, other) to separate data fields.

Fixed Width

Choose *Fixed width (each field is a fixed number of characters wide)* if the imported data uses a fixed width to separate data fields.

Start Import at Row

Use the arrow buttons or type in a row number at which to start the data import in the *Start import at row* box. For example, a value of one will start the data import at row one of the data set. A value of five will start the data import at row five of the data set.

Delimiters

The *Delimiters* group is used to specify how the fields are separated if *Delimited (fields are separated by tabs or other characters)* is the selected *Field Format*. *Delimiters* controls allow you to specify what characters act as delimiters between fields in a record. Choose the desired delimiters to be used during the import process by checking the box next to *Tab*, *Comma*, *Semicolon*, or *Space*. You may also enter a custom delimiter in the *Other* box. More than one delimiter may be checked.

Text Qualifiers

Specify *"Double Quote"* or '*Single Quote*' in the *Text Qualifiers* group to indicate the correct qualifier to identify text values in the data file.

Double Quote

Check the box next to *"Double Quote"* or *'Single Quote'* to specify that everything between those marks should be interpreted as a single value, and any delimiter characters between any two quote characters are not treated as a delimiter.

For example, if *Space* is chosen as the delimiter and *Double Quote* is chosen as the text qualifier, the string "Aspen Park" is treated as a single data value due to the double quotes surrounding it, and the space delimiter between the words is treated as part of the value.

Single Quote

Check the box next to *'Single Quote'* to specify that everything between those marks should be interpreted as a single value, and any delimiter characters between any two quote characters are not treated as a delimiter.

For example, if *Space* is chosen as the delimiter and *Single Quote* is chosen as the text qualifier, the string 'Aspen Park' is treated as a single data value due to the single quotes surrounding it, and the space delimiter between the words is treated as part of the value.

Skip Leading Spaces

Check the box next to *Skip leading spaces* to tell the software to ignore spaces that appear before initial text (not treat spaces at the beginning of the line as delimiters).

Treat Consecutive Delimiters as One

Check the box next to *Treat consecutive delimiters as one* to instruct the software to interpret any consecutive delimiters into a single delimiter rather than breaking to a new column for each consecutive delimiter.

Check the *Treat consecutive delimiters as one* box to treat any sequence of more than one consecutive delimiter as if it were one delimiter. If this box is unchecked, each delimiter character marks a new field, i.e., consecutive delimiters act to define "empty" fields.

Use Comma as Decimal Symbol

Check the box next to *Use comma as decimal symbol* if commas are used to as the divisor symbol between whole numbers and fractions. If your data uses commas as the decimal symbol, it is highly recommend that you use some other character for the *Delimiters*. This option is more common in locations where the thousands separator is the period (.) and the comma is used as the decimal.

Preview

The parsed data are shown in the *Preview* section. The *Preview* field displays a preview of how the text in the import field is divided into fields to help you set the other dialog controls appropriately.

Encoding

The *Encoding* section allows the choice of *Unicode* data or *ANSI* data when importing or opening an ASCII data file. Unicode data is often referred to as international data. It would include character sets from Russia, Israel, China, Greece, Hungary, among others. After selecting *Unicode*, select the *ANSI text translation [codepage]* option that will read the data correctly. If the data does not appear correctly in the *Preview* window, the *Encoding* may be specified incorrectly.

ANSI encoding contains characters within the first 256 characters of a font. These are normally in English.

Locale

The locale section of the dialog contains options for determining date/time values. The *Language (Country)* setting determines which month names are interpreted as part of a date. For example, if *German (Germany)* is selected, "Oktober" will be recognized as a valid month name. English month names are always recognized as valid month names. The default *Language (Country)* is determined by the user locale set in the Windows Control Panel. To change the *Language (Country)*, click the current option and select a language from the list.

The *Date order* option specifies the order in which dates are written in the data file. The date 02/03/04 is ambiguous and could be Month-Day-Year, Day-Month-Year, Year-Month-Day, etc. The *Date order* option ensures dates in the data file are imported correctly into **Voxler**. The default *Date order* is *Auto*. The standard date order for the *Language (Country)* setting is used when *Date order* is set to *Auto*. Specify the *Date order* for the data file import by clicking the current *Date order* selection and then selecting the desired *Date order* from the list. All six combinations of Day (D), Month (M), and Year (Y) are included in the *Date order* list.

OK

Click the *OK* button to proceed with the import process.

Cancel

Click the *Cancel* button to close the dialog without importing the data set.

Data Source Module

Data Source modules are created when an ASCII data file, Excel data file, database file, or other type of tabular data file is loaded into the **Network Manager** using the **File | Import** command.

Input Modules

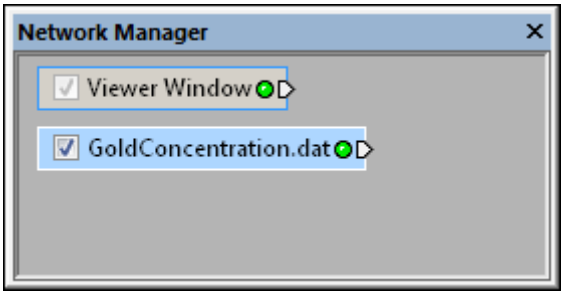
Defined from the point coordinates located in the data file.

Output Modules

Depending on the type of data imported, the *Data Source* module may be connected to the *Graphics Output Modules*, *Computational Modules*, or *WellData Module.* An *Info Module* may also be connected to the output node.

Properties

Click on the *Data Source* module in the **Network Manager** to select it. The **Property Manager** displays various data column specification properties for the data source module. The data source module properties are described below.



*Select a data set in the **Network Manager** to display its properties in the **Property Manager**.*

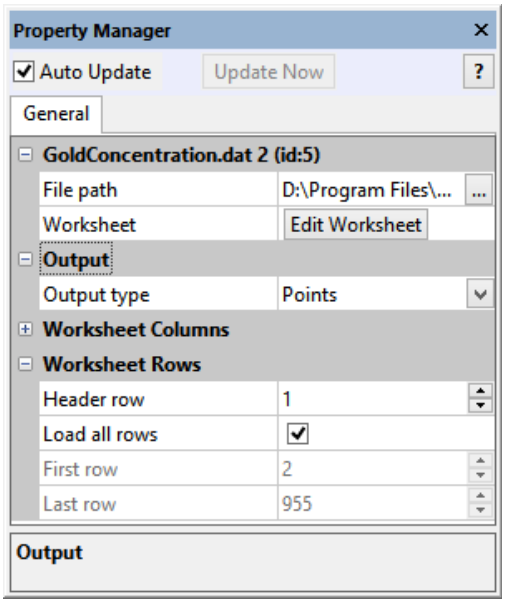
The *Data Source* module contains the following tabs in the **Property Manager:**

General

The **General** page properties change depending on the *Output type* selection. The three sections below explain the shared properties, *Points* properties, and *Wells* properties respectively.

General Options

The following options are presented in the **General** page regardless of the *Output type* selection.



The properties visible above are displayed regardless of the Output type selection.

File Path

The path of the loaded source file appears next to *File path.* Click the  button to the right of the file path to display the **Open** dialog. The **Open** dialog appears. The currently loaded file name is display in the *File name* section. The path for the current file is shown. Click the *Open* button to reload the current file.

Alternatively browse and select a new file. Click the *Open* button to load the new file. The dialog closes and the selected file is loaded. If the new file does not display geometry information, a warning message will appear and the original file remains.

Worksheet

Click the *Edit Worksheet* button to display and edit the data in a linked worksheet window. The title bar and worksheet tab will display "Linked to:" followed by the module name and ID. Changes made to the module data will be automatically saved and represented instantly in the viewer window.

Notice that the data linked to the module is being edited. The original data file, i.e. the file that was imported, is not changed. The data is saved in the **Voxler** project .VOXB file. If you wish to save the changes to a separate file (or to overwrite the original data file), use the **File | Save Copy As** command while viewing the linked worksheet.

Output

The *Output type* option specifies whether the *Data Source* module contains *Points* or *Wells* information.

Worksheet Rows

The *Worksheet Rows* section contains options for specifying which rows are used.

Specify the row containing the column titles (i.e. the header row) in the *Header row* field. If the *Header row* is incorrectly specified, the column title will not be displayed next to the column letter in the other **General** page property fields or in the **Property Manager** for downstream modules.

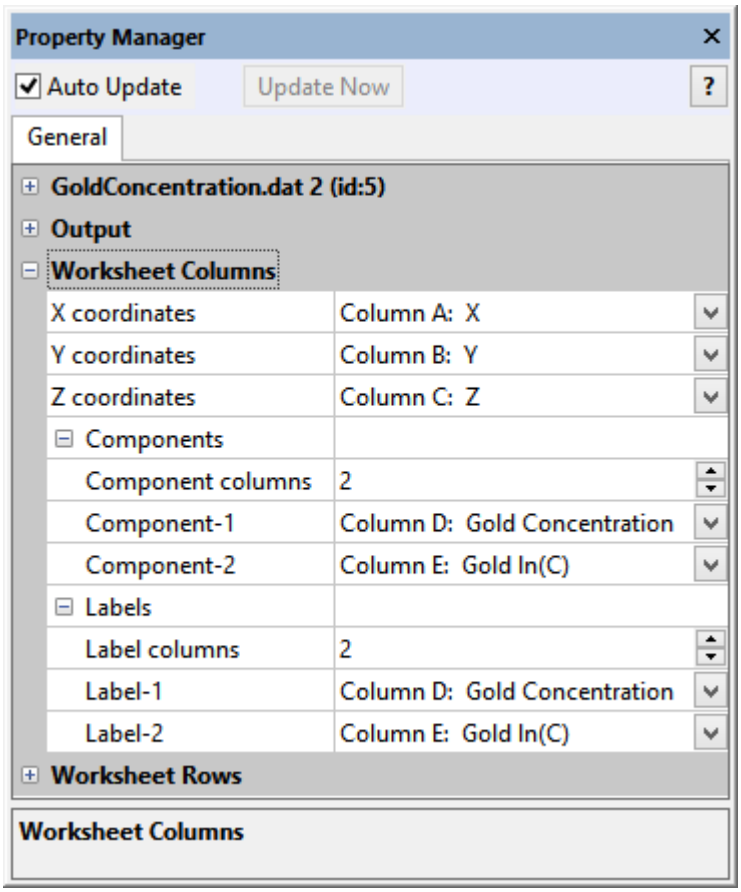
Type the row number into the *Header row* field, or click the  buttons to change the value. The header row value can be any integer between 1 and 1000.

Check the *Load all rows* check box to use all rows in the worksheet, excluding the header row if applicable. Uncheck the *Load all rows* check box to set the first and last row used in the data file.

Set the *First row* and *Last row* properties by typing a value into the field or clicking the  buttons.

General Page - Points

The following options are displayed when the *Output type* is set to *Points*.



*Specify which information is located in the worksheet on the **General** page.*

Worksheet Columns

The *Worksheet Columns* section of the **General** page contains options for specifying where information is located in the worksheet. The *Worksheet Columns* section is displayed when the *Output type* selection is *Points*.

Automatic Column Assignment

Voxler assigns the data columns automatically upon import. For *Points* data the *X coordinates* are assigned to the first column, *Y coordinates* are assigned to the second column, *Z coordinates* are assigned to the third column, and the remaining columns are assigned as component columns from left to right. It is unnecessary to reformat your data files in this order, as the coordinate and component columns can be manually specified in the **Property Manager**.

X, Y, and Z Coordinates

Select the column containing the X, Y, and Z coordinates in the *X coordinates*, *Y coordinates*, and *Z coordinates* fields respectively. Clicking the current selection in each of the fields opens a list of columns which contain data. If the data file contains a header row, the header for the column will

be displayed after the column letter. For example with the GoldConcentration.dat sample file, *Column A: X* is one of the columns displayed in the list.

Components

The *Components* section contains options for specifying the number of components and in which column(s) the component data is located.

Specify the number of component columns by typing a value or clicking the  buttons in the *Component columns* field. The *Component columns* value is limited by the number of columns in the data file.

Select the appropriate column for each component in the *Component-1*, *Component-2*, etc. fields. If the data file contains a header row, the header for the column will be displayed after the column letter. For example with the GoldConcentration.dat sample file, *Column D: Gold Concentration* is one of the columns displayed in the list. The header information is also passed to downstream modules, which makes selecting the desired component or coordinate column easier.

Labels

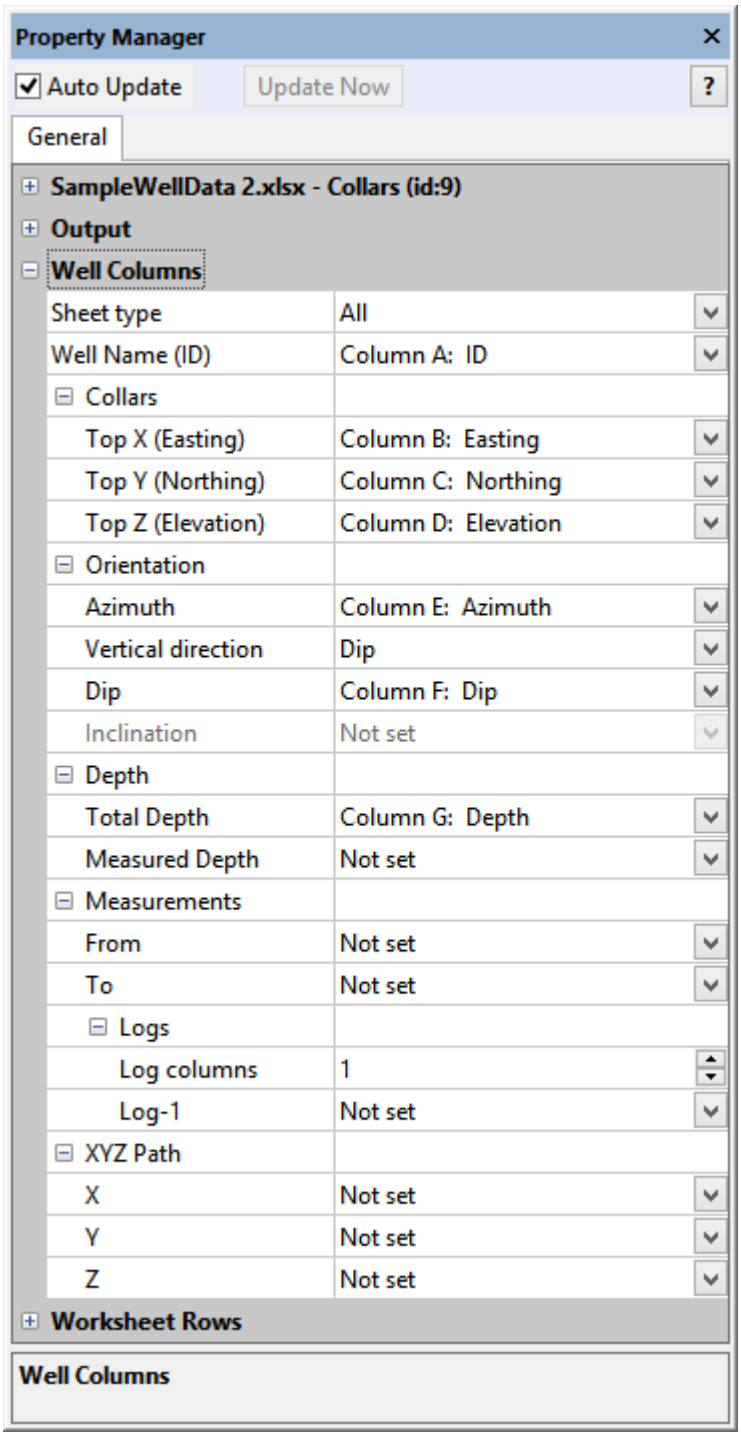
The *Labels* section contains options for specifying the number of label columns and which columns will be used for labels.

Specify the number of label columns by typing a value or clicking the  buttons in the *Label columns* field. The *Label columns* value is limited by the number of columns in the data file.

Select the appropriate column for each component in the *Label-1*, *Label-2*, etc. fields. If the data file contains a header row, the header for the column will be displayed after the column letter. The header information is also passed to downstream modules, which makes selecting the desired label column easier.

General Page - Wells

The following options are displayed when the *Output type* is set to *Wells*.



*Specify which well information is included in the worksheet on the **General** page.*

Well Columns

The *Well Columns* section of the **General** page contains options for specifying where information is located in the worksheet. The *Well Columns* section is displayed when the *Output type* selection is *Wells*.

Automatic Column Assignment

Voxler assigns the data columns automatically upon import. For *Wells* data the *Sheet type* property affects the automatic assignment. The automatic assignments are made in the column order, from left to right, specified by the table below. It is unnecessary to reformat your data files to match the following order, as the well data columns can be manually specified in the **Property Manager**.

Order	Collars	Directional Survey	From/To Logs	Logs/Curves	XYZ Path
1	Well Name (ID)	Well Name (ID)	Well Name (ID)	Well Name (ID)	Well Name (ID)
2	Top X (Easting)	Measured Depth	From	Measured Depth	Measured Depth
3	Top Y (Northing)	Azimuth	To	Log-1	X
4	Top Z (Elevation)	Inclination/Dip	Log-1	Log-2	Y
5	Azimuth		Log-2		Z
6	Dip/Inclination				
7	Total Depth				

Sheet Type

The *Sheet type* property simplifies the **General** page display by showing only the options pertaining to the selection. Select *Collars*, *Directional Survey*, *From/To Logs*, *Logs/Curves*, or *XYZ Path* from the *Sheet type* list if one of the options is appropriate for the data file. Select *All* in the *Sheet type* list to view all of the properties in the **General** page. The following table shows which properties are found under each *Sheet type* selection.

Sheet Type	Well Columns and Properties Displayed
Collars	Well Name (ID), Top X (Easting), Top Y (Northing), Top Z (Elevation), Azimuth, Vertical direction, Dip, Inclination, Total Depth
Directional Survey	Well Name (ID), Azimuth, Vertical direciton, Dip, Inclination, Measured Depth
From/To Logs	Well Name (ID), From, To, Log columns, Log-#
Logs/Curves	Well Name (ID), Measured Depth, Log columns, Log-#
XYZ Path	Well Name (ID), Measured Depth, Log columns, Log-#, X, Y, Z

Well Name (ID)

The column that contains the name assigned to a well. The well name is also referred to as *Well ID* or *Borehole Name*. The well name is a unique name assigned to each well. This name is used to connect data from multiple imports to the same well.

If the column containing the *Well Name (ID)* contains blanks, that well is not imported.

Collars

The Collars section contains the Top X (Easting), Top Y (Northing), and Top Z (Elevation) column options.

- The *Top X (Easting)* column is the column that contains the physical X (easting) location of the collar.
- The *Top Y (Northing)* column is the column that contains the physical Y (northing) location of the collar.
- The *Top Z (Elevation)* column is the column that contains the physical Z (elevation) location of the collar.

Orientation

The *Orientation* section contains the *Azimuth*, *Dip*, and *Inclination* column options and the *Vertical direction* property.

- The *Azimuth* column is the column that indicates the angle of the well bore direction, as projected on a horizontal plane and relative to true north. Values are in degrees. By industry convention, zero is north, and the angle increases clockwise.
- The *Vertical direction* property specifies whether the data file uses a *Dip* column or *Inclination* column.
- The *Dip* column is the column that indicates the deviation from horizontal of the well path. This value is in degrees and is between +90 and -90. +90 is completely vertical and pointing up. Zero is completely horizontal. -90 is completely vertical and pointing down. This is used in conjunction with *Azimuth*.
- The *Inclination* column the column that indicates the vertical angle of the well bore direction. Values are in degrees. By industry standard, zero degrees is vertical (directly down) and 90 degrees is horizontal. An inclination greater than 90 degrees would indicate drilling upward.

Depth

The *Depth* section contains the *Total Depth* and *Measured Depth* column options.

- The *Total Depth* column is the column that contains the true ending depth of the well.
- The *Measured Depth* column is the column that contains the true measured depth along the log.

Measurements

The *Measurements* section contains the *From*, *To*, and *Log-#* column options and the *Log columns* property.

- The *From* column is the column that contains the starting depth of an interval measurement for a log. This is used in connection with the *To* column when importing interval type well logs.

- The *To* column is the the column that contains the ending depth of an interval measurement for a log. This is used in connection with the *From* column when importing interval type well logs.
- The *Log columns* property specifies the number of log columns in the data file. Type the number of log columns into the *Log columns* field or click the  buttons to adjust the *Log columns* value. The *Log columns* value is limited by the number of columns in the data file.
- A *Log-1*, *Log-2*, etc. column is any column that contains values to be displayed on the well. These could be porosity, permeability, gamma, resistivity, or any other measured value along the depth of the log. Normally, these values are used to display along the log. When converted to data points, these *Log* values can be gridded and displayed as an *Isosurface, VolRender*, or *FaceRender*. If the data file contains a header row, the header for the column will be displayed after the column letter. The header information is also passed to downstream modules, which makes selecting the desired label column easier.

XYZ Path

The *XYZ Path* section contains the *X*, *Y*, and *Z* column options.

- The *X* column is the column that contains the X (easting) location of each point being imported along the well trace.
- The *Y* column is the column that contains the Y (northing) location of each point being imported along the well trace.
- The *Z* column is the column that contains the Z (elevation) location of each point being imported along the well trace.

WellData Module

The **File | Import** or **Network | Well | WellData** command adds a *WellData* module to the **Network Manager**.

The *WellData* module is a container for the well data imported into the project. Wells are displayed with a *WellRender* module. Multiple wells can be included in a single *WellData* module. Each well can have multiple logs, collar information, trajectory information, and formation directional information.

Inputs

The *WellData* inputs are well trajectory and log data files and/or *Data Source* modules. To import well log data, for example with a LAS Log ASCII Standard file, into a *WellData* module, use the **Import** command. When the well data is in a tabular file format, import the data file and then connect the *Data Source* output to one of the *WellData* module inputs.

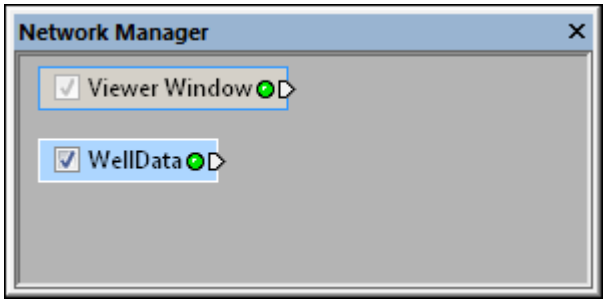
New data can be added to an existing *WellData* module with the **File | Import** command or by connecting multiple *Data Source* modules to the *WellData* module inputs.

Outputs

The *WellData* is a container for information from multiple wells. It may be connected to the *ExtractPoints* or *Transform* computational module types. It may also be connected to the *Axes*, *BoundingBox*, or *WellRender* graphics output module types. An *Info Module* may also be connected to the output node.

Properties

Click on the *WellData* module in the **Network Manager** to select it. The *WellData* properties are displayed in the **Property Manager**. The available properties are described below.



*Select the WellData module in the **Network Manager** to display its properties in the **Property Manager**.*

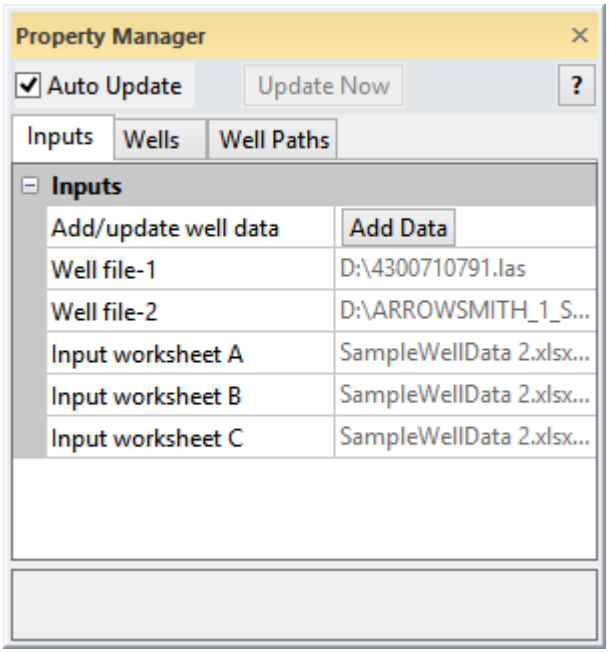
The *WellData* module contains the following tabs in the **Property Manager:**

Inputs

Wells

Well Paths

Inputs Options



Well data modules contain a list of all the wells imported into the module. Notice both well files and worksheet data can be imported into the WellData module.

Adding Additional Wells

Click the *Add Data* button to add additional wells to the existing *WellData* module. In the **Import** dialog, select the well data file and click *Open.* Select the appropriate options in the file type specific **Import Options** dialog, if one is displayed. The new or updated information is displayed in the **Property Manager**.

If the file type contains tabular data, a *Data Source* module will be created. Specify the data columns in the **Property Manager** for the *Data Source* module, and then connect the *Data Source* module to one of the *WellData* module inputs.

If the file type is not tabular, the well data well be imported directly into the *WellData* module.

Deleting an Input File

To delete a well data input, right-click on the input file and choose **Delete [file name]**. This action cannot be undone. If a well is accidentally deleted, the well can be added back to the *WellData* module by using the **File | Import** command and importing the original well data again.

Input Worksheet data cannot be deleted in the **Inputs** page. Instead, delete or disconnect the associated *Data Source* module in the Network Manager.

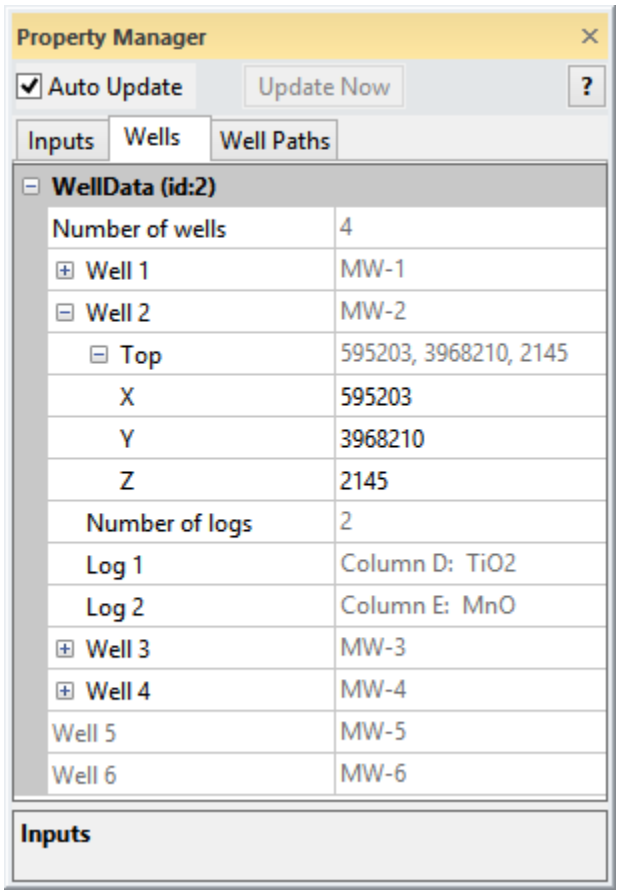
Displaying WellData

In order for well traces to be displayed in a *WellRender*, the *WellData* module needs to contain valid well paths. For a well to have a valid path, one of the following needs to be true:

1. The data is imported with azimuth, dip, and measured depth.
2. The data is imported with directions survey information. A path is then computed on the **Well Paths** tab.
3. The data is imported with pre-computed X, Y, and Z values down-the-hole.

WellData Module - Wells Page

The *WellData* module **Wells** page contains information about the imported wells. Wells can also be hidden or displayed, and well tops can be edited in the **Wells** page. To open the **Wells** page, click on the *WellData* module in the **Network Window**. In the **Property Manager**, click on the **Wells** tab.



*View information about imported wells in the **Wells** page of the **Property Manager**. Well 5 and Well 6 are hidden in this example image.*

Number of Wells

The *Number of wells* displays the number of wells that have been imported into the *WellData* module. To change the number of wells, use the **File | Import** command or the *Add Data* button on the **Inputs** page, and import new data into the existing *WellData* module.

Well

Each well is listed individually in the *Well* section. The well name is located next to the *Well* option. Click on the ⊞ to expand the well to see the properties for that well. Click the ⊟ to compress the well information. Available information about the well that may be included is *Top* and *Log* information. The additional information is not displayed when a well is hidden.

If you wish to change a well name, click the *Edit Worksheet* button on the **General** page for the input *Data Source* modules. Change the well name to the desired name in the worksheet. This

operation must be repeated for each *Data Source* module that includes data for the well. The well name will be updated automatically when you return to the viewer window.

Hiding a Well

To hide a well, right-click on the well number and click **Hide [well name]**. Click once outside of the **Property Manager** and the well will be hidden. Hidden wells are indicated on the **Wells** page by gray text. To show a hidden well, right-click on the hidden well and select **Show [well name]**. Click once outside of the **Property Manager**, and the previously hidden well will be displayed.

Top

The well *Top* can be edited. This is useful if the top was imported incorrectly or has been changed. To change the location of the well top, click on the ⊞ to expand the *Top* properties for that well. Highlight the existing value next to *X*, *Y*, or *Z* and type a new value to change the coordinates.

Number of Logs

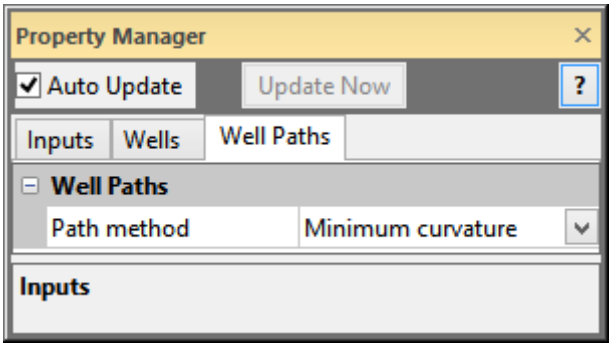
The *Number of logs* section displays how many logs were imported for each well.

Log Name

The Log 1, Log 2, Log 3, etc. lines display the name of the log imported for each well.

WellData Module - Well Paths Page

The *WellData* module **Well Paths** page contains the method for how to calculate the well paths. All wells in the *WellData* container have the same properties applied. To open the **Well Paths** page, click on the *WellData* module in the **Network Window**. In the **Property Manager**, click on the **Well Paths** tab.



Compute the well paths for all wells that have tops, azimuth, and inclination information.

Well Paths

The *Well Paths* section contains options for converting Measured Depth (MD), Azimuth, and Inclination information to X, Y, and Z data. Click on the ⊞ to expand the well path section to compute the paths for the wells. Click the ⊟ to compress the well path information. If the *WellData* module does not have Measured Depth, Azimuth, and Inclination data for at least one well, then the *Path method* is disabled, because no well paths are computed.

Path Method

The *Path method* contains various methods for computing the well paths for Measured Depth (MD), Azimuth, and Inclination well data. Available options are *Average angle*, *Balanced tangential*, *Minimum curvature*, and *Tangential*. To change the option, click on the existing method and select the desired method from the list. The most commonly used method is *Minimum curvature*.

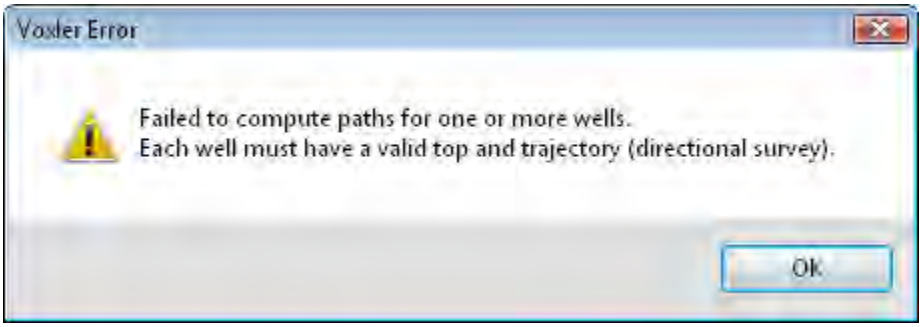
Each method creates a mathematical approximation of the true X, Y, and Z value along the well path. Each technique is used in different circumstances. The technical aspects of each method are found in:

"Bulletin on Directional Drilling Survey Calculation Methods and Terminology," American Petroleum Institute (API) Bulletin D20, December 31, 1985.

Warning Message

A warning message will appear in two instances:

- when opening a **Voxler** Project VOXB file with incomplete directional survey data
- when connecting the *WellData* module to a *Data Source* module that contains incomplete directional survey data



A warning appears if all wells do not have trajectory or top information.

Click *OK* on the warning. The wells that have valid top and trajectory information will have the paths computed. Some wells will not have computed paths. For these wells, you should examine and update the top and trajectory information for that well and reload.

How to Import and Display Wells

Voxler can import well data from numerous sources, including from LAS files. Typically the well data (logs) will be imported from one file and the physical location of the well (collars) will be imported from another file. Often, trajectory data indicating the direction of the well trace will also be imported, when the well is not assumed to be vertical.

Primary Data Requirements

In order for well traces to be displayed, the *WellData* module needs to contain valid well paths. For a well to have a valid path, one of the following needs to be true:

1. The data is imported with azimuth, dip, and measured depth.
2. The data is imported with directions survey information. A path is then computed automatically.
3. The data is imported with pre-computed X, Y, and Z values down-the-hole.

Log ASCII Standard LAS files and non-tabular format data files will be imported directly into the *WellData* module. When well data is in a tabular file format, a *Data Source* module will be created upon import.

Importing Well Collar Data

For this example, well collars, trajectory data, and log data are on three tabs of an Excel spreadsheet. Each tab is imported separately, with the following steps.

1. Click the **File | Import** command.
2. In the **Import** dialog, select the well collar file. For this example, select the *SampleWellData.xlsx* file from the Samples directory. By default, the Samples directory is located at C:\Program Files\Golden Software\Voxler 4\Samples. Click *Open*.
3. In the **XLSX Import Options** dialog, select the *Collars* table and click *OK*.
4. A Data Source module titled SampleWellData.xlsx - Collars is created.
5. Specify the following options in the **Property Manager**:
    a. Change the Output type to Wells
    b. Change the *Sheet type* to *Collars*. This step is not required. It simplifies the **Property Manager** display, showing only the properties applicable to a worksheet containing collar data.
    c. Set Well Name (ID) to Column A: ID
    d. Set Top X (Easting) to Column B: Easting
    e. Set Top Y (Northing) to Column C: Northing
    f. Set Top Z (Elevation) to Column D: Elevation
    g. Set Azimuth to Column E: Azimuth
    h. Set Vertical Direction to Dip
    i. Set Dip to Column F: Dip
    j. Set Total Depth to Column G: Depth
6. Click the **Network | Well | WellData** command to create a *WellData* module.
7. Verify that the *SampleWellData.xlsx - Collars* module output is connected to one of the *WellData* module inputs.

Importing Trajectory Data

At this point, all you have imported is the top location of the six wells. To import the well trajectories (traces), follow the steps in this section.

1. Click the **File | Import** command or click the *Add Data* button in the **Property Manager Inputs** page for the *WellData* module.
2. In the **Import** dialog, select the well file. Select the *SampleWellData.xlsx* file again and click *Open*.
3. In the **XLSX Import Options** dialog, select the *Trajectories* table and click *OK*.
4. A Data Source module titled SampleWellData.xlsx - Trajectories is created.
5. Specify the following options in the **Property Manager**:
    a. Change the Output type to Wells
    b. Change the *Sheet type* to *Directional Survey*. This step is not required, but it is recommended.
    c. Set Well Name (ID) to Column A: ID
    d. Set Azimuth to Column C: Azimuth
    e. Set Vertical direction to Inclination
    f. Set Inclination to Column D: Inclination
    g. Set Measured Depth to Column B: MD
6. Connect the *SampleWellData.xlsx - Trajectories* module output to one of the *WellData* module inputs.

The information is imported, and the actual X, Y, and Z values for the path of the well are calculated automatically by the program.

Displaying the Wells

At this point, we can display the wells by clicking the **Network | Graphic Output | WellRender** command. The well traces are then displayed.



*The initial well traces are displayed in the **Viewer** window.*

Importing Log Data

The well appearance can be altered by importing additional data.

1. Click on the *WellData* module in the **Network Manager**.
2. Click the **File | Import** command or click the *Add Data* button on the **Inputs** page of the **Property Manager**.
3. In the **Import** dialog, select the well file. Select the SampleWellData.xlsx again and click *Open*.
4. In the **XLSX Import Options** dialog, select the *Samples* table and click *OK*.
5. A Data Source module titled SampleWellData.xlsx - Samples is created.
6. Specify the following options in the **Property Manager**:
    a. Change the Output type to Wells
    b. Change the *Sheet type* to *From / To Logs*. This step is not required, but it is recommended.
    c. Set Well Name (ID) to Column A: ID
    d. Set From to Column B: From
    e. Set To to Column C: To
    f. Set Log columns to 2
    g. Set Log-1 to Column D: TiO2
    h. Set Log-2 to Column E: MnO
7. Connect the *SampleWellData.xlsx - Samples* module output to one of the *WellData* module inputs.

Displaying Log Data on the Wells

Once the log data is imported, the well appearance can be modified to display the logs using these steps:

1. Click on the *WellRender* module in the **Network Manager**.
2. Click on the Interval Data tab in the Property Manager.
3. Check the box next to *Show intervals* to add variable width log information.
4. Change the *Interval log* to MnO to use the MnO log *To Depth* and *From Depth* as the interval definition.
5. Change the *Color method* to *By log* so the colors of the intervals vary with log data.
6. Set the *Color log* to MnO.
7. Change the *Colormap* to *Rainbow* to display different colors along the length. The colors are determined by mapping the values in the MnO log to the colors in the *Rainbow* colormap.

8. Set the Size method to By log.
9. Set the *Size log* to TiO2. The width of the log is now determined by the data in the TiO2 log.



The wells are now displayed as tubes of variable width, based on the data in the TiO2 log, and color, based on the data in the MnO log.

Modifying Wells

Wells are imported into *WellData* modules. Once the wells exist in the *WellData* module, the wells can be displayed with a *WellRender* module or the data can be extracted with an *ExtractPoints* module. The below list a few of the features of the wells that can be changed.

Importing Wells

To import wells in a tabular format data file:

1. Click the **File | Import** command.
2. In the **Import** dialog, select the well data file and click *Open*.
3. In the **Property Manager**, specify the *Output type*, *Sheet type*, and data columns.
4. If necessary, create a *WellData* module by double-clicking *Well | WellData* in the **Module Manager**.
5. Connect the *Data Source* module to the *WellData* module.

To import wells in a non-tabular format:

1. Click the **File | Import** command.
2. In the **Import** dialog, select the well data file and click *Open*.
3. Specify any file import options in the file type specific **Import Options** dialog.
4. In the **Network Manager**, you can see that a *WellData* module was created. Click on the *WellData* module to select it. In the **Property Manager**, you can see the wells that were imported. If you click on the ⊞ next to any well, you can see the information that was imported for that well.

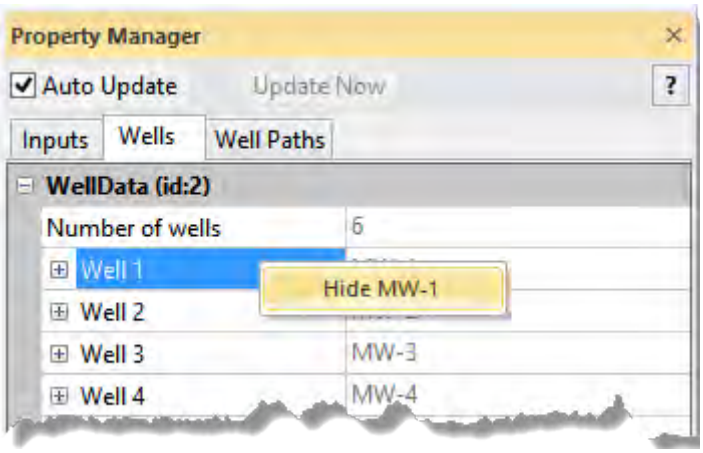
Importing Additional Wells

If an existing *WellData* module exists and your files are in a tabular format, you can use the steps in the *Importing Wells* section above to import additional wells.

If the data file is in a non-tabular format, such as LAS, check the *Add to existing WellData module* check box in the **Import Options** dialog. Alternatively, click on the *WellData* module. In the **Property Manager**, click the *Add Data* button.

Hiding Wells

To hide wells, click on the *WellData* module in the **Network Manager**. In the **Property Manager**, click on the well to be hidden. Right-click and select **Hide [well name]** from the menu. The well is disabled from the *WellData* module and hidden in the viewer window. To show a hidden well, right-click on the well and select **Show [well name]** from the menu.



Select the Hide MW-1 text to hide the MW-1 well in the viewer window.

Alternatively you can hide wells in the *WellRender* module properties. Click on the *WellRender* module. In the **Property Manager**, click on the **General** tab. In the *Wells* section, uncheck the box next to the well you don't want to display.

Changing Wells

To change well information, update the well information in the **Voxler** worksheet. If the well data is imported into a *Data Source* module, click the *Edit Worksheet* button in the **Property Manager** to update the well data in the worksheet. Changes made to the well information are automatically visible when you return to the **Viewer** window.

If the well data was imported directly into the *WellData* module, update the well information in the original data file. Next, in **Voxler**, use the steps in the *Importing Wells* section above to import the changed data. **Voxler** will automatically change the data in the *WellData* node to the new data.

Displaying Wells

To display wells, add a *WellRender* module to the *WellData* module. Click on the *WellData* module to select it. Click the **Network | Graphics Output | WellRender** command. The default display for the wells is shown.

Setting Well Properties

Well properties are set in the *WellRender* module. Click on the *WellRender* module in the **Network Manager** to select it. Properties are changed in the **Property Manager**.

Gridding and Mapping WellData

Once *WellData* is imported, it can be displayed as a *WellRender* or it can be extracted into X, Y, and Z values and gridded for display as a *VolRender*, *Isosurface*, *HeightField*, or any other type of lattice-based map.

This section starts with an already existing *WellData* module. To import the data used for these sections, follow the steps on the *Importing WellData and Displaying Wells* page.

Extracting WellData

Before anything can be done with the log information, the wells need to be extracted to a data file.

1. Click on the *WellData* module in the **Network Manager**.
2. Click the **Network | Computational | ExtractPoints** command to attach an *ExtractPoints* module to the data.
3. Click on the *ExtractPoints* module in the **Network Manager**.
4. In the **Property Manager**, change the *Output components* to 2, so that both the MnO and TiO2 can be extracted.
5. Change the *Component [2]* to the TiO2 log.

Gridding WellData

The extracted data can be gridded using a *Gridder* module.

1. Click the **Network | Computational | Gridder** command to attach a *Gridder* module to the extracted points.
2. In the **Property Manager**, set the *input component* to 2 to grid the TiO2 data.
3. Because the data is more likely to be relevant to other values in the same Z plane, an anisotropic search method or gridding method may need to be set. For now, we'll use the default isotropic settings.
4. Click the *Begin Gridding* button to grid the TiO2 data.

Mapping WellData

Once the data is gridded, any lattice based maps can be added. Click the **Network | Graphics Output | Isosurface** to add an *Isosurface* module. The *Isosurface* properties can be changed to better connect the wells.

Definitions

Well data are collections of XYZ points along a well, optionally with associated log values, well tops, azimuth, inclination, and measured depth values.

A **Well Name** is the name assigned to a well. The **Well Name** is also referred to as **Well ID** or **Borehole Name**.

The **Well Name** is a unique name assigned to each well. This name is used to connect data from multiple imports to the same well.

A **Well Top** is the physical well location in X, Y, and Z coordinates. When you are loading *WellData*, the well top is imported into the *Top X (Collar Easting)*, *Top Y (Collar Northing)*, and *Top Z (Collar Elevation)*. The **Well Top** is also referred to as the **Collar** or **survey location**.

The **Well Top** X, Y, and Z values can be in any units. It is highly recommended that the X, Y, and Z units are the same (all meters or all feet) for best presentation.

The **Inclination** is the deviation from vertical in a directional survey. This value is in degrees and is between zero and 90. Zero is completely vertical and 90 is completely horizontal. This is used in conjunction with **Azimuth** when importing directional survey information into a *WellData* module.

The **Dip** and **Inclination** are similar, but are used in different ways. The **Dip** measures the deviation from horizontal of the well path and the **Inclination** measures the deviation from vertical of log measurements in a directional survey.

The **Azimuth** is the compass orientation of the well. This is recorded in degrees and the value is between 0 and 360. When importing a collars table, this is used in conjunction with **Dip** to describe the default well path. When importing a directional survey, this is used in conjunction with **Inclination** to describe the raw well trajectory.

The **MD (Measured Depth)** is a positive value that measures down from the top of a well. An **MD** value of zero is the top of the well. The final depth of the well is the largest the **MD** value can be.

MD is used in combination with X, Y, and Z values to determine pre-computed well paths.

MD is used in combination with azimuth and inclination when importing directional data.

MD is used in combination with logs to display certain characteristics at specific locations along the well.

The **Total Depth** is the total distance travelled along a well trace from the well collar (Top X, Top Y, Top Z values) to the bottom of a vertical well.

The **Total Depth** value can be used in combination with top X, top Y, top Z, azimuth, and dip to import a well path from a collars table.

The **From Depth** is the starting depth of an interval measurement for a log. This is used in connection with the **To Depth** when importing interval type well logs.

The **To Depth** is the ending depth of an interval measurement for a log. This is used in connection with the **From Depth** when importing interval type well logs.

A **Well Top** is the physical well location in X, Y, and Z coordinates. When you are loading *WellData*, the well top is imported into the *Top X (Collar Easting)*, *Top Y (Collar Northing)*, and *Top Z (Collar Elevation).* The **Well Top** is also referred to as the **Collar** or **survey location**.

The **Well Top** X, Y, and Z values can be in any units. It is highly recommended that the X, Y, and Z units are the same (all meters or all feet) for best presentation.

The **Top X (Easting)** value is the X value of the well collar. Used in combination with the Top Y (Northing) and Top Z (Elevation), the well top position is known.

The **Top X (Easting)** value can be used in combination with top Y, top Z, azimuth, dip, and total depth to import a well path from a collars table.

The **Top Y (Northing)** value is the Y value of the well collar. Used in combination with the Top X (Easting) and Top Z (Elevation), the well top position is known.

The **Top Y (Northing)** value can be used in combination with top X, top Z, azimuth, dip, and total depth to import a well path from a collars table.

The **Top Z (Elevation)** value is the Z value of the well collar. Used in combination with the Top X (Easting) and Top Y (Northing), the well top position is known.

The **Top Z (Elevation)** value can be used in combination with top Y, top Z, azimuth, dip, and total depth to import a well path from a collars table.

The **X (Easting)** value is the X position along the computed well path. A computed well path describes a line in 3D space.

The **Y (Northing)** value is the Y position along the computed well path. A computed well path describes a line in 3D space.

The **Z (Elevation)** value is the Z position along the computed well path. A computed well path describes a line in 3D space.

Lattice Source Module

Lattice source modules are created when a grid or lattice file is loaded into the **Network Manager** using the **File | Import** command. A lattice consists of a one-, two-, or three-dimensional array of data. Lattice files typically contain X, Y, Z, and component information.

Inputs

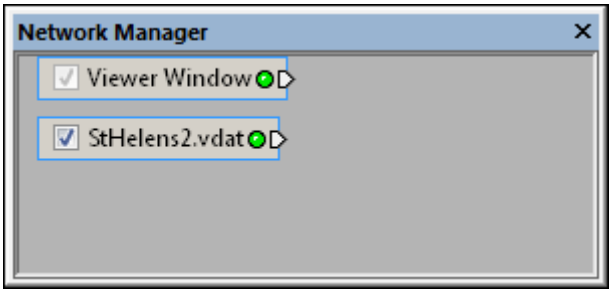
Defined from the lattice defined in the file.

Outputs

Depending on the type of lattice imported, the lattice source module may be connected to the *Graphics Output Modules* or *Computational Modules.* An *Info Module* may also be connected to the output node.

Properties

Lattice source module properties include various statistics about the loaded data set and a section that allows the coordinate limits to be changed. Click on the *lattice source* module in the **Network Manager** to display the properties in the **Property Manager**. The *lattice source* module properties are described below.



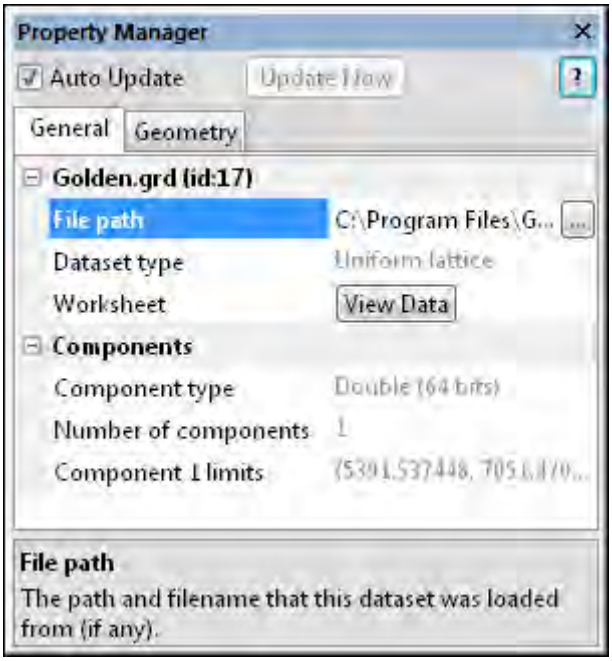
*Select a lattice module in the **Network Manager** to display its properties in the **Property Manager**.*

The *Info* module contains the following tabs in the **Property Manager:**

General

Geometry

General Options



You can change the File path of lattice source modules in the ***Property Manager****.*

File Path

The path of the loaded source file appears next to *File path.* Click the ... button to the right of the file path to display the **Open** dialog. The **Open** dialog appears. The currently loaded file name is display in the *File name* section. The path for the current file is shown. Click the *Open* button to reload the current file. This can be used to update the network after the data file has been modified.

Alternatively browse and select a new file. Click the *Open* button to load the new file. The dialog closes and the selected file is loaded. If the new file does not display geometry information, a warning message will appear and the original file remains.

When a source file that contains multiple layers is selected, the Select Dataset dialog appears. Select the correct layer and click *OK*. That layer is imported to replace the existing data. If the new file format is not compatible with the currently specified file format, an warning message will appear.

Dataset Type

The *Dataset type* displays the type of data set to which this module is connected. This property cannot be changed.

Worksheet

Click the *View Data* button next to the *Worksheet* command to display the data in a worksheet view. The **Lattice View** dialog opens.

Component Type

The *Component type* details the type of data associated with each data component. This property cannot be changed.

Number of Components

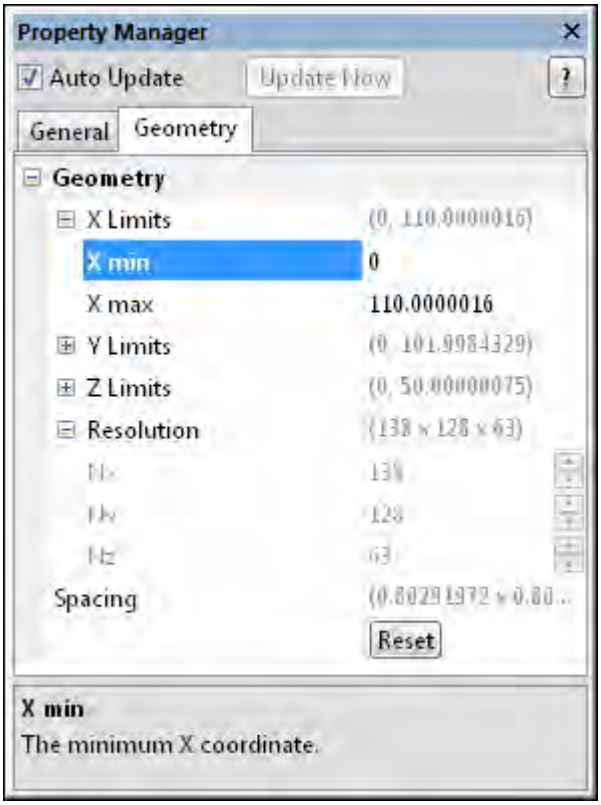
The *Number of components* shows the number of data components per node. This property cannot be changed.

Component Limits

The *Component limits* display the limits for each data component. This property cannot be changed.

Lattice Source Module - Geometry Page

The lattice source module **Geometry** page displays the extents of the lattice. To open the **Geometry** page, click on the *lattice source* module in the **Network Manager**. In the **Property Manager**, click on the **Geometry** tab.



This is an example of the information displayed in the Property Manager on the Geometry tab for a lattice source module.

X Limits

The *X Limits* item displays the limits in the X direction for the function lattice. Enter the *X min* and *X max* coordinates. These ranges are the limits of the lattice that is created. The values are in map coordinates. To change the values, highlight the existing value and type the desired minimum or maximum value.

Y Limits

The *Y Limits* item displays the limits in the Y direction for the function lattice. Enter the *Y min* and *Y max* coordinates. These ranges are the limits of the lattice that is created. The values are in map coordinates. To change the values, highlight the existing value and type the desired minimum or maximum value.

Z Limits

The *Z Limits* item displays the limits in the Z direction for the function lattice. Enter the *Z min* and *Z max* coordinates. These ranges are the limits of the lattice that is created. The values are in map coordinates. To change the values, highlight the existing value and type the desired minimum or maximum value.

Resolution

Resolution is the number of nodes in the lattice. For a lattice source module, these values cannot be changed. The *Resample* module may be used to calculate a new lattice with a different size or resolution.

Spacing

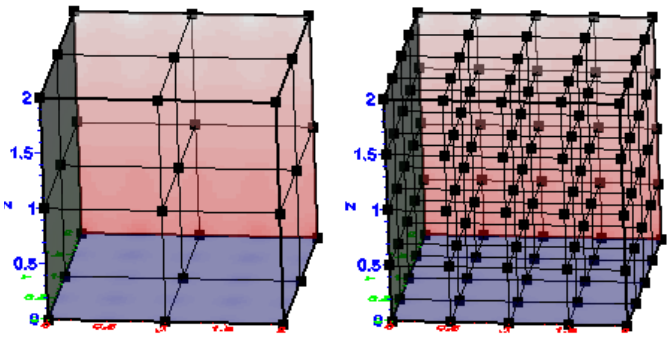
Spacing displays the distance between nodes in the X, Y, and Z directions. This property cannot be changed.

Reset

Click the *Reset* button to restore the coordinates in the *X Limits*, *Y Limits*, or *Z Limits* sections to their original values as loaded from the file.

Lattice Size Example

Consider two lattices: one is 3x3x3, the other is 5x5x5. The smaller lattice has 27 (3^3) nodes and the larger lattice has 125 (5^3) nodes. Increasing the resolution rapidly increases the number of nodes and the corresponding memory requirements.



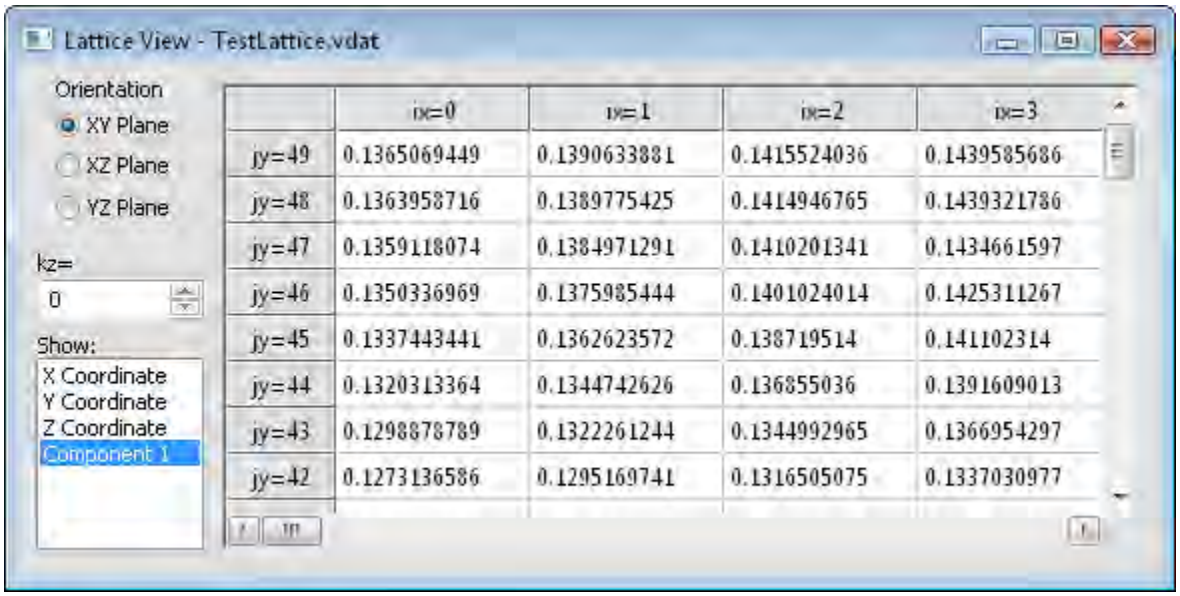
The 3x3x3 lattice on the left has 27 nodes.
The 5x5x5 lattice on the right has 125 nodes.

The default lattice size is 50x50x50 with 125,000 nodes. This size works well with the amount of memory on most computers.

Lattice View Dialog

The **Lattice View** window allows you to display a worksheet view of a lattice data set. Lattice views are static snapshots of lattice at the time the view is opened. The displayed values cannot be edited, nor are they updated if the lattice changes; however, multiple lattice views may be opened simultaneously to allow comparison between points in time. Worksheets for lattices and point data differ in appearance. These views are not saved to disk with the network.

Click the *View Data* button while displaying a lattice source module's properties. The **Lattice View** window displays. Its included fields are described below.



*The **Lattice View** window allows you to view a data set in a static worksheet format.*

Orientation

The *Orientation* controls specify which direction the volume is sliced. Only a single two-dimensional slice through the volume is displayed in the worksheet at a time. Choose the type of orientation for the data view.

XY Plane	X increases across, Y increases up
XZ Plane	X increases across, Z increases up
YZ Plane	Y increases across, Z increases up

Index (kz, jy, or ix)

The type of index displayed depends on the type of orientation selected: *kz* corresponds to the XY plane; *jy* corresponds to the XZ plane; and *ix* corresponds to the YZ plane. Once the orientation is selected, you can easily "spin" through the data by selecting different index values. The indices for the ix, jy, and kz directions within the worksheet are all 0-based. This indicates that the first index is zero, not one.

Show Components

The *Show* box allows you to select which component to include from the lattice for display in the worksheet.

Resize Dialog

If the data exceeds the size of the window, horizontal and/or vertical scroll bars will be displayed. You can resize, minimize, or maximize the **Lattice View** window by dragging its border, clicking the  button, or clicking the  button, respectively.

Close Window

To close the **Lattice View** window, click the  button at the top right of the window.

How can I create an ASCII XYZC lattice file?

Use Plot-3D files .P3D and .Q to import a regularly spaced lattice that is in ASCII format. The .P3D and .Q files can be created in a text editor, like Notepad. Both files need to have the same name so they can read each other.

P3D File

The top line the P3D file shows how many nodes there are in the lattice (X, Y, Z). Then, all of the X values are listed, all of the Y values are listed, and all of the Z values are listed. The order for the points is X increasing while Y and Z stays the same. At the end of the max X, increase to the next Y value continue until all X values are listed for all Y values at the lowest Z. Then, increase to the next Z and repeat the listing of X values at each Y. After all X values are listed, list all Y values. This is the Y value associated with the X value at the same location. Then, all Z values are listed. This is contained in the P3D file.

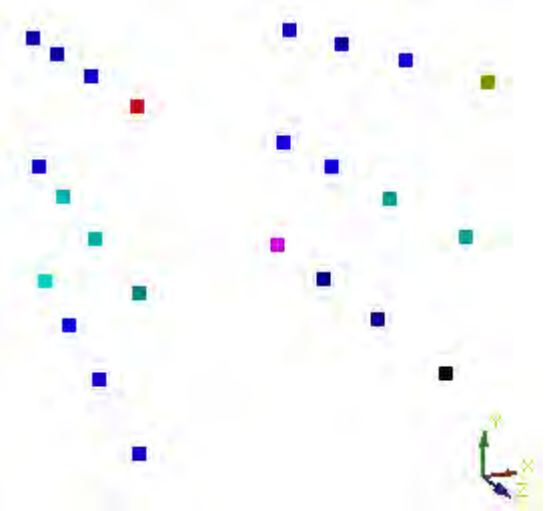
Q File

The Q file contains the same initial row header and then four cells of zeros, the component information for each node (in the same order as the XYZ data), and then another set of zeros. To determine how many zeros to include at the end of the file, multiple the grid size by 4.

Example

This is an example of a 2x3x4 regular lattice (P3D and Q files).

1. Choose the **File | Import** command.
2. Select the .P3D file. Accept default values in **Import** dialog.
3. The *your-file-name.p3d* data module is added to **Network Manager**.
4. Right-click on the module and select **Computational | ScatterPlot**. A scatter plot of the 2x3x4 lattice is displayed.



This is a scatter plot display of a 2x3x4 lattice created from a .P3D and .Q file.

Geometry Source Module

Geometry source modules are created when an Inventor .IV, AutoCAD DXF, Esri SHP, Golden Software BLN file or other type of geometry file is loaded. It may be useful to draw *geometry source* modules in different styles, including *Shaded*, *Lines*, or *Points*.

Inputs

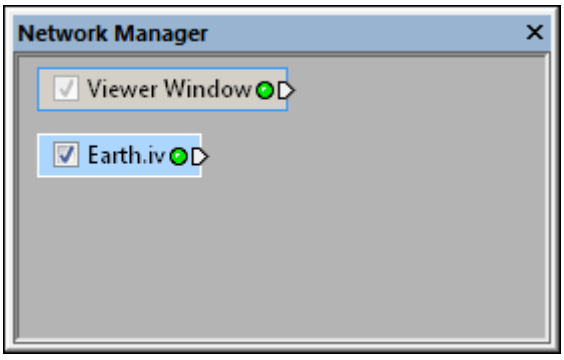
Defined from the geometry defined in the file.

Outputs

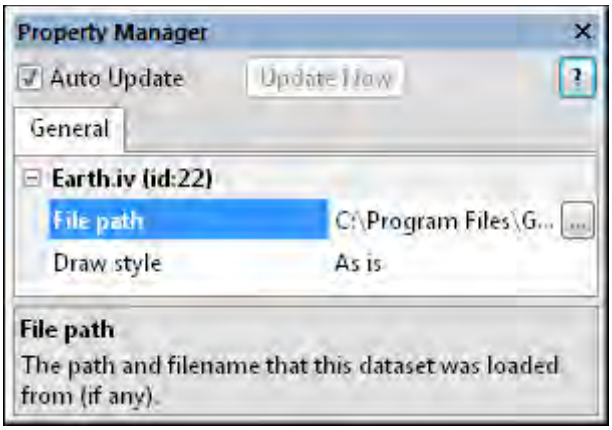
A *ClipPlane Module*, *Info Module*, and *BoundingBox* can be connected to the output node.

Properties

Click on the *geometry source module* in the **Network Manager** to select it. The properties are displayed in the **Property Manager**. The *geometry source* module properties are described below.



Select a geometry data set in the Network Manager to display its properties in the Property Manager.



*Select a geometry file in the **Network Manager** to display its properties in the **Property Manager**.*

File Path

The path of the loaded source file appears next to *File path.* Click the ... button to the right of the file path to display the **Open** dialog. The **Open** dialog appears. The currently loaded file name is display in the *File name* section. The path for the current file is shown. Click the *Open* button to reload the current file. This can be used to update the network after the data file has been modified.

Alternatively browse and select a new file. Click the *Open* button to load the new file. The dialog closes and the selected file is loaded. If the new file does not display geometry information, a warning message will appear and the original file remains.

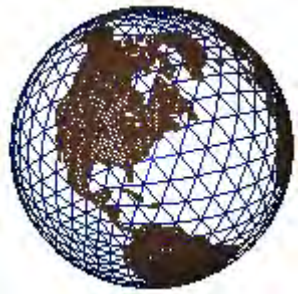
Draw Style

The *Draw style* option can be used to force the geometry to display in a certain way, overriding the drawing styles specified within the loaded file. The *Draw Style* options are *As is*, *Shaded*, *Lines*, or *Points.*

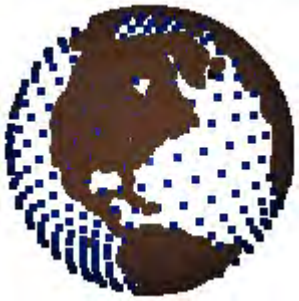
- Choose *As is* to render all geometry as specified in the input file.
- Choose *Shaded* to render all polygonal geometry as shaded surfaces.
- Choose *Lines* to render all geometry as border lines.
- Choose *Points* to render all geometry as vertex points.



The Earth.IV file with a Draw Style of Shaded.



The Earth.IV file with a Draw Style of Lines.



The Earth.IV file with a Draw Style of Points.

Point Source Module

Point source modules are created when a previous version (**Voxler 3** or older) **Voxler** project file (*.voxb) containing point source modules is opened in **Voxler 4** with the **File | Open** command. A point source module will be created when using the **File | Import** command to import a points data file that is NOT in a tabular file format, such as LAS LiDAR or some KML files.

In **Voxler 4**, the *Data Source* module is created for point data in a tabular file format, such as Excel XLS and XLSX, DAT, or CSV files.

Input Modules

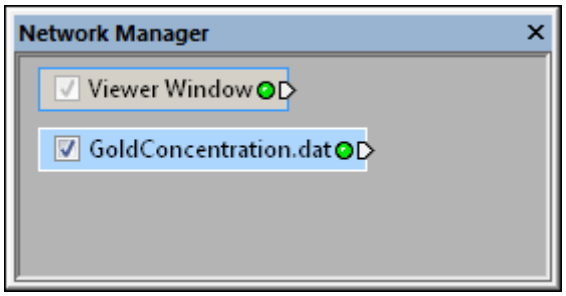
Defined from the point coordinates located in the data file.

Output Modules

Depending on the type of data imported, the point module may be connected to the *Graphics Output Modules* or the *Computational Modules.* An *Info Module* may also be connected to the output node.

Properties

Click on the *point source* module in the **Network Manager** to select it. The **Property Manager** displays various statistics about the loaded data set, the point source module. This is the same information displayed by the *Info* module. The *point source* module properties are described below.



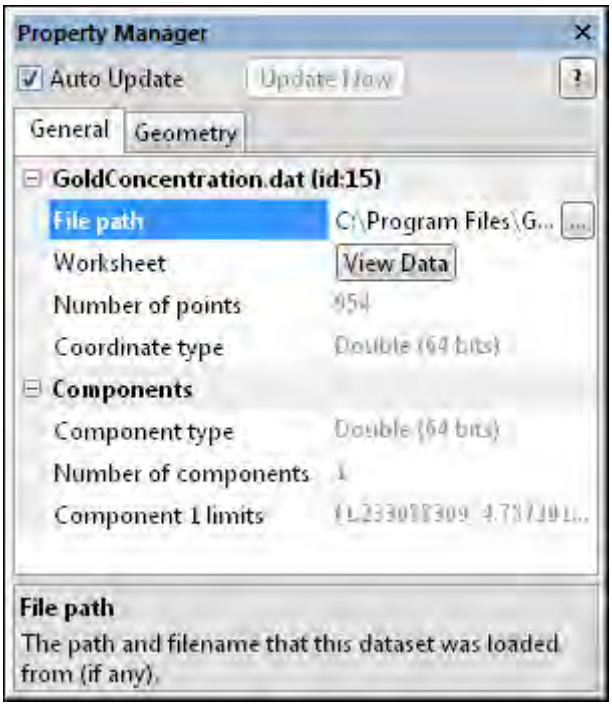
*Select a point data set in the **Network Manager** to display its properties in the **Property Manager**.*

The *point source* module contains the following tabs in the **Property Manager:**

General

Geometry

General Options



The information displayed for point set data is the same as that shown in the Info module.

File Path

The path of the loaded source file appears next to *File path.* Click the  button to the right of the file path to display the **Open** dialog. The **Open** dialog appears. The currently loaded file name is display in the *File name* section. The path for the current file is shown. Click the *Open* button to reload the current file. This can be used to update the network after the data file has been modified.

Alternatively browse and select a new file. Click the *Open* button to load the new file. The dialog closes and the selected file is loaded. If the new file does not display geometry information, a warning message will appear and the original file remains.

Worksheet

Click the *View Data* button to display the data in a worksheet view. The **Data View** dialog opens.

Number of Points

The *Number of points* displays the number of points in the data file. This property cannot be changed.

Coordinate Type

The *Coordinate type* details the type of data storage associated with the X, Y, and Z directions. This property cannot be changed.

Component Type

The *Component type* details the type of data storage associated with each data component. This property cannot be changed.

Number of Components

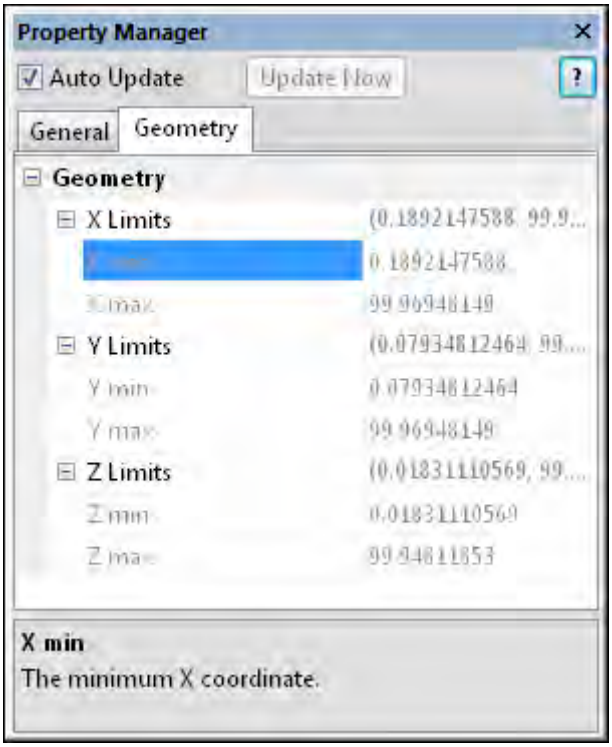
The *Number of components* item details the number of data components per data point. This property cannot be changed.

Component Limits

The *Component limits* display the minimum and maximum values for the component. Each component is listed separately, if the input contains more than one component. This property cannot be changed.

Point Source Module - Geometry Page

The *point source* module **Geometry** page displays the extents of the data. To open the **Geometry** page, click on the *point source* module in the **Network Manager**. In the **Property Manager**, click on the **Geometry** tab.



This is an example of the information displayed in the Property Manager on the Geometry tab for a point source module.

X Limits

The *X limits* item displays the limits in the X direction in the point set. The *X min* displays the smallest X value and the *X max* displays the largest X value in the data set.

Y Limits

The *Y limits* item displays the limits in the Y direction in the point set. The *Y min* displays the smallest Y value and the *Y max* displays the largest Y value in the data set.

Z Limits

The *Z limits* item displays the limits in the Z direction in the point set. The *Z min* displays the smallest Z value and the *Z max* displays the largest Z value in the data set.

Data Source Commands

Choose the **Network | Data Source** commands to generate test lattices. These modules serve as the source of raw data. The data created from the **Data Source | FunctionLattice** or **TestLattice** commands are created from mathematical functions.

Data source modules include the following:

- *FunctionLattice* creates a uniform three-dimensional lattice from a user-defined mathematical function
- *TestLattice* generates various predefined lattices for testing

Import Data, Well Data, Geometry, or Lattices

The **File | Import** command creates a new source module by loading a file. The module is named with the imported file name. See the *File Format Chart* for more details on supported file formats.

FunctionLattice

The **Network | Data Source | FunctionLattice** command adds a *FunctionLattice* module to the **Network Manager**.

The *FunctionLattice* module creates a new lattice from a user-defined function. The range and resolution of the output lattice and the number of components are specified in the **Property Manager**.

Inputs

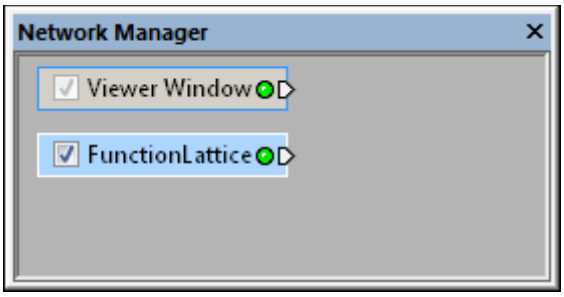
The *FunctionLattice* has no input. To change the data displayed by the *FunctionLattice*, select the *FunctionLattice* module in the **Network Manager** and change the *Function Expressions* property in the **Property Manager**.

Outputs

The *FunctionLattice* creates a uniform lattice. It may be connected to the *Graphics Output Modules* or the *Computational Modules*. An *Info Module* may also be connected to the output node.

Properties

Click on the *FunctionLattice* in the **Network Manager** to select it. The *FunctionLattice* properties are displayed in the **Property Manager**. The available properties are described below.



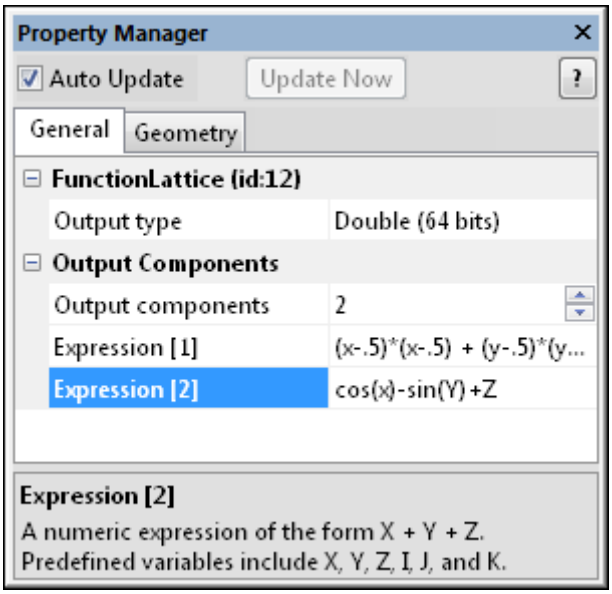
*Select the FunctionLattice module in the **Network Manager** to display its properties in the **Property Manager**.*

The *FunctionLattice* module contains the following tabs in the **Property Manager:**

General

Geometry

General Options



In addition to changing the output type, you can enter a custom expression to define the FunctionLattice.

Output Type

The *Output type* is the numeric format used for the lattice X, Y, Z, and component values. Select a *Signed*, *Unsigned*, *Float*, *or Double* output primitive type to be stored in the output lattice. The larger the output type, the larger the resulting lattice is. If you are using large coordinate or component values, you may need to use a larger output type, such as *Float (32 bits)* or *Double (64 bits)*, but this will increase the output lattice size and the amount of memory needed to create the lattice. To change the type, click on the existing option and select the desired option from the list.

Function Expressions

The *Output components* shows the number of components that will be created by the *FunctionLattice*. To change the number of components, highlight the existing value and type the desired value or click the  to increase or decrease the number of components. This value can be

between 1 and 20 components for each *FunctionLattice*. If you require additional components, you will need to create a new *FunctionLattice*.

Expression represents a mathematical expression of the form X + Y + Z. Predefined variables include the global coordinate locations of the current node expressed as x, y, z, and the indices of the current node expressed as i, j, and k. If the function is omitted, all values for that component are blanked. A new *Expression* will be listed for each of the components. Many different predefined mathematical functions can be used to create the equation. To change the equation, highlight the existing text and type the desired equation.

TestLattice

The **Network | Data Source | TestLattice** command adds a *TestLattice* module to the **Network Manager**.

The *TestLattice* module generates a variety of predefined lattices that can be used for testing or experimenting with various modules. The output dimensions, limits, and primitive type may be specified for the output lattice.

Inputs

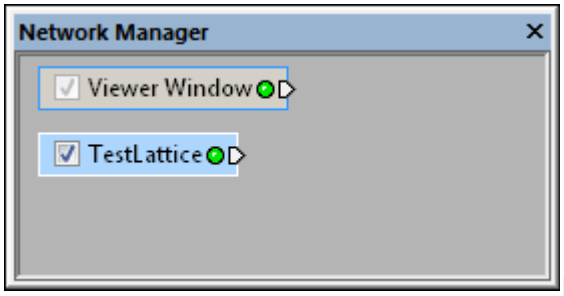
The *TestLattice* has no input. To change the data displayed by the *TestLattice*, select the *TestLattice* module in the **Network Manager** and change the *Function* property in the **Property Manager**.

Outputs

The *TestLattice* creates a uniform lattice. It may be connected to the *Graphics Output Modules* or the *Computational Modules*. An *Info Module* may also be connected to the output node.

Properties

Click on the *TestLattice* module in the **Network Manager** to select it. The *TestLattice* properties are displayed in the **Property Manager**. The available properties are described below.



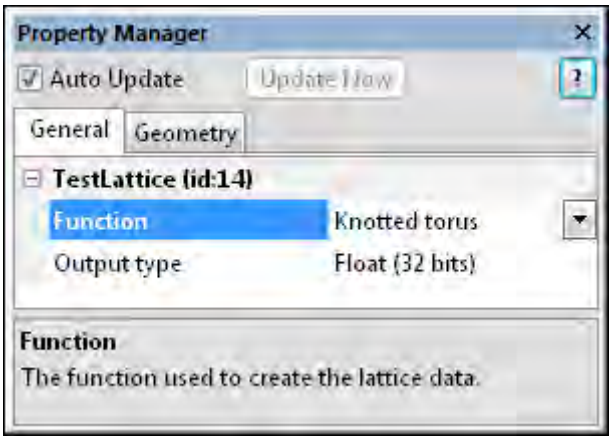
*Select the TestLattice module in the **Network Manager** to display its properties in the **Property Manager**.*

The *FunctionLattice* module contains the following tabs in the **Property Manager:**

General

Geometry

General Options

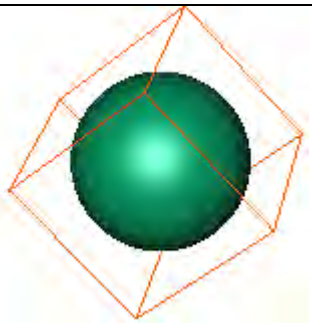


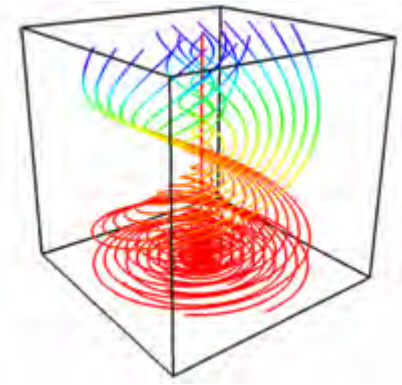
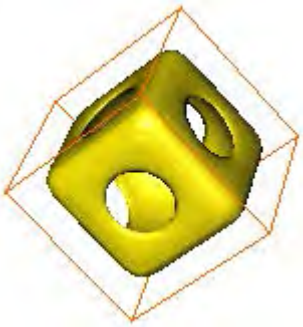
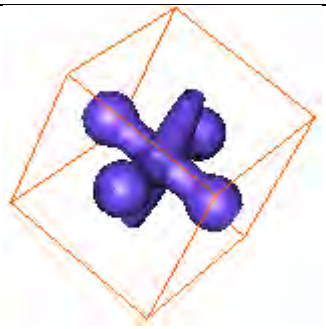
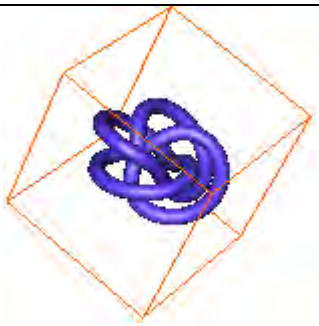
Test lattices are created using one of six predefined Function options.

Function

Choose the desired predefined *Function* to use to create the lattice data.

Examples of each of the function types appear below. References are located at the bottom of this page.

Knotted Torus	The *Knotted Torus* function is based on equations presented by Poston, et al.	
Spherical Density	The *Spherical Density* function models the distance squared from the point 0.5, 0.5, 0.5 over the unit cube ranging from 0.0 to 1.0.	

Vector Vortex	The *Vector Vortex* function computes a spiral vector field. The x, y, and z components of the vector V at (x, y, z) are given by: • Vx = -y / s; Range = -1.0 to +1.0 • Vy = x / s; Range = -1.0 to +1.0 • Vz = α * z; Range = 0.0 to α where s = sqrt(x*x + y*y), the radial distance in the xy plane.	
Wiffle Cube	The *Wiffle Cube* function is a cube with a sphere removed from the center, as described by Bourke.	
Jack	The *Jack* function simulates a jack (from the game of jacks) by creating a density function from four spheres and three ellipsoids.	
Knotted Rope	The *Knotted Rope* function computes a distance function from the center of a knotted rope. Imagine a rope tied into a loose knot in which every point in the lattice is the minimum distance to the center of the rope.	

Output Type

The *Output Type* is the numeric format used for the lattice X, Y, Z and component values stored in the output data. Select a *Signed*, *Unsigned*, *Float*, and *Double* output primitive type to be stored in the output lattice. The larger the output type, the larger the resulting lattice is. If you are using large coordinate or component values, you may need to use a larger output type, such as *Float (32 bits)* or *Double (64 bits)*, but this will increase the output lattice size and the amount of memory

needed to create the lattice. To change the type, click on the existing option and select the desired option from the list.

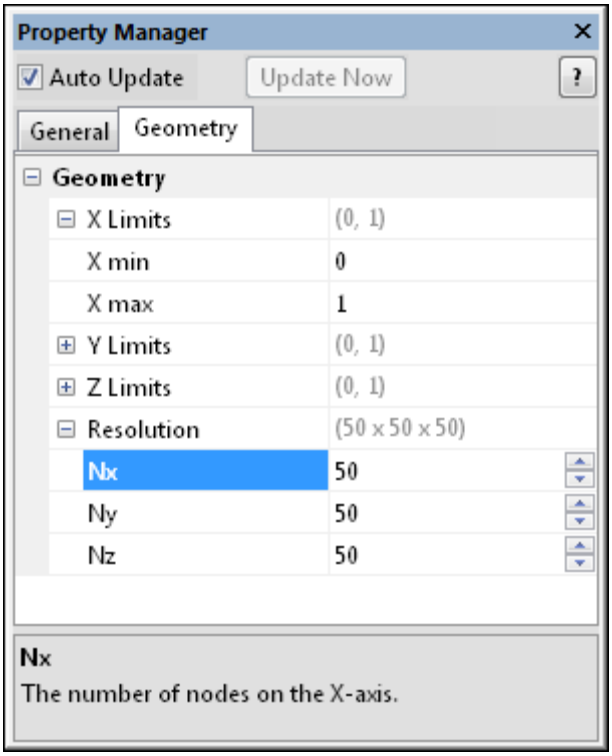
References

Bourke, P. (1990) *Wiffle cube*. Accessed at http:// astronomy.swin.edu.au/~ pbourke/surfaces/ wiffle.

Poston, T., Nguyen, H. T., Heng, P-A, and Wong, T-T. (1997) *"Skeleton climbing": fast isosurfaces with fewer triangles*. Proceedings of the 5th Pacific Conference on Computer Graphics and Applications. References can be accessed at http://www.cse.cuhk.edu.hk/~ttwong/papers/sc/sc.html

Geometry Page

The *FunctionLattice* and *TestLattice* modules **Geometry** page displays the extents and resolution of the lattice. To open the **Geometry** page, click on the *FunctionLattice* or *TestLattice* module in the **Network Window**. In the **Property Manager**, click on the **Geometry** tab.



This is an example of the information displayed in the Property Manager on the Geometry tab for a FunctionLattice module.

X Limits

The *X Limits* item displays the limits in the X direction for the function lattice. Enter the *X min* and *X max* coordinates. These ranges are the limits of the lattice that is created. The values are in map coordinates. To change the values, highlight the existing value and type the desired minimum or maximum value.

Y Limits

The *Y Limits* item displays the limits in the Y direction for the function lattice. Enter the *Y min* and *Y max* coordinates. These ranges are the limits of the lattice that is created. The values are in map coordinates. To change the values, highlight the existing value and type the desired minimum or maximum value.

Z Limits

The *Z Limits* item displays the limits in the Z direction for the function lattice. Enter the *Z min* and *Z max* coordinates. These ranges are the limits of the lattice that is created. The values are in map coordinates. To change the values, highlight the existing value and type the desired minimum or maximum value.

Resolution

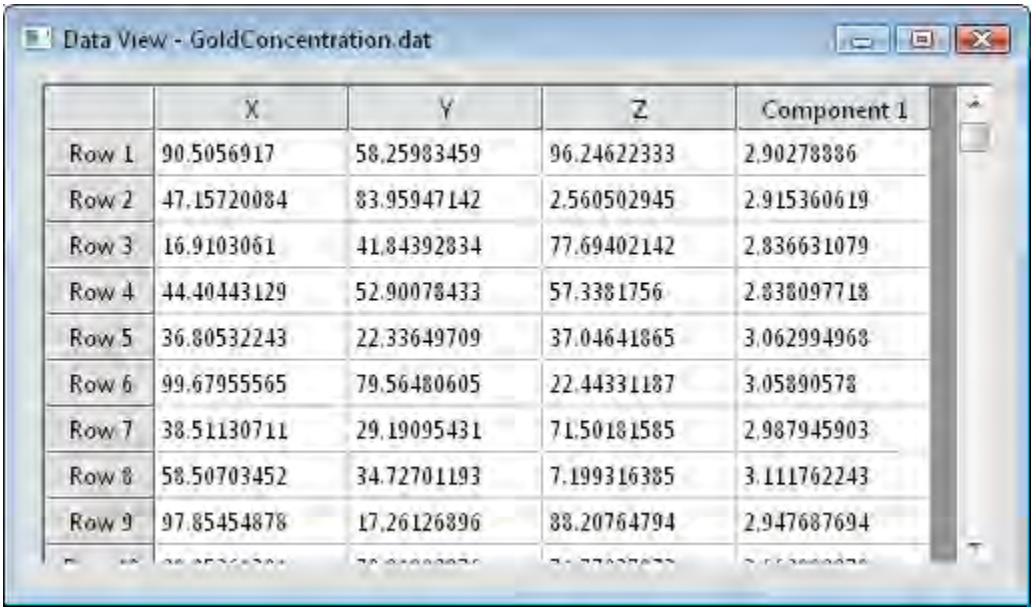
Resolution is the number of nodes created in the lattice. By default, this value is 50 in each direction. Enter the number of nodes (*Nx*, *Ny*, and *Nz*) on each axis in the lattice. An increase in the number of lattice nodes provides more resolution, but also increases the amount of time needed to create the lattice and to draw output graphics. It also increases the amount of memory needed to display graphics on-screen and to save the graphic to a .VOXB file. To change the resolution, highlight the existing value and type a new value or click the  button to increase or decrease the values. Values can be between 1 and 10000.

The *Resample* module may be used to calculate a new lattice with a different size or resolution.

Data View Dialog

The **Data View** dialog allows you to display a worksheet view of a data set. Data views are static snapshots of data at the time the view is opened and is intended for display purposes only. The displayed values cannot be edited, nor are they updated if the data changes; however, multiple data views may be opened simultaneously to allow comparison between points in time. Worksheets for lattices and point sets differ in appearance. These views are not saved to disk with the network.

In the **Property Manager**, click the *View Data* button while displaying a point source module's properties to display the **Data View** dialog. Each data point occupies a single row of the worksheet.



	X	Y	Z	Component 1
Row 1	90.5056917	58.25983459	96.24622333	2.90278886
Row 2	47.15720084	83.95947142	2.560502945	2.915360619
Row 3	16.9103061	41.84392834	77.69402142	2.836631079
Row 4	44.40443129	52.90078433	57.3381756	2.838097718
Row 5	36.80532243	22.33649709	37.04641865	3.062994968
Row 6	99.67955565	79.56480605	22.44331187	3.05890578
Row 7	38.51130711	29.19095431	71.50181585	2.987945903
Row 8	58.50703452	34.72701193	7.199316385	3.111762243
Row 9	97.85454878	17.26126896	88.20764794	2.947687694

*The **Data View** window displays each point in a data set in a worksheet view for display only.*

Resize Window

If the data exceeds the size of the window, horizontal and/or vertical scroll bars will be displayed. You can resize, minimize, or maximize the **Data View** window by dragging its border, clicking the button, or clicking the button, respectively.

Close Window

To close the **Data View** window, click the button at the top right of the window.

How can I display 3D lines?

Create an Inventor .IV file in a text editor to display 3D lines in **Voxler**.

Example

The following example shows three lines with coordinates in the point section. The number of vertices for each line is specified in the `numVertices` section. Extra white space and carriage returns are shown for readability. The `#Inventor` text must be the first characters in the file, without any leading spaces that can occur when you copy and paste the text to a text editor.

This Inventor IV example displays three lines:

```
#Inventor V2.1 ascii
 Separator {
    Coordinate3 {
       point
       0 0 0,
       1 1 1,
       2 1 1,

       2 2 1,
       2 2 2,
```

```
        2 2 3,
        2 3 2,

        2 3 3,
        3 3 3
     }
     LineSet {
        numVertices 3, 4, 2
     }
 }
```

For complete technical details about the .IV format, refer to the Inventor Mentor website or publication.

How can I display geometric shapes in Voxler?

The Inventor .IV format is commonly used to display geometric shapes in **Voxler**. Refer to the Inventor Mentor for the programming manual for the Inventor language. You can also use the Standford Polygon .PLY format to display geometric shapes.

```
Diamond
Diamond .IV Format Example
#Inventor V2.0 ascii

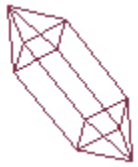
Separator {
  Separator {
    LightModel {
      model PHONG
    }
    Material {
      ambientColor  0.5 0.2 0.2
      diffuseColor  0.5 0.2 0.3
      emissiveColor 0.5 0.0 0.0
      specularColor 0.5 0.0 0.0
      shininess     0.5
      transparency  0.0
    }
    Coordinate3 {
      point
        0.0000 0.0000 2.0000,
        0.5000 -0.5000 1.0000,
        0.5000 0.5000 1.0000,
        -0.5000 0.5000 1.0000,
        -0.5000 -0.5000 1.0000,
        0.5000 -0.5000 -1.0000,
        0.5000 0.5000 -1.0000,
        -0.5000 0.5000 -1.0000,
        -0.5000 -0.5000 -1.0000,
```

```
        0.0000 0.0000 -2.0000,

    }
    IndexedLineSet {
      coordIndex
               0,       1,      -1,
               0,       2,      -1,
               0,       3,      -1,
               0,       4,      -1,
               1,       2,      -1,
               1,       4,      -1,
               1,       5,      -1,
               2,       3,      -1,
               2,       6,      -1,
               3,       4,      -1,
               3,       7,      -1,
               4,       8,      -1,
               5,       6,      -1,
               5,       8,      -1,
               5,       9,      -1,
               6,       7,      -1,
               6,       9,      -1,
               7,       8,      -1,
               7,       9,      -1,
               8,       9,      -1,

    }
    IndexedFaceSet {
      coordIndex
        2, 1, 0, -1,
        4, 1, 0, -1,
        3, 2, 0, -1,
        4, 3, 0, -1,
        5, 6, 2, 1, -1,
        5, 8, 4, 1, -1,
        6, 7, 3, 2, -1,
        7, 8, 4, 3, -1,
        9, 6, 5, -1,
        9, 8, 5, -1,
        9, 7, 6, -1,
        9, 8, 7, -1,

    }
  }
}
```



```
Dodecahedron
Dodecahedron .IV Format Example
#Inventor V2.0 ascii

Separator {
  Info {
    string "dodec.iv generated by IVREAD."
    string "Original data in file dodec.wrl."
  }
  Separator {
    LightModel {
      model PHONG
    }
    MatrixTransform { matrix
           1.00000      0.000000E+00  0.000000E+00  0.000000E+00
          0.000000E+00   1.00000      0.000000E+00  0.000000E+00
          0.000000E+00  0.000000E+00   1.00000      0.000000E+00
          0.000000E+00  0.000000E+00  0.000000E+00   1.00000
    }
    Material {
      ambientColor  0.2 0.2 0.2
      diffuseColor
          0.8000  0.8000  0.8000,
          0.0000  0.0000  1.0000,
          0.0000  1.0000  0.0000,
          0.0000  1.0000  1.0000,
          1.0000  0.0000  0.0000,
          1.0000  0.0000  1.0000,
          1.0000  1.0000  0.0000,

      emissiveColor 0.0 0.0 0.0
      specularColor 0.0 0.0 0.0
      shininess     0.2
      transparency
          0.000,  0.000,  0.000,  0.000,  0.000,  0.000,  0.000,

    }
    TextureCoordinateBinding {
      value PER_VERTEX_INDEXED
    }
    MaterialBinding {
      value PER_VERTEX_INDEXED
    }
    NormalBinding {
      value PER_VERTEX_INDEXED
    }
    ShapeHints {
      vertexOrdering COUNTERCLOCKWISE
      shapeType UNKNOWN_SHAPE_TYPE
      faceType CONVEX
      creaseAngle 6.28319
    }
    Coordinate3 {
      point
        1.0000 1.0000 1.0000,
```

```
            1.0000 1.0000 -1.0000,
            1.0000 -1.0000 1.0000,
            1.0000 -1.0000 -1.0000,
            -1.0000 1.0000 1.0000,
            -1.0000 1.0000 -1.0000,
            -1.0000 -1.0000 1.0000,
            -1.0000 -1.0000 -1.0000,
            0.6180 1.6180 0.0000,
            -0.6180 1.6180 0.0000,
            0.6180 -1.6180 0.0000,
            -0.6180 -1.6180 0.0000,
            1.6180 0.0000 0.6180,
            1.6180 0.0000 -0.6180,
            -1.6180 0.0000 0.6180,
            -1.6180 0.0000 -0.6180,
            0.0000 0.6180 1.6180,
            0.0000 -0.6180 1.6180,
            0.0000 0.6180 -1.6180,
            0.0000 -0.6180 -1.6180,

          }
          Normal {
            vector
              0.5774 0.5774 0.5774,
              0.5774 0.5774 -0.5774,
              0.5774 -0.5774 0.5774,
              0.5774 -0.5774 -0.5774,
              -0.5774 0.5774 0.5774,
              -0.5774 0.5774 -0.5774,
              -0.5774 -0.5774 0.5774,
              -0.5774 -0.5774 -0.5774,
              0.3568 0.9342 0.0000,
              -0.3568 0.9342 0.0000,
              0.3568 -0.9342 0.0000,
              -0.3568 -0.9342 0.0000,
              0.9342 0.0000 0.3568,
              0.9342 0.0000 -0.3568,
              -0.9342 0.0000 0.3568,
              -0.9342 0.0000 -0.3568,
              0.0000 0.3568 0.9342,
              0.0000 -0.3568 0.9342,
              0.0000 0.3568 -0.9342,
              0.0000 -0.3568 -0.9342,

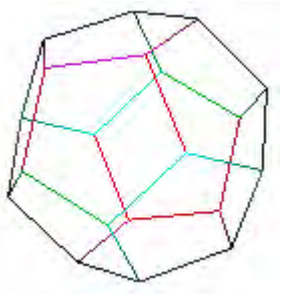
          }
          IndexedFaceSet {
            coordIndex
              1, 8, 0, 12, 13, -1,
              4, 9, 5, 15, 14, -1,
              2, 10, 3, 13, 12, -1,
              7, 11, 6, 14, 15, -1,
              2, 12, 0, 16, 17, -1,
              1, 13, 3, 19, 18, -1,
              4, 14, 6, 17, 16, -1,
              7, 15, 5, 18, 19, -1,
              4, 16, 0, 8, 9, -1,
              2, 17, 6, 11, 10, -1,
              1, 18, 5, 9, 8, -1,
              7, 19, 3, 10, 11, -1,

            textureCoordIndex
              0, 1, 2, 3, 4, -1,
              5, 6, 7, 8, 9, -1,
```

```
        10, 11, 12, 13, 14, -1,
        15, 16, 17, 18, 19, -1,
        20, 21, 22, 23, 24, -1,
        25, 26, 27, 28, 29, -1,
        30, 31, 32, 33, 34, -1,
        35, 36, 37, 38, 39, -1,
        40, 41, 42, 43, 44, -1,
        45, 46, 47, 48, 49, -1,
        50, 51, 52, 53, 54, -1,
        55, 56, 57, 58, 59, -1,

      materialIndex
        2, 2, 2, 2, 2, -1,
        3, 3, 3, 3, 3, -1,
        3, 3, 3, 3, 3, -1,
        2, 2, 2, 2, 2, -1,
        4, 4, 4, 4, 4, -1,
        5, 5, 5, 5, 5, -1,
        5, 5, 5, 5, 5, -1,
        4, 4, 4, 4, 4, -1,
        6, 6, 6, 6, 6, -1,
        0, 0, 0, 0, 0, -1,
        0, 0, 0, 0, 0, -1,
        6, 6, 6, 6, 6, -1,

    }
  }
}
```



```
Prism
Prism .PLY Format Example
ply
format ascii 1.0
comment - construction by Neil Schofield
comment - cube file
element vertex 6
property int x
property int y
property int z
element face 5
property list uchar int vertex_indices
end_header
0 0 0
0 0 1
0 1 1
1 0 0
1 0 1
1 1 1
3 0 1 2
3 3 4 5
4 0 3 4 1
4 1 4 5 2
```

```
4 0 3 5 2
```



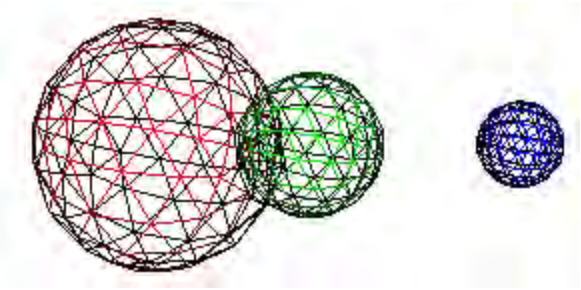
```
Cylinder
Cylinder .IV Format Example
#Inventor V2.0 ascii

Separator {
Material {
diffuseColor 1 0 0
transparency .5
}
Transform {
scaleFactor 1 1 1
translation 0 2 0
rotation 0 0 1 45
}
Cylinder { parts ALL radius 1 height 3 }
}

Sphere
Sphere .IV Format Example
#Inventor V2.0 ascii

Separator {
  Info {
    string "spheres.iv generated by hand."
  }
  Separator {
    LightModel {
      model PHONG
    }
    Separator {
      Material {
        diffuseColor 1.0 0.0 0.0
      }
      Separator {
        Transform {
          translation  0.0 0.0 0.0
        }
        Sphere { radius 1.0 }
      }
    }
    Separator {
      Material {
        diffuseColor 0.0 1.0 0.0
      }
      Separator {
        Transform {
          translation  0.0 0.0 2.0
        }
        Sphere { radius 0.5 }
      }
    }
    Separator {
      Material {
```

```
        diffuseColor 0.0 0.0 1.0
      }
      Separator {
        Transform {
          translation  0.0 0.0 4.0
        }
        Sphere { radius 0.25 }
      }
    }
  }
}
```



```
Tetrahedron
Tetrahedron .IV Format Example
#Inventor V2.0 ascii

Separator {
  Info {
    string "tetrahedron.iv generated by IVREAD."
    string "Original data in file tetrahedron.wrl."
  }
  Separator {
    LightModel {
      model BASE_COLOR
    }
    MatrixTransform { matrix
           1.00000      0.000000E+00  0.000000E+00  0.000000E+00
         0.000000E+00   1.00000      0.000000E+00  0.000000E+00
         0.000000E+00  0.000000E+00   1.00000      0.000000E+00
         0.000000E+00  0.000000E+00  0.000000E+00   1.00000
    }
    Material {
      ambientColor  0.2 0.2 0.2
      diffuseColor
          0.8000  0.8000  0.8000,
          0.0000  1.0000  0.0000,
          1.0000  1.0000  1.0000,
          0.0000  0.0000  1.0000,
          1.0000  0.0000  0.0000,

      emissiveColor 0.0 0.0 0.0
      specularColor 0.0 0.0 0.0
      shininess     0.2
      transparency
          0.000,  0.000,  0.000,  0.000,  0.000,

    }
    TextureCoordinateBinding {
      value PER_VERTEX_INDEXED
    }
    MaterialBinding {
      value PER_VERTEX_INDEXED
```

```
        }
        NormalBinding {
          value PER_VERTEX_INDEXED
        }
        ShapeHints {
          vertexOrdering COUNTERCLOCKWISE
          shapeType UNKNOWN_SHAPE_TYPE
          faceType CONVEX
          creaseAngle 6.28319
        }
        Coordinate3 {
          point
            1.0000 1.0000 1.0000,
            1.0000 -1.0000 -1.0000,
            -1.0000 1.0000 -1.0000,
            -1.0000 -1.0000 1.0000,

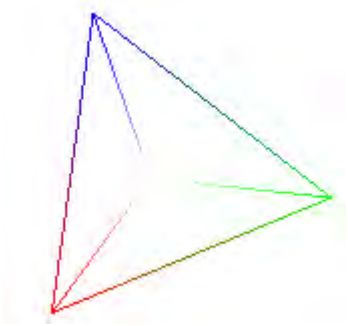
        }
        Normal {
          vector
            0.5774 0.5774 0.5774,
            0.5774 -0.5774 -0.5774,
            -0.5774 0.5774 -0.5774,
            -0.5774 -0.5774 0.5774,

        }
        IndexedFaceSet {
          coordIndex
            3, 2, 1, -1,
            2, 3, 0, -1,
            1, 0, 3, -1,
            0, 1, 2, -1,

          textureCoordIndex
            0, 1, 2, -1,
            3, 4, 5, -1,
            6, 7, 8, -1,
            9, 10, 11, -1,

          materialIndex
            4, 3, 2, -1,
            3, 4, 1, -1,
            2, 1, 4, -1,
            1, 2, 3, -1,

        }
      }
    }
```



Other Shapes

You can use the Inventor language to create a variety of shapes.



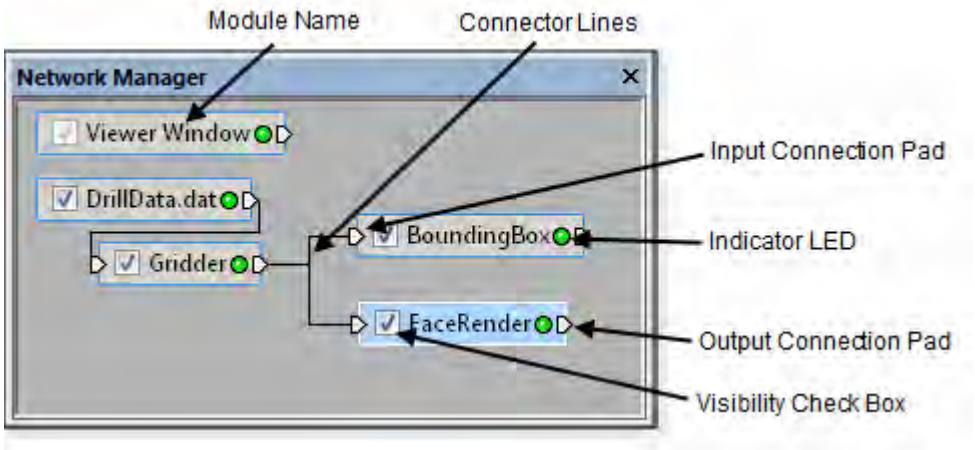
*This is an example of the sample file earth.iv loaded into **Voxler**.*

Chapter 5 - Computational Modules

Introduction to Modules

A module is a data set or a process to be applied to a data set or process. Modules are the building blocks from which the final **Voxler** output is constructed. Modules accept data on their input connection pads ▷, modify the data, and pass it along through the output connection pad ▷.

Modules are displayed in the **Network Manager**. You can connect and disconnect modules to create a visualization network representing the flow of data. Modules need to be connected in order to generate an output. The current geometry output of the modules in the **Network Manager** are displayed in a three-dimensional view in the **Viewer** window. If the data is not connected to a graphics output module, nothing is displayed in the **Viewer** window.



*The **Network Manager** displays the visualization network, which includes all loaded modules and their connections.*

View All Modules

All modules are accessed using the Network menu commands. Alternatively, all modules are accessed in the **Module Manager** when the *Show all modules* button is selected .

View Applicable Modules

When a module is selected in the **Network Manager** and the *Show all modules* button  is not selected in the **Module Manager**, available modules that can be connected to the selected module output port are displayed in the **Module Manager**. Alternatively, right-click a module in the **Network Manager** to display only the applicable modules in the context menu.

On the module description pages, there are **Inputs** and **Outputs** sections to discuss the type of input and output modules that each module are compatible.

Import

Click **File | Import** to open the **Import** dialog. The type of data determines what type of module can be attached to it. **Voxler** supports several different data types. See the *File Format Chart* for a detailed list of supported file formats.

Viewer Window

The **Viewer Window** is a unique module that is automatically created when a new instance of **Voxler** is generated. The *Viewer Window* module appears in the **Network Manager**. The *Viewer Window* module cannot be deleted. The purpose of the *Viewer Window* module is to control the properties of the **Viewer** window for the current instance of **Voxler**. To change the **Viewer** window properties for future instances of **Voxler**, choose the **Tools | Options** command and adjust properties on the **Colors** page of the **Options** dialog.

Module Types

There are four types of modules: *computational*, *data source*, *general*, and *graphics output.* Each module type is discussed below.

Computational Modules

Computational modules alter the data by changing the data type, filtering, creating a gradient, gridding, performing mathematical transformations, merging, resampling, slicing, creating a subset, or transforming coordinates. Click on one of the following computational modules for detailed information on using the module and module properties.

ChangeType

DuplicateFilter

ExclusionFilter

ExtractPoints

Filter

Gradient

Gridder

Math

Merge

Resample

Slice

Subset

Transform

Data Source Modules

Data source modules serve as a source of raw data. The data can be imported or created from mathematical functions. Click on one of the following data source modules for detailed information on using the module and module properties.

Import (*Data Source*, *Point Source*, *Lattice Source*, *Geometry Source*)

FunctionLattice

TestLattice

WellData (combines multiple *Data Source* modules into a single output)

General Modules

General modules display module information and provide custom lighting in the **Viewer** window. Click on one of the following general modules for detailed information on using the module and module properties.

Info

Light

Graphics Output

Graphics output modules create graphics in the **Viewer** window. Typically, these modules require data input. Click on one of the following graphics output modules for detailed information on using the module and module properties.

Annotation

Axes

BoundingBox

ClipPlane

Contours

FaceRender

HeightField

Isosurface

ObliqueImage

OrthoImage

ScatterPlot

StreamLines

Text

VectorPlot

VolRender

WellRender

Computational Modules

Choose the **Network | Computational** command to add a module that changes how data is handled.

The modules are added to the menu in alphabetical order to make them easier to find. If a module is selected in the **Network Manager** when this command is chosen, only compatible output modules for the selected module are displayed. Choose a module from the list and **Voxler** automatically connects it to the selected module.

Computational modules alter the data by changing their type, filtering, creating a gradient, gridding, performing mathematical transformations, merging, resampling, slicing, creating a subset, or transforming coordinates.

Computational modules include the following:

- *ChangeType* changes the type of the input data
- *DuplicateFilter* removes duplicate points from the point set
- *ExclusionFilter* excludes points from the point set according to a user-specified criterion

- *ExtractPoints* creates point data from lattices or well data
- *Filter* filters a uniform lattice
- *Gradient* computes the gradient for a three-dimensional lattice
- *Gridder* converts a point set to a uniform lattice
- *Math* performs a user-specified mathematical function on one or more lattices
- *Merge* combines two or more compatible lattices
- *Resample* alters the resolution of a lattice to a user-specified resolution
- *Slice* extracts a two-dimensional lattice from a three-dimensional lattice
- *Subset* extracts a smaller portion from a lattice
- *Transform* alters lattice or point coordinates

ChangeType

The **Network | Computational | ChangeType** command adds a *ChangeType* module to the network.

The *ChangeType* module changes the data type from one primitive type, e.g., int, float, to another type. Smaller types save memory at the expense of reduced numeric precision. All components of the input data set are converted. This module changes the type of the data components only (not the coordinates). Use the *Transform* module to change the coordinates

The *ChangeType* module is essentially a subset of the *Math* module— each allows the values to be scaled and controls the output size. *ChangeType* has been included primarily for convenience and to make the scaling equations easier to specify.

Inputs

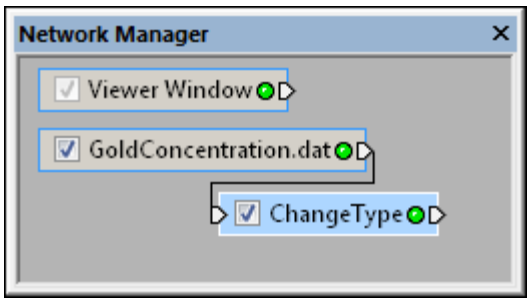
Lattice and point data are input types for the *ChangeType* module.

Outputs

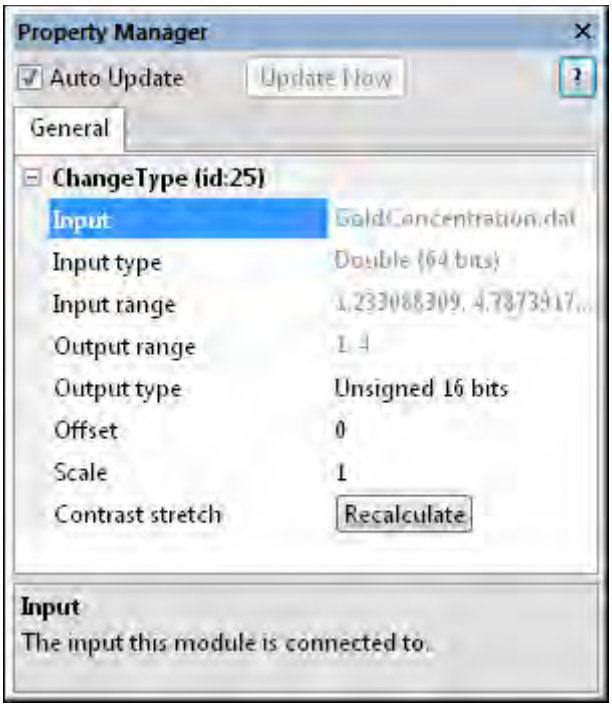
The *ChangeType* module may be connected to *Graphics Output Modules* or *Computational Modules*. An *Info Module* may also be connected to the output node.

Properties

The *ChangeType* module properties are described below.



*Select the ChangeType module in the **Network Manager** to display its properties in the **Property Manager**.*



Use the ChangeType module to change the input or output data type.

Input

The *Input* property shows the source to which the module is connected. This option cannot be changed in the **Property Manager**, but can be changed in the **Network Manager** by changing the module input.

Input Type

The *Input type* is the type of values stored in the input data. Various primitive type choices exist for this option, including *Signed*, *Unsigned*, *Float*, and *Double.*

Input Range

Input range is the range of the input component data.

Output Range

Output range is the range of the output component data.

The message "truncated!" appears after the output range if the input data is too large or small to be represented by the output type after scaling.

Output Type

The Output type is the type of values stored in the output data. Various choices exist for this option, including Signed, Unsigned, Float, and Double. Select from Signed 8 bits, Unsigned 8 bits, Signed 16 bits, Unsigned 16 bits, Signed 32 bits, Unsigned 32 bits, Signed 64 bits, Float (32 bits), and Double (64 bits).

Offset

The *Offset* option defines a linear transformation to convert the input to the output in the scaling transformation equation. The *Offset* adds or subtracts values from the *Input range* to compute the *Output range.*

output = (input+offset) * scale

To change this value, highlight the existing value and enter a new value. The default *Offset* is zero, which means to not offset the values.

Scale

The *Scale* option defines a linear transformation to convert the input to the output in the scaling transformation equation. The *Scale* multiplies or divides values from the *Input range* to compute the *Output range.*

output = (input+offset) * scale

To change this value, highlight the existing value and enter a new value. The default *Scale* is one, which means to not scale the values.

Contrast Stretch

Click the *Recalculate* button next to *Contrast stretch* to scale the input to completely span the range of the output data. This button is disabled for floating point types.

DuplicateFilter

The **Network | Computational | DuplicateFilter** command adds a *DuplicateFilter* module.

The *DuplicateFilter* module removes duplicate data points in a point set. Duplicate data are two or more data points having nearly identical X, Y, and Z coordinates. This module changes the order of the points within the point set, i.e., the output points are sorted on their coordinates. The *DuplicateFilter* properties control the definition of a duplicate point. Several options are available for determining which point, if any, to keep when points are considered duplicates. Some duplicate filter methods create a new artificial point to represent a set of two or more duplicate points, e.g., *Average*. In this case, components of all duplicate points are averaged in order to compute the new point.

Inputs

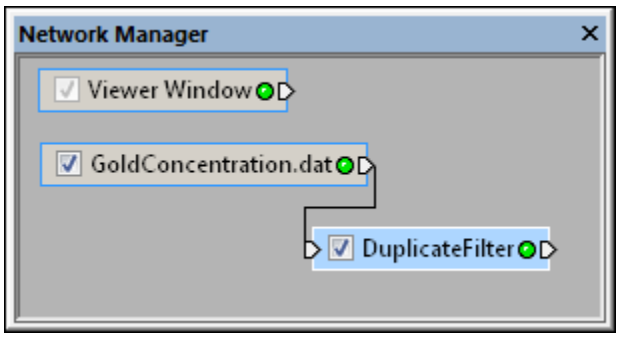
Point data is the input type for the *DuplicateFilter* module.

Outputs

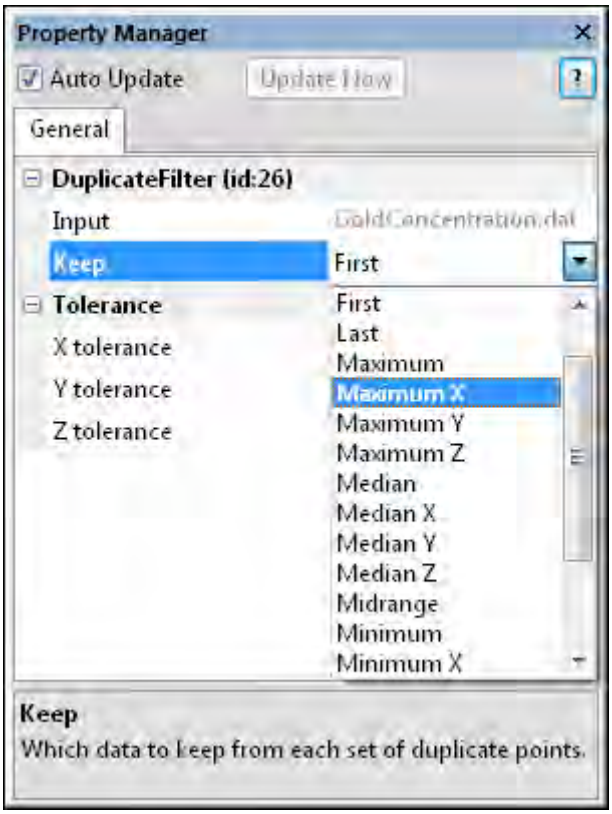
The *DuplicateFilter* module creates point data. It may be connected to the *Graphics Output Modules* or the *Computational Modules*. An *Info Module* may also be connected to the output node.

Properties

The *DuplicateFilter* properties, used to specify what actually constitutes a duplicate point in the original data set, are described below.



*Select the DuplicateFilter module in the **Network Manager** to display its properties in the **Property Manager**.*



Use the Keep option of the DuplicateFilter module to choose which duplicate data points to keep.

Input

The *Input* property shows the source to which the module is connected. This option cannot be changed in the **Property Manager**, but can be changed in the **Network Manager** by changing the module input.

Keep

The *Keep* section specifies which data to keep from each set of duplicate points. The *Keep* options are described below.

All	Do not delete any duplicates
None	Eliminate all of the duplicates
First	Keep the first point (as defined by the order in the data file) from each set of duplicates
Last	Keep the last point (as defined by the order in the data file) from each set of duplicates
Minimum X	Keep the point with the minimum X coordinate
Maximum X	Keep the point with the maximum X coordinate
Median X	Keep the point with the median X coordinate
Minimum Y	Keep the point with the minimum Y coordinate
Maximum Y	Keep the point with the maximum Y coordinate
Median Y	Keep the point with the median Y coordinate
Minimum Z	Keep the point with the minimum Z coordinate
Maximum Z	Keep the point with the maximum Z coordinate
Median Z	Keep the point with the median Z coordinate
Sum	Create an artificial datum at the centroid of the duplicate points with a Z value equal to the sum of the duplicate set's Z values
Average	Create an artificial datum at the centroid of the duplicate points with a Z value equal to the average of the duplicate set's Z values
Midrange	Create an artificial datum at the centroid of the duplicate points with a Z value equal to the midrange of the duplicate observation's Z values halfway between the minimum Z and the maximum Z
Random	Keep a single randomly selected representative point

X Tolerance, Y Tolerance, Z Tolerance

Points separated by a distance less than or equal to the X, Y, and Z tolerances are considered duplicates. Set the tolerances to 0.0 to require duplicate points to have exactly the same coordinates. Set one or more tolerance value to some "small" value to allow slight variations and perform a "fuzzy" match.

ExclusionFilter

The **Network | Computational | ExclusionFilter** command adds an *ExclusionFilter* module.

The *ExclusionFilter* module excludes data points according to a user-specified Boolean function. See the complete list of functions and operators on the *Mathematical Functions* page.

Boolean functions can include the following operators:

- logical operators (AND, OR, XOR, NOT)
- comparison operators (=, <>, <, >, <=, >=)
- IF condition, true_value, false_value or condition? true_value, false_value

Inputs

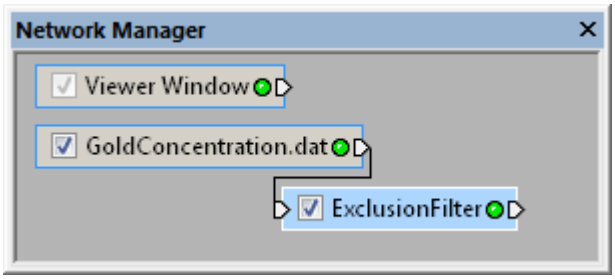
Point data is the input type for the *ExclusionFilter* module.

Outputs

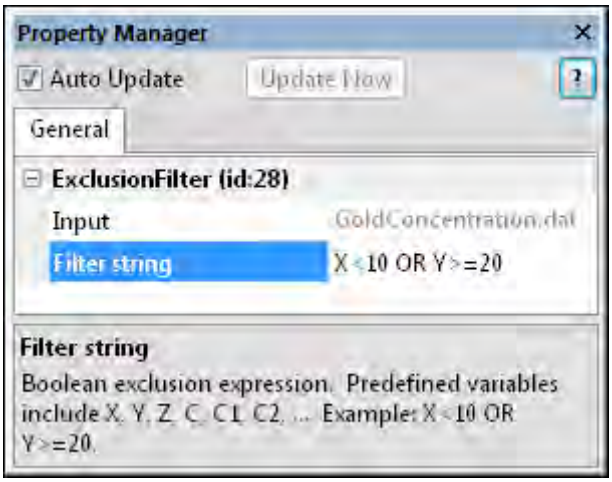
The *ExclusionFilter* module creates point data. It may be connected to the *Graphics Output Modules* or the *Computational Modules*. An *Info Module* may also be connected to the output node.

Properties

The one *ExclusionFilter* property is described below.



*Select the ExclusionFilter module in the **Network Manager** to display its properties in the **Property Manager**.*



Use the ExclusionFilter module to define a Boolean function that excludes data points.

Input

The *Input* property shows the source to which the module is connected. This option cannot be changed in the **Property Manager**, but can be changed in the **Network Manager** by changing the module input.

Filter String

The *Filter string* option contains a Boolean exclusion expression. Predefined variables include X, Y, Z, C, C1, C2, etc. C is another name for C1. Remember that this is an EXCLUSION filter; as such, points matching the expression are excluded and removed from the output.

When you use the AND in the filter, points are only excluded if they meet both conditions.

When you use the OR in the filter, points are only excluded if they meet either filter.

Examples

The exclusion filter can be used to exclude a range of particular values from a scatter plot. The filter string X<10 OR Y>=20 excludes all points with an X coordinate less than 10 or with a Y coordinate greater than or equal to 20.

The exclusion filter can be used to exclude data above or below a specific value. The filter string Z<0 excludes all points with a Z coordinate less than 0.

ExtractPoints

The **Network | Computational | Extract Points** command adds an *ExtractPoints* module.

The *ExtractPoints* module converts points on well paths into points to use for gridding. It will also convert a lattice to a point data set. Users can specify the number of output components, based on the number of input log columns or components in the original lattice.

Inputs

Well data or lattice data is the input type for the *ExtractPoints* module.

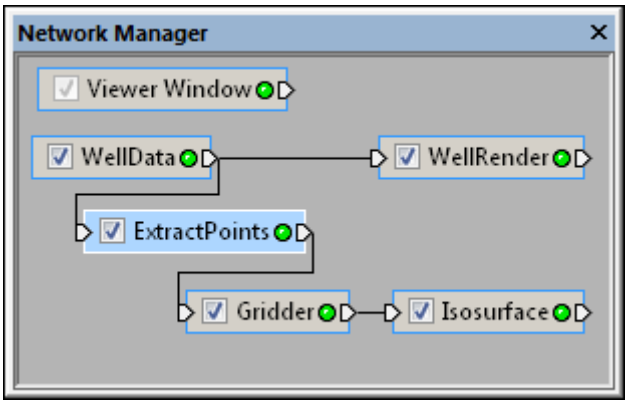
Outputs

The *ExtractPoints* module creates point data. It may be connected to the *Graphics Output Modules* or the *Computational Modules*. An *Info Module* may also be connected to the output node.

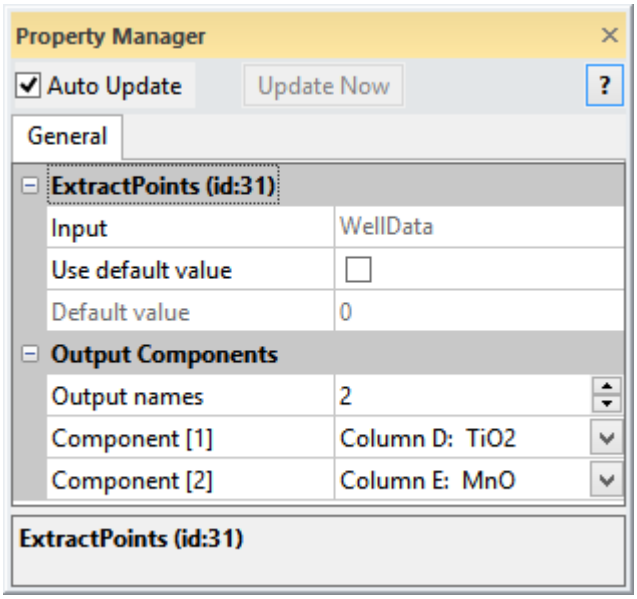
For well data, the output X, Y, Z coordinates and components will be a double. For lattices, the X, Y, Z coordinates will be output as a double, but the components will be output with the same precision as the input components.

Properties

The properties of the *ExtractPoints* module are described below.



*Select the ExtractPoints module in the **Network Manager** to display its properties in the **Property Manager**.*



Use the ExtractPoints module to extract X, Y, Z, C data from well or lattice input.

Input

The *Input* property shows the source to which the module is connected. This option cannot be changed in the **Property Manager**, but can be changed in the **Network Manager** by changing the module input.

Use Default Value

The *Use default value* option substitutes blank values in the well data or lattice with the *Default value*. Check the box next to replace blank values with the specified numeric value. Or, uncheck the box to leave blank values as blank in the output data.

Default Value

After checking the box next to the *Use default value*, the *Default value* specifies the numeric value to use in the output. To change the value, highlight the existing value and type a new number.

Components

Click the ⊞ next to *Output Components* to open the *Output Components* section. The *Output names* sets the number of components being exported. This can be the same as the number of input logs or input components from the lattice or a different number. To change the number of output components, highlight the *Output names* value and type a new value or click the  button to increase or decrease the number of components. The *Output names* is a value between 1 and 20. The maximum *Output names* value is automatically limited by the number of components from the input *WellData* or *Lattice* module.

The *Component[1]* option contains the name of the first component. For well data, the log name is listed. For lattices, the lattice component number is shown. To change the *Component[1]* name, click on the existing option and select the desired value from the list. Each additional component is listed as *Component[2]*, *Component[3]*, etc. The number of components listed matches the *Output names* value.

Filter

The **Network | Computational | Filter** command adds a *Filter* module.

The *Filter* module applies a digital filter to a uniform lattice. The lattice may be two-dimensional (images) or three-dimensional (volumes). Each filter reads the input lattice, performs a particular filtering operation on the data values in the lattice nodes, and sends the results to the output lattice. The input and output lattices are always the same size and type.

Filter module computations include data statistics such as local minimum, maximum, median, average, standard deviation; and image modification such as brightness and contrast.

Inputs

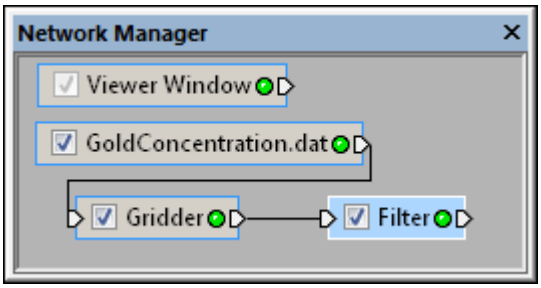
Uniform lattice is the input type for the *Filter* module.

Outputs

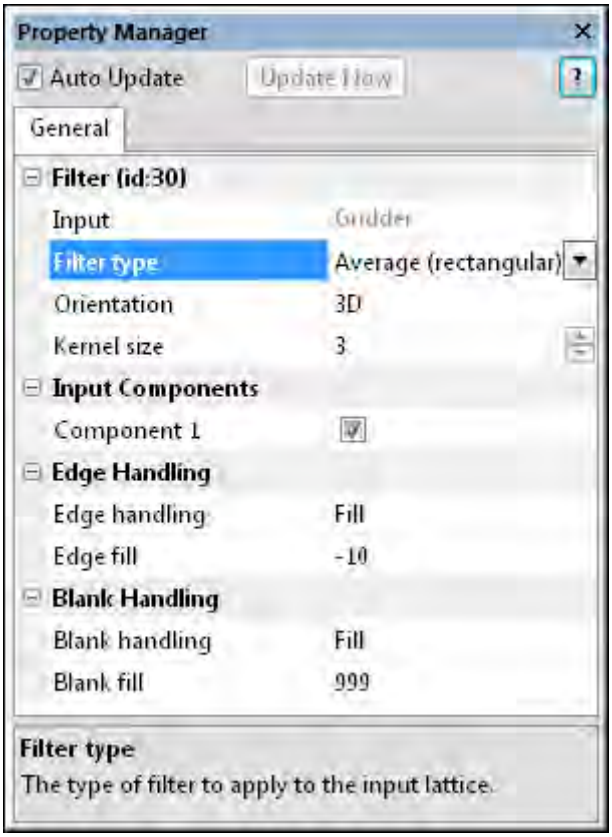
The *Filter* module may be connected to the *Graphics Output Modules* or the *Computational Modules*. An *Info Module* may also be connected to the output node.

Properties

The *Filter* properties are described below.



*Select the Filter module in the **Network Manager** to display its properties in the **Property Manager**.*



Use the Filter module to choose one of many filter types to apply to the input lattice.

Input

The *Input* property shows the source to which the module is connected. This option cannot be changed in the **Property Manager**, but can be changed in the **Network Manager** by changing the module input.

Filter Type

The Filter type is the type of filter to apply to the input lattice. There are six categories of filters: Linear Convolution; Nonlinear - Order Statistics; Nonlinear - Moment Statistics; Nonlinear - Other; Nonlinear - Edge Detection; and Nodal. To change the filter type, click on the existing option and select the desired option from the list.

All of the filters are space-domain filters (currently, there are no frequency-domain filters). Note: This filter is applied to ALL components. Use the *Subset* module to apply filtering to individual components.

Linear Convolution

All linear convolution filters compute weighted averages of the neighboring input lattice nodes. The only differences between the various filters are the shape of the neighborhood and the specific weights used. Consider the computation of the output lattice value at node (a, b, c) using a linear convolution filter with a kernel size *S*.

$$\text{Numerator} = \sum_{ijk} \begin{cases} 0 & \text{if } Z_{in} \text{ is blank} \\ W_{ijk} Z(a+i, b+j, c+k) & \text{otherwise} \end{cases}$$

$$\text{Denominator} = \sum_{ijk} \begin{cases} 0 & \text{if } Z_{in} \text{ is blank} \\ W_{ijk} & \text{otherwise} \end{cases}$$

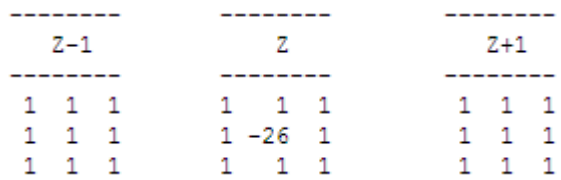
where *Wijk* are the weights defined for the specified filter. The output lattice node value is then

$$Z_{out} = \begin{cases} \frac{\text{Numerator}}{\text{Denominator}} & \text{if } |\text{Denominator}| > 0 \\ \text{Numerator} & \text{otherwise} \end{cases}$$

When we compute the value at node (a, b, c), the weights are functions of the nodal separation distance between (a, b, c) and the neighboring nodes.

Filter Type	Description	Equations
Average (Rectangular)	A moving average with a rectangular neighborhood with dimensions of SxSxS, S being the kernel size. All of the weights in the rectangle are 1.	
Average (Spherical)	A moving average with an axis-aligned spherical neighborhood and axis dimensions of S/2, S/2, S/2, S being the kernel size. All of the weights in the sphere are 1.	

Distance	A distance weighted averaging filter. The resulting isoweight lines are concentric ellipsoids.	$W_{ijk} = 1 - \min\left(\sqrt{D_i^2 + D_j^2 + D_k^2}, 1\right)$ where Wijk represents the weights defined for the specified filter and $D_i = \frac{i - \lfloor S/2 \rfloor}{\lfloor S/2 \rfloor}$ $D_j = \frac{j - \lfloor S/2 \rfloor}{\lfloor S/2 \rfloor}$ $D_k = \frac{k - \lfloor S/2 \rfloor}{\lfloor S/2 \rfloor}$
Distance (Inverse)	A distance weighted averaging filter. By definition, the central weight is 1.0. The resulting iso-weights are concentric spheres.	$W_{ijk} = \begin{cases} 1 \\ \frac{1}{\max\left(\sqrt{d_i^2 + d_j^2 + d_k^2}, 1\right)} \end{cases}$ where $d_i = i - \lfloor S/2 \rfloor$ $d_j = j - \lfloor S/2 \rfloor$ $d_k = k - \lfloor S/2 \rfloor$
Distance ' Inf Norm'	A distance weighted averaging filter. The resulting iso-weight lines are concentric shoeboxes.	$W_{ijk} = 1 - \max(D_i, D_j, D_k)$ where $D_i = \frac{i - \lfloor S/2 \rfloor}{\lfloor S/2 \rfloor}$ $D_j = \frac{j - \lfloor S/2 \rfloor}{\lfloor S/2 \rfloor}$ $D_k = \frac{k - \lfloor S/2 \rfloor}{\lfloor S/2 \rfloor}$
Gauss	A distance weighted averaging filter. The Gauss filter smooths or blurs an image by performing a convolution operation with a Gaussian filter kernel. The distance weighting function is given by a Gaussian bell-shaped curve. The resulting iso-weight lines are concentric ellipsoids.	$W_{ijk} = e^{-\alpha\left(D_i^2 + D_j^2 + D_k^2\right)}$ where $D_i = \frac{i - \lfloor S/2 \rfloor}{\lfloor S/2 \rfloor}$ $D_j = \frac{j - \lfloor S/2 \rfloor}{\lfloor S/2 \rfloor}$ $D_k = \frac{k - \lfloor S/2 \rfloor}{\lfloor S/2 \rfloor}$

Laplacian Edge Detect	Used for edge detection, this implementation uses a 3x3x3 kernel. Mathematically, there exist an infinite number of valid possibilities for the convoluted weights; those used in the filter include those illustrated on the right.	
Unsharp Mask	This filter first blurs an image and then subtracts the blurred image from the original image. The difference between the blurred image and the original image (known as the "mask") is then added back to the original image. The resulting combined image appears sharper to the human eye. The sharpness parameter, *c*, is used to compare the original image against the blurred image.	Original image: (W=c/(2c-1)) Blurred image: (W=(1-c)/(2c-1))

Nonlinear - Order Statistics

The nonlinear filters are not weighted averages of the neighboring input lattice values.

Filter Type	Description
Interquartile Range	A nonlinear filter that retains the local interquartile range (75th minus 25th percentile) in a moving shoebox-shaped neighborhood.
Maximum (dilation)	A nonlinear filter that retains the maximum in a moving shoebox-shaped neighborhood. The result is similar to a morphological dilation operation.
Median	A simple, nonlinear, edge-preserving smoothing filter that retains the median (50th percentile) in a moving shoebox-shaped neighborhood. Median filters are most effective on high-amplitude noise that has a low probability occurring. There are two ways to control the amount and scale of the removed noise: variation of kernel size; and multiple application of the filter.

Minimum (erosion)	A nonlinear function that retains the minimum in a moving shoebox-shaped neighborhood. The result is similar to a morphological dilation operation.
Quartile (lower)	A nonlinear function that retains the lower quartile (25th percentile) in a moving shoebox-shaped neighborhood.
Quartile (upper)	A nonlinear function that retains the upper quartile (75th percentile) in a moving shoebox-shaped neighborhood.
Range	A nonlinear function that retains the local data range (maximum minus the minimum) in a moving shoebox-shaped neighborhood.

Nonlinear - Moment Statistics

Filter Type	Description	Equation
Central Moment	A nonlinear filter that retains the local data central moment in a moving shoebox-shaped neighborhood. The filter calculates the Nth centralized moment of the data in a gliding window. The second moment (power = 2) is, therefore, the local variance in the data. For some data sets, this can be used to mask out noisy regions or to detect edges.	$\text{Central Moment} = \frac{1}{N-1}\sum_{ijk}\begin{cases}0\\(Z(a+i,b+j,c+k)-\overline{Z})^p\end{cases}$ where *p* is a positive integer.
Coefficient of Variation	A nonlinear filter that retains the local data coefficient of variation (std. dev. / average) in a moving shoebox-shaped neighborhood.	$\text{Coefficient of Variation} = \frac{\text{Standard Deviation}}{\overline{Z}}$
Standard Deviation	A nonlinear filter that retains the local data standard deviation in a moving shoebox-shaped neighborhood. The standard deviation is the square root of the variance, which is computed in three steps, as follows: First, count the number of non-blank input lattice nodes in the neighborhood. With a neighborhood height *S* and width *T*, calculate *N* as shown to the right. Second, calculate the local average $\overline{Z}$ as	$\text{Standard Deviation} = \sqrt{\text{Variance}}$

	shown on the right. Third, compute the local variance as shown on the right.	

Nonlinear - Other

Filter Type	Description	Equation
Mean Removal	A simple high-pass filter that computes and then subtracts the neighborhood mean, including the voxel itself. This filter emphasizes outliers in the lattice.	$Z_{out} = Z_{in} - \overline{Z}$ where $\overline{Z}$ represents the mean
Median Difference	A nonlinear filter that retains the local median difference (z - median) in a moving shoebox-shaped neighborhood. The median filter is a simple edge-preserving smoothing filter. This filter emphasizes outliers in the lattice.	$Z_{out} = Z_{in} - B$ where B represents the median
Rank	A nonlinear filter that retains the rank in the local neighborhood for the current voxel.	
Threshold Averaging	A nonlinear filter that retains the local "threshold" average in a moving shoebox-shaped neighborhood.	$Z_{out} = \begin{cases} Z_{in} & \text{if } \lvert Z_{in} - A \rvert \leq \text{Threshold} \\ A & \text{if } \lvert Z_{in} - A \rvert > \text{Threshold} \end{cases}$ where A represents the average value
Threshold Crossing	A nonlinear filter that sets the current voxel to 1 if a value crosses the threshold somewhere within the local neighborhood. Without a crossed threshold, it sets the voxel to 0.	
Zero Crossing	A nonlinear filter that sets the current voxel to 1 if there is a change within the local neighborhood. Without a change, it sets the voxel to 0. This filter is usually used in conjunction with a second derivative filter (eg, Laplacian).	

Nonlinear - Edge Detection

Filter Type	Description

Prewitt	A nonlinear filter that applies the Prewitt edge-detector to a 3x3x3 neighborhood. This is a Prewitt "first derivative" filter. This is also a generalization of the 2D Sobel edge-detection filter, as suggested by Nikolaidis (2001, p 96) and uses a weighted central scheme to compute the directional derivatives but averages over nine neighborhood voxels to reduce noise.
Sobel Max	A nonlinear filter that applies the Sobel edge-detector to a 3x3x3 neighborhood. This is an "edge-detection" filter that keeps the maximum of the absolute value of the gradient components. This is a generalization of the 2D Sobel edge-detection filter as suggested by Nikolaidis (2001, p 96) and uses a weighted central scheme to compute the directional derivatives but averages over nine neighboring voxels to reduce noise. See also: Crane (1997, p 86), Gonzalas (1983, p 336), and Pitas (2000, p 242).
Sobel Norm	A nonlinear filter that applies the Sobel edge-detector to a 3x3x3 neighborhood. This is an "edge-detector" filter that keeps the Euclidean norm of the gradient components. This is also a generalization of the 2D Sobel edge-detection filter, as suggested by Nikolaidis (2001, p 96) and uses a weighted central scheme to compute the directional derivatives but averages over nine neighboring voxels to reduce noise. See also: Crane (1997, p 86), Gonzalas (1983, p 336), and Pitas (2000, p 242).

Nodal

Filter Type	Description	Equation
Bounding	The output is filtered on a voxel-by-voxel basis. If the value is above the specified upper bound, the value is set equal to the upper bound; if the value is below the lower bound, it is set equal to the lower bound. This filter operates node by node (ie, Ni = Nj = Nk = 1). Blanks and edges have no impact on this filter (ie, blanks stay blanked).	
Brightness & Contrast	Modifies the image brightness and contrast. The range increases by a factor of C and the midrange shifts by a factor of B. The output is filtered on a voxel-by-voxel basis. This filter applies an affine transformation node by node (ie, Ni = Nj = Nk = 1). Blanks and edges have no impact on this filter (ie, blanks stay blanked).	$Z_{out} = C\,(Z_{in} - M) + M + B$ where B represents brightness, C represents contrast, and M is the global midrange for the entire lattice.
Gamma Correction	This filter performs gamma correction on a voxel-by-voxel basis: Gamma=1 performs no adjustment; Gamma<1 reduces gamma; Gamma>1 increases gamma. Blanks and edges have no impact on this filter (ie, blanks stay blanked).	$Z_{out} = \left(\frac{Z_{in}}{M}\right)^{1/gamma} M$ where M is the maximum intensity throughout the entire lattice.

Orientation

Use the *Orientation* option to specify the orientation of the filter. The first three choices— *XY planes*, *XZ planes*, and *YZ planes*— orient the filter in the specified direction and apply it to all slices of the lattice. The *3D* option applies a full three-dimensional filter. Note: when this module is connected to a two-dimensional lattice, the three-dimensional orientation is disabled and this choice defaults to the two-dimensional lattice orientation. To change the orientation, click on the existing option and select the desired option from the list.

Kernel Size

The *Kernal size* is the the size of the convolution kernel. The kernel is centered on the node being filtered and defines the neighborhood of surrounding nodes used to calculate the center node. If the kernel size is N, then the filtering kernel is NxNxN nodes in three dimensions. Not all filters use a convolution kernel, and for these the kernel size property is hidden. Kernel sizes must be odd to maintain symmetry. To change the *Kernal size*, highlight the existing value and type a new value or click the  button to increase or decrease the neighborhood size.

The neighborhood of an output lattice node is a cubic sub- array of nodes in the input lattice that is centered on the corresponding input lattice node. It has three equal dimensions, known as the kernel size. It must be noted that this concept of a neighborhood does not disallow that application of a lattice filter to a two-dimensional lattice. In the case of a two-dimensional lattice, one of the three dimensions is simply equal to 1. Since the neighborhood is centered on a node, the width, height, and depth must all be odd numbers. For example, if the kernel size is 3, the neighborhood of the output lattice node at (21, 31, 16) is the following rectangular sub-array of 27 input lattice nodes:

(20, 36, 15) (21, 36, 15) (22, 36, 15)
(20, 35, 15) (21, 35, 15) (22, 35, 15)
(20, 34, 15) (21, 34, 15) (22, 34, 15)
(20, 36, 16) (21, 36, 16) (22, 36, 16)
(20, 35, 16) (21, 35, 16) (22, 35, 16)
(20, 34, 16) (21, 34, 16) (22, 34, 16)
(20, 36, 17) (21, 36, 17) (22, 36, 17)
(20, 35, 17) (21, 35, 17) (22, 35, 17)
(20, 34, 17) (21, 34, 17) (22, 34, 17)

If kernel size of the neighborhood is represented by S, then the number of nodes in the neighborhood equals SxSxS. Furthermore, the nodes in the neighborhood of node (a,b,c) can be enumerated as $\{Z(a + i, b + j, c + k)\}$ where

$i = -[S/2], \ldots, [S/2]$
$j = -[S/2], \ldots, [S/2]$
$k = -[S/2], \ldots, [S/2]$

where [S/2] represents the largest integer less than or equal to S/2.

Components to Filter

Click the ⊞ next to the *Input Components* to open the components list. Check the box next to the *Component* to pass the checked component through the filter. Unchecked components pass through the filter unaltered (unfiltered). This option allows the alpha channel of RGBA images to be excluded since this is not usually desired.

Edge Handling

Click the ⊞ next to the *Edge Handling* to open the options for handling data near the edge of the lattice.

Use the *Edge handling* option to specify how to handle edge effects when the kernel overhangs the edge of the lattice. Many filters, e.g., convolution filters, require a neighborhood of data around the lattice node being filtered. When filtering nodes along the edge, the neighborhood is incomplete on one or more sides. To change the edge handling, click on the existing option and select the desired option from the list. These options specify how to handle this condition:

- *Blank* - Any lattice node with a neighborhood that overlaps more than one edge is blanked. For example, if the neighborhood size is 3x3x3, then every application of the filter blanks one plane (sheet) of nodes on each lattice face. Every application of the filter shrinks the active lattice by 2 rows, 2 columns, and 2 planes. Similarly, a 5x5x5 filter neighborhood blanks 4 rows, 4 columns, and 4 planes.
- *Ignore* - Only the available neighborhood is used; calculations involving off-lattice nodes are ignored.
- *Replicate* - Voxels at the edge of the input lattice are replicated. For example, consider a lattice with Ni rows, Nj columns, and Nk planes. If the specified coordinate (a,b,c) is off the edge of the lattice, the rules are $Z(a,b,c) = Z(i,j,k)$ where $a = \min(\max(i,0), Ni-1)$, $b = \min(\max(j,0), Nj-1)$, and $c = \min(\max(k,0), Nk-1)$.
- *Mirror* - Set off-lattice nodes to the symmetrically reflected node on the other side of the node being filtered; that is, treat the out-of-range coordinates as a mirror (reverse) of the corresponding in-range coordinates. Consider a lattice with Ni rows, Nj columns, and Nk planes. If the specified coordinate (a,b,c) is off the edge of the lattice, the rules are $Z(a,b,c) = Z(i,j,k)$ where

$$i = \begin{cases} -a & \text{if } a < 0 \\ a & \text{if } 0 \le a < Ni \\ 2Ni - 2 - a & \text{if } a \ge Ni \end{cases}$$

$$j = \begin{cases} -b & \text{if } b < 0 \\ b & \text{if } 0 \le b < Nj \\ 2Nj - 2 - b & \text{if } b \ge Nj \end{cases}$$

$$k = \begin{cases} -c & \text{if } c < 0 \\ c & \text{if } 0 \le c < Nk \\ 2Nk - 2 - c & \text{if } c \ge Nk \end{cases}$$

- *Cyclic wrap* - Set off-lattice nodes by wrapping around to the opposite side of the lattice in three dimensions. If you go off the lattice on the right face, you come back on the left; if you go off the top, you come back on the bottom. This method is consistent with Fourier analysis which does this implicitly. For a lattice with Ni rows, Nj columns, and Nk planes, the rules are $Z(a,b,c) = Z(i,j,k)$ where

$$i = \begin{cases} a + Ni & \text{if } a < 0 \\ a & \text{if } 0 \le a < Ni \\ a - Ni & \text{if } a \ge Ni \end{cases}$$

$$j = \begin{cases} b + Nj & \text{if } b < 0 \\ b & \text{if } 0 \le b < Nj \\ b - Nj & \text{if } b \ge Nj \end{cases}$$

$$k = \begin{cases} c + Nk & \text{if } c < 0 \\ c & \text{if } 0 \le c < Nk \\ c - Nk & \text{if } c \ge Nk \end{cases}$$

- *Fill* - Fill beyond the edge with a user-specified constant. A common value for the fill is the arithmetic average of the lattice. This is the default method. For a lattice with Ni rows, Nj columns, and Nk planes, the rules are

  $$Z(a,b,c) = \begin{cases} Z(a,b,c) & \text{if } 0 \le a < Ni \text{ and } 0 \le b < Nj \text{ and } 0 \le c < Nk \\ A & \text{otherwise} \end{cases}$$

  where A is the user-specified constant.

Edge Fill

When the *Edge handling* is set to *Fill*, the *Edge fill* option becomes available. This is the value used to apply to the edge of the lattice. To change the value, highlight the existing value and type the desired value.

Blank Handling

Click the ⊞ next to the *Blank Handling* to open the options for handling blanked nodes in the lattice.

Use the *Blank handling* option to determine how to handle blanked nodes within the lattice. To change the way blank values are handled, click on the existing option and select the desired option from the list. Options include:

- *Expanded* - Blanked nodes and all nodes dependent on them are blanked in the output lattice. If the neighborhood of an output lattice node contains one or more blanked nodes in the input lattice, then the output node is blanked. Like blanking on the edge, this approach leads to blanked areas that grow with every application of the filter.
- *Leave alone* - Blank every output lattice node for which the corresponding input lattice node is blank. When the corresponding lattice node is not blank but a neighboring lattice node is blank, modify the filter to ignore the blank (ie, remove the blank node from the neighborhood). For example, if the filter called for computing the median value in the neighborhood, the blanked values would not be considered when determining the median. This keeps the blanked areas consistent; however, it may also cause edge artifacts for some filter types.
- *Ignore* - Filter across the blanks when possible by not including them in the filtering calculations (ie, remove the blank nodes from the neighborhood). For example, if the filter called for computing the median value in the neighborhood, the blanked values would not be considered when determining the median. This option is similar to the *Leave Alone* option; however, *Ignore* does not blank the output lattice nodes corresponding to blank input lattice nodes. *Ignore* is essentially simultaneous filtering and interpolation. Every application of the filter sees a shrinking of the blanked regions since the only blank output lattice nodes are those with completely blank neighborhoods.

- *Fill* - Set blanked nodes to the specified fill value. This is the default setting.

Blank Fill

When the *Blank handling* is set to *Fill*, the *Blank fill* option becomes available. This is the value used to substitute for values in the lattice that are blanked. To change the value, highlight the existing value and type the desired value.

Filter References

Crane, R. (1990) *A Simplified Approach to Image Processing*. Prentice-Hall PTR, Upper Saddle River, NJ, 317 pp. ISBN 0-13-226416-1.

Gonzalas, R, & Wintz, P. (1983) *Digital Image Processing*. Addison-Wesley Publishing Company, Reading, MA, 431 pp. ISBN 0-201-02597-3.

Nikolaidis, N, & Pitas, I. (2001) *3-D Image Processing Algorithms*. John Wiley & Sons, New York, NY, 176 pp. ISBN 0-471-37736-8.

Pitas, I. (2000) *Digital Image Processing Algorithms and Applications*. John Wiley & Sons, New York, NY, 419 pp. ISBN 0-471-37739-2.

Gradient

The **Network | Computational | Gradient** command adds a *Gradient* module.

The *Gradient* module computes a gradient field from a single component of a two- or three-dimensional lattice. A gradient is a three-dimensional vector pointing in the direction of greatest slope. The output lattice contains three-component data at each lattice node. A centered difference algorithm is used to calculate the gradient. The output lattice geometry is identical to the input lattice geometry.

Inputs

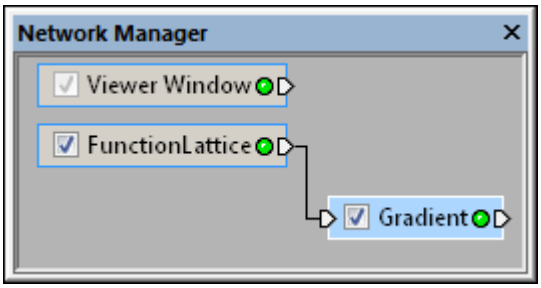
Lattice is the input type for the *Gradient* module.

Outputs

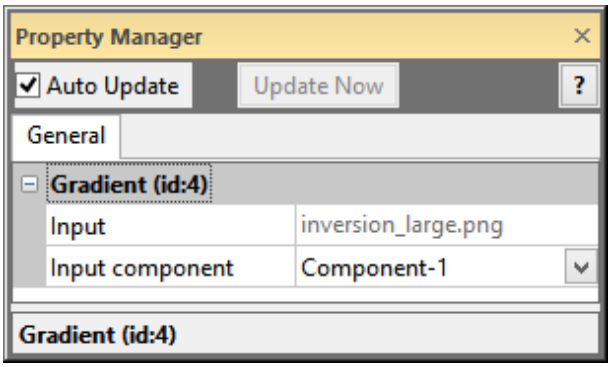
The *Gradient* module creates a lattice. It may be connected to the *Graphics Output Modules* or the *Computational Modules*. An *Info Module* may also be connected to the output node.

Properties

The *Gradient* properties are described below.



*Select the Gradient module in the **Network Manager** to display its properties in the **Property Manager**.*



Use the Gradient module to create a gradient field from a component of the input lattice.

Input

The *Input* property shows the source to which the module is connected. This option cannot be changed in the **Property Manager**, but can be changed in the **Network Manager** by changing the module input.

Input Component

The *Input component* shows the component from the input lattice used to calculate the gradient. To change the component, click the current selection and select the desired component from the list. The geometry of the output lattice mirrors the geometry of the input lattice.

For uniform and rectilinear input lattices, the components of the output lattice are:

- dT/ dU, dT/ dV, dT/ dW if the input lattice has three dimensions,
- dT/ dU, dT/ dV, 0 if the input lattice has two dimensions, and
- dT/ dU, 0, 0 if the input lattice has one dimension.

The first derivatives are calculated using central differences, if possible, and forward or backward differences as necessary.

For curvilinear input lattices, the components of the output lattice are always dT/ dU, dT/ dV, dT/ dW .

Unlike the cases of uniform and rectilinear lattices, a one- or two-dimensional gradient lattice makes no sense with a curvilinear geometry. The components of the local gradient are computed by fitting a local linear model to the current node and its six cardinal neighbors: T(U, V, W) = aU + bV + cW + d. With this local linear model, the gradient vector is simply (a, b, c).

Gridder

The **Network | Computational | Gridder** command adds a *Gridder* module.

The *Gridder* module interpolates scattered point data onto a uniform lattice. This type of lattice is used to create several types of output graphics, including *Isosurfaces*, *VolRenders*, *FaceRenders*, *VectorPlots*, and *StreamLines*. Other graphics can be created from slices through a uniform lattice, including *Contours*, *HeightFields*, *OrthoImages*, and *ObliqueImages*.

A uniform lattice is a one-, two-, or three-dimensional orthogonal array of data points arranged in the XYZ directions with points equally spaced in each direction. The distance between data points in the X, Y, and Z directions is the same throughout the lattice, but the X separation distance is not necessarily the same as the Y or Z separation distances.

The *Gridder* module is also used to calculate data values in areas where there are no data.

The range and resolution of the output lattice may be specified along with the interpolation method and associated parameters. Since gridding can take quite a while to execute, it is necessary to click the *Begin Gridding* button to start the process. No output is generated until this is done. The *Gridder* module in the **Network Manager** will display a yellow LED indicator light . After specifying the gridding properties and clicking the *Begin Gridding* button, the indicator light changes to green to specify that the module is up to date and ready to add addition modules.

Inputs

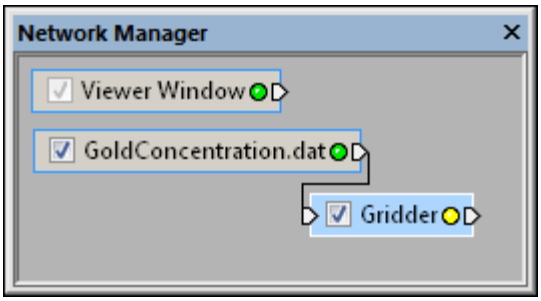
Point data is the input type for the *Gridder* module.

Outputs

The *Gridder* module creates a uniform lattice. It may be connected to the *Graphics Output Modules* or the *Computational Modules*. An *Info Module* may also be connected to the output node.

Properties

The *Gridder* module properties are described below.



*Select the Gridder module in the **Network Manager** to display its properties in the **Property Manager**.*

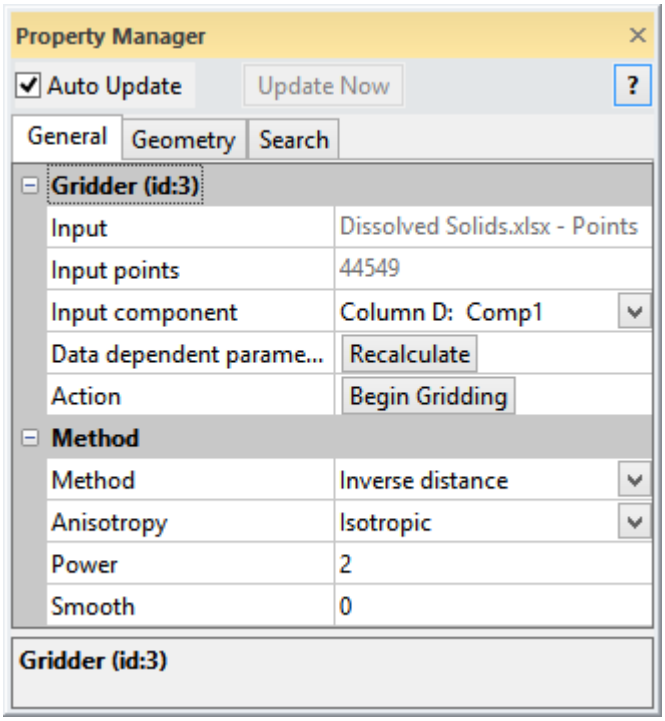
The *Gridder* module contains the following tabs in the **Property Manager:**

General

Geometry

Search

General Options



*View and change the gridding method in the **Property Manager**. Click the Begin Gridding button to begin the gridding process.*

Input

The *Input* property shows the source to which the module is connected. This option cannot be changed in the **Property Manager**, but can be changed in the **Network Manager** by changing the module input.

Input Points

The *Input points* item displays the number of points in the point set.

Input Component

The *Input component* property is shown if the input set has more than one component. The *Input component* property specifies which data component to grid if the input data set has more than one component. To change which component is used by the *Gridder* module, click the current selection in the *Input component* field and select the desired component from the list. The *Input component* list contains the components specified in the *Data Source* properties for the module displayed by the *Input* property.

The output lattice is always a single component, i.e., the *Gridder* module grids only one component at a time, even if the input data set has multiple components.

Data Dependent Parameters

Click the *Recalculate* button to calculate default values based on the input data. This function is helpful when the input data has changed or the *Gridder* module is connected to a new data set and the old values are no longer appropriate.

Action

Click the *Begin Gridding* button to begin the gridding process. The Progress dialog appears and displays the gridding progress.

The *Gridder* module indicator LED in the **Network Manager** glows yellow when the *Gridder* module needs to be updated. Once the *Begin Gridding* button has been clicked and the module finishes gridding, the light changes to green.

Method

Click the ⊞ next to *Method* to open the *Method* section where gridding options are set.

The differences between gridding methods are in the mathematical algorithms used to compute the weights during lattice node interpolation. Each method can result in a different representation of the data. You may want to test each method with a known data set to determine the method that provides you with the most satisfying interpretation of the data.

The default method is *Inverse distance*. This method produces the best results in most cases. To change the method, click on the existing option and select the desired method from the list. Choose *Data metric*, *Inverse distance*, or *Local polynomial* as the interpolation method to use for generating the output lattice.

Metric

The *Metric* option is available when the *Method* is set to *Data metric*. The *Metric* controls the type of data metric being calculated by the gridding algorithm. To change the metric, click on the existing option and select the desired option from the list.

Order

The *Order* option is available when the *Method* is set to *Local polynomial*. The order is the local polynomial order for the interpolation. The *Order* is 0, 1, 2, or 3. The larger the order, the more complex the local polynomial equation can be. To change the *Order*, highlight the existing value and type the new value or click the  to increase or decrease the order.

Power

The *Power* option is available when the *Method* is set to *Inverse distance* or *Local polynomial*. The power is the exponent in the inverse distance weighting formula or the local polynomial weighting formula. To change the power, highlight the existing value and type a new value.

Smooth

The *Smooth* option is available when the *Method* is set to *Inverse distance*. The smooth value controls the amount of smoothing in the interpolation. The higher the value, the more smoothing occurs. To change the value, highlight the existing value and type the desired value.

Anisotropy

The *Anisotropy* property of the *Inverse distance* gridding method lets you specify that data points in one direction from the lattice node have a higher weight than points in another direction. Points closer to the grid node are usually given more weight than points farther from the grid node, regardless of direction. If the points in one direction have more similarity than points in another direction, it is advantageous to give points in a specific direction more weight in determining the value of a lattice node. The relative weighting is defined by the anisotropy lengths of the three axes of the search ellipse or ellipsoid.

The available options for the *Anisotropy* setting are *Isotropic*, *Anisotropic*, and *General*. Isotropic sets all directions to equal weighting. Anisotropic allows an *Ellipse* to be specified to set the direction of weighting. The ellipse axes are parallel to the X, Y, and Z axis directions. General allows each axis of the 3D ellipse to be manually altered. The ellipse is specified with vectors defined by the ray going from 0, 0, 0 to the X, Y, Z values specified. To change the *Anisotropy* setting, click on the existing option and select the desired option from the list.

Ellipse

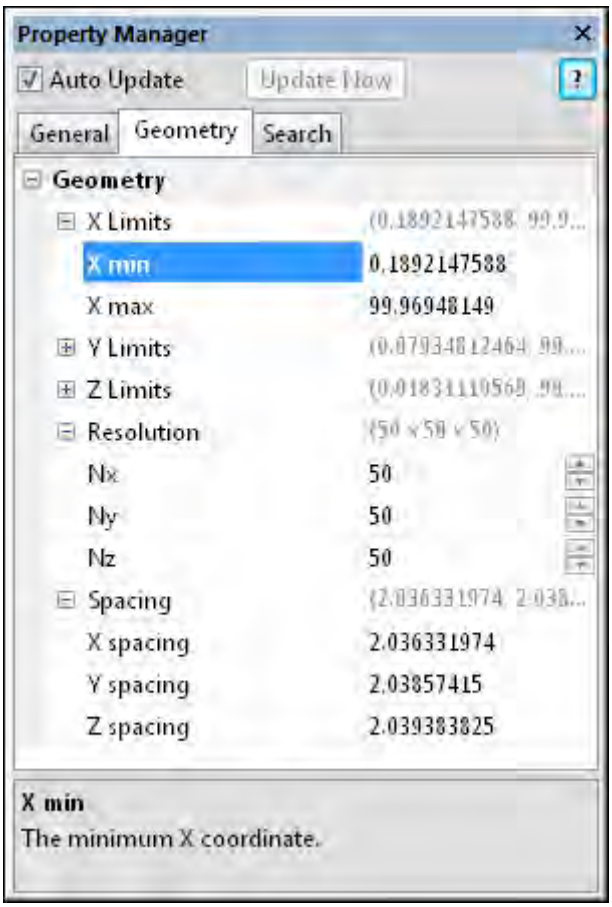
When the *Anisotropy* is set to *Anisotropic* or *General*, the *Ellipse* option is available. To open the *Ellipse* section, click the ⊞ next to *Ellipse*.

For an *Anistropic* ellipse, the options available are *X length*, *Y length*, and *Z length*. To change these values, highlight the existing value and type a new value. The lengths are relative and in data file units. For example, if you specify the *X length* equal to 2, the *Y length* equal to 4, and the *Z length* equal to 1, the Y direction is given the most weight and the Z direction is given the least weight. Points that are 4 units away in the Y direction have the same weight as points that are 1 unit away in the Z direction.

For a *General* ellipse, open the *First Axis*, *Second Axis*, or *Third Axis* sections by clicking the ⊞ next to the section name. Each section has an *X*, *Y*, and *Z* value, indicating the direction of each ellipse axis. To change the value, highlight the existing value and type the desired value.

Gridder Module - Geometry Page

The *Gridder* module **Geometry** page displays the extents of the grid being created. To open the **Geometry** page, click on the *Gridder* module in the **Network Manager**. In the **Property Manager**, click on the **Geometry** tab.



This is an example of the information displayed in the Property Manager on the Geometry tab for a Gridder module.

The **Geometry** tab contains the limits, resolution, and spacing for the lattice that is being created. Click the ⊞ next to each section name to open that section.

X Limits

The *X limits* item displays the limits in the X direction for the lattice being created. The *X min* and *X max* coordinates define the extent of *Gridder* calculations and the extent of the output graphics attached to the *Gridder* module in the X direction. These coordinates are taken from the input data file by default; you can enter different coordinates to calculate values for a larger or smaller area. To change the value, highlight the existing value and type a new value.

Y Limits

The *Y limits* item displays the limits in the Y direction for the lattice being created. The *Y min* and *Y max* coordinates define the extent of *Gridder* calculations and the extent of the output graphics attached to the *Gridder* module in the Y direction. These coordinates are taken from the input data

file by default; you can enter different coordinates to calculate values for a larger or smaller area. To change the value, highlight the existing value and type a new value.

Z Limits

The *Z limits* item displays the limits in the Z direction for the lattice being created. The *Z min* and *Z max* coordinates define the extent of *Gridder* calculations and the extent of the output graphics attached to the *Gridder* module in the Z direction. These coordinates are taken from the input data file by default; you can enter different coordinates to calculate values for a larger or smaller area. To change the value, highlight the existing value and type a new value.

Resolution

Resolution is the number of nodes in the lattice. To change the number of nodes in any direction, highlight the value next to the *Nx*, *Ny*, or *Nz* option and type a new value or click the  to increase or decrease the number of nodes in that direction. The number of lattice nodes defaults to 50 in each of the X, Y, and Z directions.

Spacing

Spacing displays the distance between grid nodes in the x, y, and z directions. When the *Resolution* or *Limits* values change, the *Spacing* values automatically change. To change the *Spacing* value, highlight the number next to *X spacing*, *Y spacing*, or *Z spacing* and type a new value. The *Resolution* will automatically update.

For some data sets, additional resolution is needed to delineate closely spaced data points, especially in the Z or depth direction.

An increase in the number of lattice nodes provides more resolution, but it also increases the amount of time needed to grid the lattice and to draw output graphics. It also increases the amount of memory needed to display graphics on-screen and to save the graphics to a .VOXB network file.

Gridder Module - Search Page

The *Gridder* module **Search** page displays the search options for the lattice being created. To open the **Search** page, click on the *Gridder* module in the **Network Manager**. In the **Property Manager**, click on the **Search** tab.



This is an example of the information displayed in the Property Manager on the Search tab for a Gridder module.

The **Search** tab contains the search type and search options of the original data that can be used to create the lattice.

Search Type

The *Search type* controls the type of search used to locate input data at each grid node for gridding. To change the search type, click on the existing option and select the desired option from the list. Available options are *All data*, *Simple*, *Anisotropic*, or *General* as the search type to use for locating the closest input data points to each calculated node. To see other search options, click the ⊞ next to *Search type*.

The *All data* search type uses all the data points in the data set to calculate each lattice node.

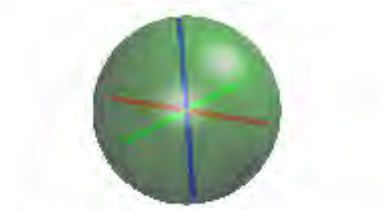
Number of Points

The *Min count* and *Max count* properties of the *Simple*, *Anisotropic,* and *General* search types let you specify the minimum and maximum number of points to search in the search neighborhood. If fewer data are found in the search neighborhood than specified in the *Min count* property, the lattice node is blanked. If more data are found in the search neighborhood than specified in the *Max*

count property, the value of *Max count* is the number of points that are used in the search. Only those points closest to the lattice node are included in the interpolation calculation.

Radius

When the *Search type* is set to *Simple*, the gridding process searchs the data from the location of the grid node to the radius value. **Voxler** searches the same distance in all directions from the lattice node. This creates a spherical search ellipse of *Radius* size. The size is in data units. To change the *Radius*, highlight the existing value and type a new value.



The Simple search type produces a spherical search neighborhood.

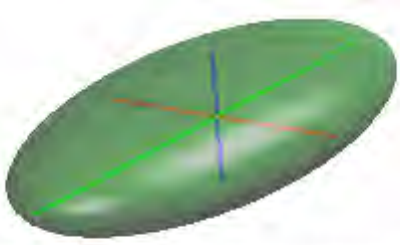
Search Ellipse (Ellipsoid)

When the *Search type* is set to either *Anisotropic* or *General*, the *Search Ellipse* options become available. The *Search Ellipse* defines the local neighborhood of points to consider when interpolating each lattice node. **Voxler** uses the term *search ellipse* when referring to the three-dimensional ellipsoid. This defines the distance in data units from the lattice node that **Voxler** looks to find data points when calculating the value of the node. Data points outside the search ellipse are not considered during interpolation.

When one axis of the search ellipse is longer than the others, it does not imply that greater weight is applied to that direction. Use the *Anisotropy* settings described above to apply additional weight to points in the long direction of the axis.

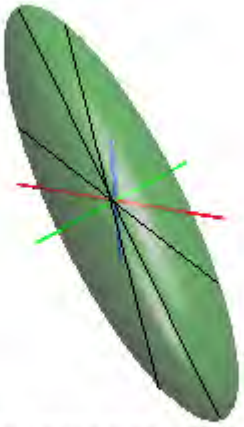
The *Anisotropic* search type lets you specify the lengths of the three orthogonal axes of the search ellipse to create a search neighborhood shaped like a three-dimensional ellipsoid. **Voxler** searches further for data points in the direction of the longer axes. The *Anisotropic* search type does not apply different weight to points in the direction of the longer axes. Use *Inverse distance* as the gridding *Method* and set the *Anisotropy* settings on the **General** tab to apply more weight to points in a particular direction.

For an *Anistropic* search type, the options available are *X length*, *Y length*, and *Z length*. To change these values, highlight the existing value and type a new value. The lengths are relative and in data file units. For example, if you specify the *X length* equal to 2, the *Y length* equal to 4, and the *Z length* equal to 1, the Y direction is searched the furthest and the Z direction is searched the least far for data points.



The Anisotropic search type lets you specify different values for the X, Y, and Z search directions to produce a search neighborhood shaped like an ellipsoid.

The *General* search type lets you specify the X, Y, and Z lengths of three non-orthogonal axes to create a search neighborhood shaped like an ellipsoid at any angle. For a *General* search type, open the *First Axis*, *Second Axis*, or *Third Axis* sections by clicking the ⊞ next to the section name. Each section has an *X*, *Y*, and *Z* value, indicating the direction of each search ellipse axis. To change the value, highlight the existing value and type the desired value. The larger the values, the further the gridding will search for data in that direction.



The General search type lets you specify an elliptical search with axes at any angle.

Anisotropy

The *Anisotropy* property of the *Inverse distance* gridding method lets you specify that data points in one direction from the lattice node have a higher weight than points in another direction. Points closer to the grid node are usually given more weight than points farther from the grid node, regardless of direction. If the points in one direction have more similarity than points in another direction, it is advantageous to give points in a specific direction more weight in determining the value of a lattice node. The relative weighting is defined by the anisotropy lengths of the three axes of the search ellipse or ellipsoid.

The available options for the *Anisotropy* setting are *Isotropic*, *Anisotropic*, and *General*. Isotropic sets all directions to equal weighting. Anisotropic allows an *Ellipse* to be specified to set the direction of weighting. The ellipse is along the X, Y, and Z axis directions. General allows each axis of the 3D ellipse to be manually altered. To change the *Anisotropy* setting, click on the existing option and select the desired option.

Why Should I Specify Anistropy

Natural phenomena are created by physical processes that often have preferred orientations. For example, course material settles out fastest at the mouth of a river while finer materials take longer to settle. Thus, the closer one is to the shoreline, the coarser are the sediments; the farther one is from the shoreline, the finer are the sediments. When interpolating a point, an observation 100 meters away but in a direction parallel to the shoreline is more likely to be similar to the value at the interpolation point than is an equidistant observation in a direction perpendicular to the shoreline. Anisotropy takes these trends in the data into account during the gridding process.

Anisotropy is useful when data sets use fundamentally different units in the X, Y, and Z directions. For example, consider plotting a flood profile along a river. The X coordinates are locations measured in miles along the river channel; the Y coordinates are measured in days; the Z coordinates are river depth; and the data values are water temperature, salinity, and concentration of pollutant. The X, Y, and Z coordinates are different units and most likely span different ranges, and one unit of X is not equal to one unit of Y or Z. Apply anisotropy to compensate for these differences.

Another example of anisotropy is creating an isotherm map of temperatures in the U.S. Although X and Y coordinates are in the same units, the Z or height units are different. In addition, temperatures tend to be most similar at constant Y and Z values, with more rapid variations as elevation and distance north of the equator change. When interpolating a lattice node, observations that lie in an east-west direction are given greater weight than observations lying an equivalent distance in the north-south direction or along the Z axis.

Using Anisotropy

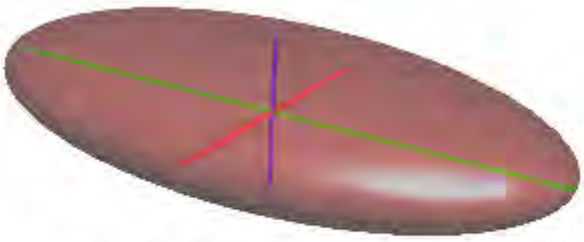
The *Anisotropy* property has three possible settings: *Isotropic*, *Anisotropic*, and *General*.

The *Isotropic* setting specifies that all directions are treated the same when weighting the distance between the lattice node and data point.



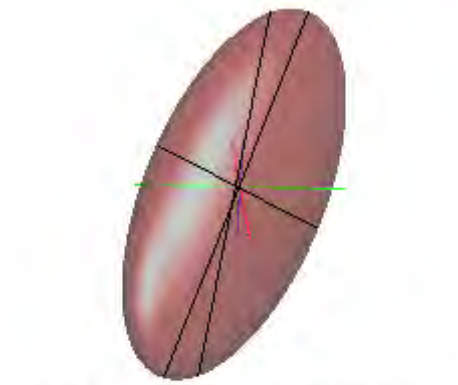
The Isotropic anisotropy setting gives all directions the same weight.

The *Anisotropic* setting lets you specify the lengths of the ellipsoid with orthogonal directions parallel to the X, Y, and Z axes. Points in the direction of the longest axis are given the most weight; points in the direction of the shortest axis are given the least weight. **Voxler** uses the anisotropy term *Ellipse* to refer to the three-dimensional ellipsoid. For example, if you specify the *X length* equal to 2, the *Y length* equal to 4, and the *Z length* equal to 1, the Y direction is given the most weight and the Z direction is given the least weight. Points that are 4 units away in the Y direction have the same weight as points that are 1 unit away in the Z direction.



The Anisotropic anisotropy setting lets you specify three orthogonal axis directions for weighting the difference between a lattice node and data points.

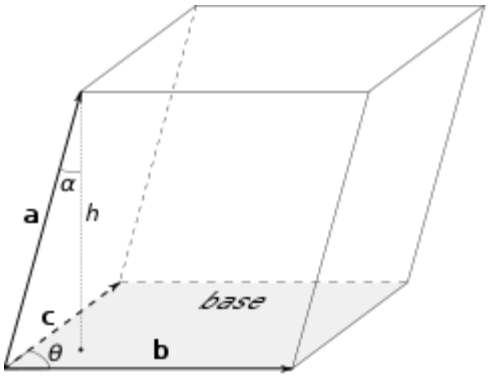
The *General* setting lets you specify three ellipsoid axes that are non-orthogonal. Points in the direction of the longest axis are given the most weight; points in the direction of the shortest direction are given the least weight.



The General anisotropy setting lets you specify three non-orthogonal axes.

Setting Ellipsoid Axes

When using the *General* option, each axis in the ellipsoid is specified separately with a different vector. Sometimes axes of the ellipsoid will result in coplanar warnings. To understand how **Voxler** is calculating whether axes are coplanar, consider this diagram:



Vectors a, b, and c shown in 3D space.

Think of the vectors a, b, and c as being the positive rays of the three axes. These three axes are coplanar if the volume of the parallelepiped is zero. We know that the axes are not coplanar if the parallelepiped has non-zero volume. Sometimes, even when the rays create a volume that is

greater than zero, round-off errors in the floating point numbers create a volume of zero. This can happen because the vectors a, b, and c that are specified are normalized into a unit cube where the longest of the lengths is 1.

Gridding Methods

Voxler has several different gridding methods available for the *Gridder* module. The differences between gridding methods are in the mathematical algorithms used to compute the weights during lattice node interpolation. Each method can result in a different representation of the data. You may want to test each method with a known data set to determine the method that provides you with the most satisfying interpretation of the data.

Gridding methods include:

- Inverse Distance to a Power
- Local Polynomial
- Data Metric

Choosing a Gridding Method

Grid method parameters control the interpolation procedures. When you create a lattice with the *Gridder* module, accept the default method and parameters for acceptable output in most cases. Different methods provide different interpretations of the data because each method calculates lattice node values using a different algorithm. If you are not satisfied with the graphic output from the lattice, consider different methods and parameters and compare the results.

General Gridding Recommendations

The *Data metrics* method is used to calculate statistics about a data set.

The *Local polynomial* method is most applicable to data sets that are locally smooth, i.e., relatively smooth surfaces within the search neighborhoods. The computational speed of the method is not significantly affected by the size of the data set.

The default *Inverse distance* method is fast but has the tendency to generate concentric spheres around high and low values unless you increase the *Smooth* value. This method does not extrapolate beyond the Z range of the data.

Inverse Distance to a Power Gridding Method

The *Inverse distance* method is a weighted average interpolator and can be either an exact or a smoothing interpolation.

Data are weighted during interpolation such that the influence of a point declines with distance from the lattice node. Weighting is assigned to data using a weighting *Power* that controls how the weighting factors drop off as distance from a lattice node increases. The greater the power, the less effective points far from the lattice node have during interpolation. As the power increases, the lattice node value approaches the value of the nearest point. For a smaller power, the weights are more evenly distributed among the neighboring data points.

Inverse distance normally behaves as an exact interpolator. When a grid node is calculated, the weights assigned to the data points are fractions, and the sum of all the weights is equal to 1.0. When a particular observation coincides with a lattice node, the distance between that observation and the node is 0.0, and that observation is given a weight of 1.0 while all other observations are

given weights of 0.0. Thus, the grid node is assigned the value of the coincidental observation. The *Smooth* parameter is a mechanism for buffering this behavior. When you assign a non-zero *Smooth* value, no point is given an overwhelming weight so that no point is given a weighting factor equal to 1.0.

One of the characteristics of inverse distance is the generation of concentric spheres surrounding the position of observations within the lattice. To reduce the effect, assign a *Smooth* value other than 0.

Inverse Distance to a Power Equations

The following equations are used for Inverse Distance to a Power:

$$\hat{C}_j = \frac{\sum_{i=1}^{n} \frac{C_i}{h_{ij}^{\beta}}}{\sum_{i=1}^{n} \frac{1}{h_{ij}^{\beta}}}$$

$$h_{ij} = \sqrt{d_{ij}^2 + \delta^2}$$

where:

h_{ij} is the effective separation distance between grid node " j" and the neighboring point "i";

$\hat{C}_j$ is the interpolated value for grid node " j";

C_i are the neighboring points;

d_{ij} is the distance between grid node " j" and the neighboring point "i";

β is the *Power* or weighting parameter; and

δ is the *Smooth* parameter.

Inverse Distance to a Power References

Davis, John C. (1986) *Statistics and Data Analysis in Geology.* John Wiley and Sons, New York, NY.

Franke, R. (1982) *Scattered Data Interpolation: Test of Some Methods*, Mathematics of Computations, v. 33, n. 157, p. 181-200.

Local Polynomial Gridding Method

The *Local polynomial* method assigns values to lattice nodes by using a weighted least squares fit with data within the search ellipsoid.

For each lattice node, the neighboring data are identified by the user-specified *Search type* and *Count*. Using only data that match the search criteria, a local polynomial is fit using weighted least squares; the lattice node value is set equal to this value. Local polynomials can be order 1, 2, or 3.

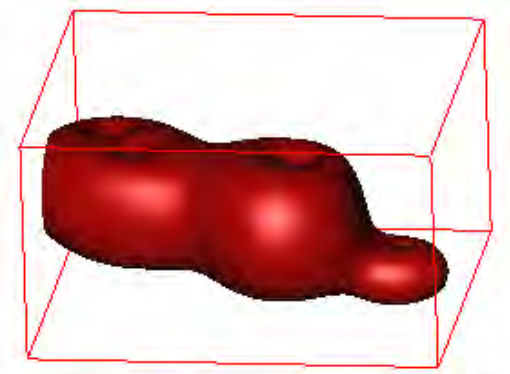
Data Metric Gridding Method

The *Data Metric* method is used to calculate statistical values using the data points found within the search. Define the search with the *Search Type* parameters. These search parameters are applied to each grid node to determine the local data set.

Metric	Description
Minimum	Minimum Z value within the search
Lower quartile	Lower quartile Z within the search
Median	Median Z value within the search
Upper quartile	Upper quartile Z within the search
Maximum	Maximum Z value within the search
Range	Range of Z values within the search
Midrange	Midrange Z value within the search
Inter-quartile range	Inter-quartile range Z value within the search
Mean	Mean Z value within the search
Standard deviation	Z standard deviation value within the search
Variance	Z variance value within the search
Coefficient of variance	Z coefficient of variation
Sum	Sum of the Z within the search
Median absolute deviation	Median absolute deviation
Root mean square	Root mean square
Count	Number of samples within the search
Density	Sample density within the search
Nearest	Distance to the nearest sample within the search
Farthest	Distance to the farthest sample within the search
Median distance	Median distance to the samples within the search
Average distance	Average distance samples within the search

Comparison of Gridding Methods

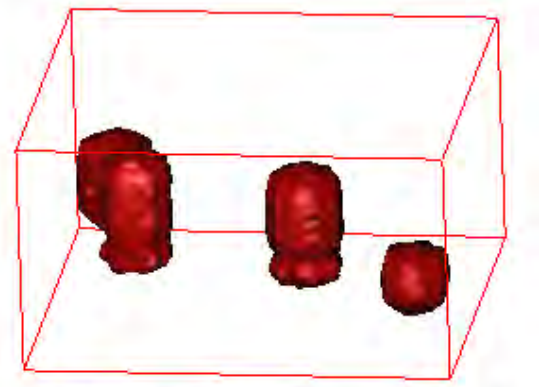
The following comparison example is using the same data set, gridding with each of the three *Method* options.



The Inverse Distance *method used with xyc1.dat shows an isosurface with a value of 20.*



This example illustrates the Local Polynomial *method with a* Simple *search. This method is not a good choice for gridding this data set since the data are not smooth.*

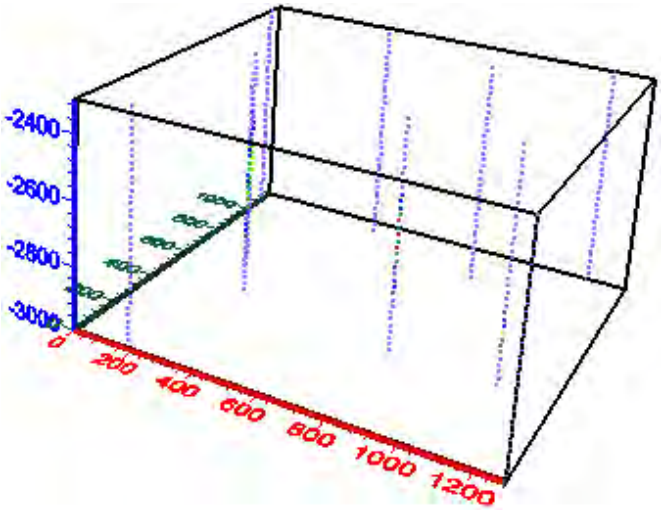


The Data Metrics *method shows an isosurface for which the* Mean *value within 1000 feet is 20 or more.*

Gridding Example

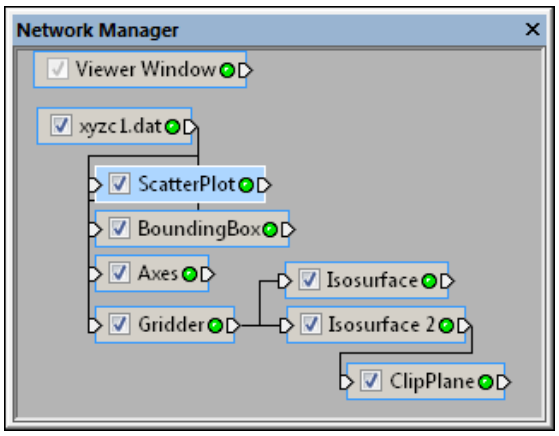
In this gridding example, consider the problem of producing isosurfaces of gold concentration given well data collected from boreholes over a region. The well locations are not regularly spaced over the area of interest. Use the **File | Import** command to load the data into the **Network Manager**, connect a *Gridder* module to the data, select the *Gridder* module in the **Network Manager** to display the properties in the **Property Manager** and click the *Begin Gridding* button to produce a uniform regularly-spaced lattice over the XYZ extents. Add an *Isosurface* module to the *Gridder* to display surfaces of equal concentration based on the lattice node values. The steps are detailed below.

1. Click the **File | Import** command.
2. In the **Import** dialog, select the xyzc1.dat data file, located in the **Voxler** Samples directory, and click *Open*.
3. If the **Data Import Options** dialog is displayed, select *Delimited* as the *Field Format*. Check the *Comma* box in the *Delimiters* section. Check the *Double Quote* box in the *Text Qualifiers* section and click *OK*.
4. In the **Property Manager** specify the data columns for the X, Y, and Z coordinates and C component.
    a. Set the Output type to Points
    b. Set X coordinates to Column A: X
    c. Set Y coordinates to Column B: Y
    d. Set Z coordinates to Column C: Z
    e. Set Component columns to 1
    f. Set Component-1 to Column D: C
5. To view the data that was imported, click on the *xyzc1.dat* module in the **Network Manager**. In the **Property Manager**, click the *Edit Worksheet* button. The linked data is opened in a worksheet document. The document title includes the module name and number. Any changes made to the data while viewing the linked worksheet are automatically saved and applied in the project.
6. Close the worksheet window, or click on the project file tab, to return to the viewer window.
7. In the **Network Manager**, click on the *xyzc1.dat* module. Click the **Network | Graphics Output | ScatterPlot** to add a scatter plot to the view. You can change any properties of the *ScatterPlot*, such as the symbol size and color.
8. In the **Network Manager**, click on the *xyzc1.dat* module. Click the **Network | Graphics Output | BoundingBox** to add a bounding box to the view.
9. In the **Network Manager**, click on the *xyzc1.dat* module. Click the **Network | Graphics Output | Axes** to add axes to the view.

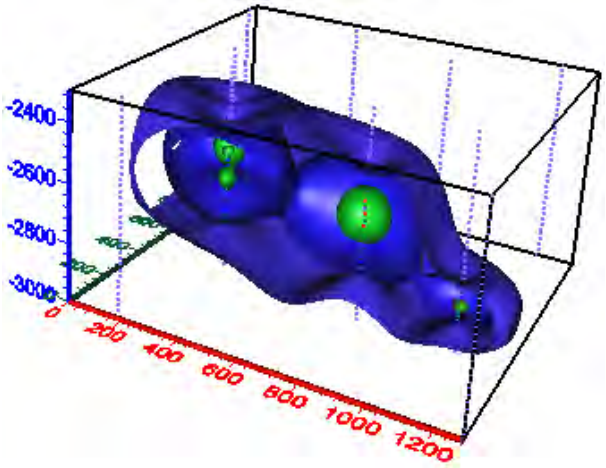


A ScatterPlot of the data set shows the distribution of the XYZ data points from the borehole sampling. BoundingBox and Axes modules have also been added.

10. In the **Network Manager**, click on the *xyzc1.dat* module. Click the **Network | Computational | Gridder** command to attach a *Gridder* module.
11. Click on the *Gridder* module. In the **Property Manager**, click the *Begin Gridding* button to start the gridding process using the default options.
12. In the **Network Manager**, click on the *Gridder* module. Click the **Network | Graphics Output | Isosurface** command to display the gridded data as an *Isosurface* module. Change any properties of the *Isosurface*, such as the *Isovalue* or color.



The full network is shown. A ClipPlane module has been added to the isosurfaces to show the interior surfaces.



Isosurfaces have been added to the Gridder module to display surfaces of equal concentration based on the lattice node values.

Can I create a Gridder lattice using the Nearest Neighbor method?

The *Gridder* module *Method* options include *Data Metric, Inverse Distance*, and *Local Polynomial.* Adjust the *Gridder* properties as follows to simulate a *Nearest Neighbor* gridding method.

Method: Inverse Distance

Power: 0

Using a power of 0 causes all distances to be treated the same, since n^0 = 1 where n is any number.

Search Type: Simple

Min Count: 1

Max Count: 1

Setting the *Search Max Count* to 1 causes the *Gridder* to use only the closest data point.

How can I create a lattice from an XYZC data file and save the lattice data?

Use a *Gridder module* to create a regularly spaced lattice from an irregularly spaced XYZC data file. The lattice information can be saved directly from the *Gridder* module using the **File | Save Data** command.

Example

1. Choose the **File | Import** command and load your XYZC data file (ie. .DAT or .XLS) into **Voxler**.
2. Right-click the data module in the **Network Manager** and select **Computational | Gridder** to attach a *Gridder* module to the data module.
3. Select the *Gridder* module in the **Network Manager, c**hange any of the properties of the *Gridder* module in the **Property Manager**, and click the *Begin Gridding* button.
4. Once the gridding process is complete, your data has been converted into a 3D lattice. You can right-click the *Gridder* module and select **Graphics Output** to visualize the data, or you can right-click the *Gridder* module and select **Save Data** to save the lattice data to a .DAT or .XLS file.

How do I create a surface from my XYZ data file?

To create a surface from an XYZ data file, follow these steps:

1. Create a column of 0's in the data file.
2. Open **Voxler** and choose the **File | Import** command.
3. Select the data file and click the *Open* button.
4. In the **Data Import Options** dialog, set the appropriate settings for your data and click the *OK* button.
5. In the **Property Manager**, specify the *X Coordinate* and *Y Coordinate* columns, select the column of 0's as the *Z Coordinate*, and select the Z data column as the *Single Component*. Click the *OK* button.
6. Right-click the data module in the **Network** window and select **Computational | Gridder** to create a *Gridder* module.
7. Select the *Gridder* module in the **Network Manager**. In the **Properties** window, set any settings you wish and click the *Begin Gridding* button.
8. Once the gridding is complete, right-click the *Gridder* module and select **Graphics Output | HeightField**.

Alternatively, if you own **Surfer** you can grid the XYZ data in **Surfer** (using the **Grid | Data** command), load the Surfer grid file into **Voxler** (using the **File | Import** command), and attach a *HeightField* module directly to the grid file.

Math

The **Network | Computational | Math** command adds a *Math* module.

The *Math* module creates a new output lattice from one or more input lattices by applying a numeric expression to one or more input lattices. The output lattice is calculated one node at a time by applying the numeric expression to the input lattice nodes.

It is NOT necessary for the input lattices to be have identical X, Y, or Z ranges. This module is currently limited to a maximum of 3 input lattices.

When evaluating expressions that may contain 'no data' (either NULL, or blank) values as part of a grid file, the *Math* module can specify which numeric value is to be treated as 'no data'. See the *Math Module Examples* page for sample math expressions.

Inputs

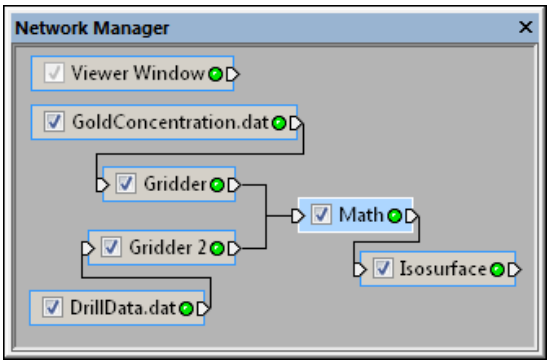
Lattice is the input type for the *Math* module.

Outputs

The *Math* module creates a uniform lattice. It may be connected to the *Graphics Output Modules* or the *Computational Modules*. An *Info Module* may also be connected to the output node.

Properties

The *Math* module properties are described below.



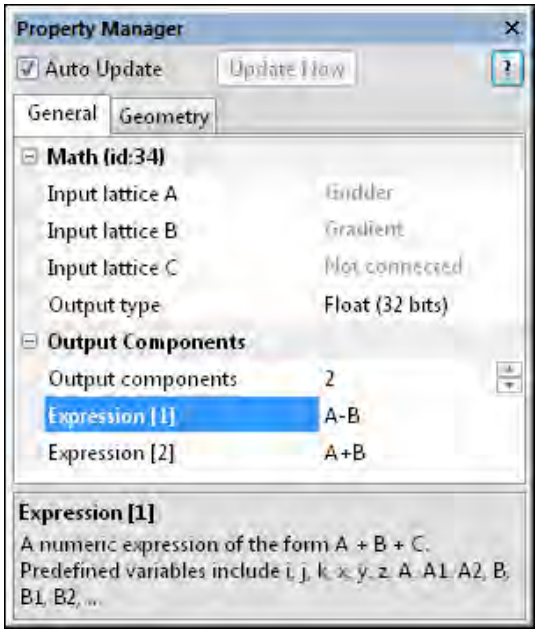
*Select the Math module in the **Network Manager** to display its properties in the **Property Manager**.*

The *Math* module contains the following tabs in the **Property Manager:**

General

Geometry

General Options



Enter a numeric expression to define the generated lattice values.

Input Lattice Names

Next to *Input lattice A*, *Input lattice B*, and *Input lattice C*, are the module names of the lattices attached to the math module. These can be changed by connecting a different lattice module to the *Math* module.

Output Type

The *Output type* is the format that is used to save the output lattice. To change the type, click on the existing option and select the desired output lattice type from the list.

Output Components

Click the ⊞ next to *Output Components* to open the section and display the output component options. The *Output components* is the number of components to create in the output lattice. To change the number of components, highlight the existing value and type the desired value or click the  to increase or decrease the number of components.

Expression

The *Expression[1]* is the mathematical equation that will create the first component in the output lattice. The math module can have between one and 100 expressions which create that number of components in the output lattice. To change the equation, highlight the existing text and type the equation, in the form A + B + C. A, B, and C refer to the input lattice components. When used without a number after the letter, the first component is used. When used in the form A1+B2+C4, the letter refers to the input lattice and the number is the component where 1 is the first component of the input lattice, 2 is the second, and so on.

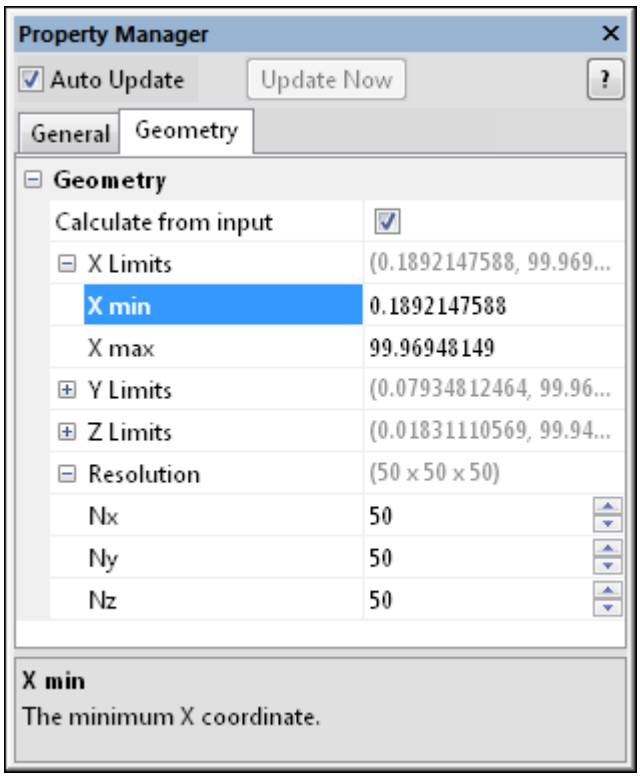
To use the X, Y, or Z values in the equation, type the letter X, Y, or Z in the expression. The X, Y, and Z value of the node being calculated is used in the equation for that node.

The index value of the current node can also be used in the equation. To use the index value, type the letter i, j, or k in the equation for the X, Y, or Z position. The i is the X node index, j is the Y node index and k is the Z node index. The first node in each direction has the value 1. The next node in each direction has the value 2 and so on.

If the function cannot be evaluated, the corresponding lattice node is blanked.

Geometry Page

The *FunctionLattice* and *TestLattice* modules **Geometry** page displays the extents and resolution of the lattice. To open the **Geometry** page, click on the *FunctionLattice* or *TestLattice* module in the **Network Manager**. In the **Property Manager**, click on the **Geometry** tab.



This is an example of the information displayed in the Property Manager on the Geometry tab for a Math module.

Automatically Calculate Limits

Check the box next to *Calculate From input* to have **Voxler** automatically calculate the extents of the output lattice based on the combined extents of the input lattices.

X Limits

The *X Limits* item displays the limits in the X direction for the function lattice. Enter the *X min* and *X max* coordinates. These ranges are the limits of the lattice that is created. The values are in map coordinates. To change the values, highlight the existing value and type the desired minimum or maximum value.

Y Limits

The *Y Limits* item displays the limits in the Y direction for the function lattice. Enter the *Y min* and *Y max* coordinates. These ranges are the limits of the lattice that is created. The values are in map coordinates. To change the values, highlight the existing value and type the desired minimum or maximum value.

Z Limits

The *Z Limits* item displays the limits in the Z direction for the function lattice. Enter the *Z min* and *Z max* coordinates. These ranges are the limits of the lattice that is created. The values are in map coordinates. To change the values, highlight the existing value and type the desired minimum or maximum value.

Resolution

Click the ⊞ next to *Resolution* to set the number of lattice nodes in each of the X, Y, and Z directions. The default number of nodes is 50 in each direction. To change the *Nx*, *Ny*, or *Nz* value, highlight the existing value and type the desired value or click the  button to increase or decrease the number of nodes. An increase in the number of lattice nodes provides more resolution, but it also increases the amount of time needed to create the lattice and to draw output graphics. It also increases the amount of memory needed to display graphics on-screen and to save the graphics to a .VOXB network file.

Math Module Examples

The math module can be used in multiple ways. Below are three ways to use the *Math* module.

Example 1: Blanking

The math module can be used to truncate a 3D grid with a 2D grid. Using grid A and grid B, the expression IF (Z < B, 0, A) tells **Voxler** if Z is less than the value in grid B, set the data value to 0, else use the value in grid A.

The math module can be used to blank a vertical region of a grid file. Create a GRD file in **Surfer** (or another program) with the Z value set to the X coordinate. Load the GRD file in **Voxler** and use a *Math* module to set the data values to 0 if the X values in the **Voxler** *Gridder* module output are less than the data values in the GRD file. Use an IF function with the syntax IF Z > B, 0, A OR Z > B? 0:A. With a GRD file, the elevation data is imported as component 1. Both syntax statements say to set the output value to 0 if the Z value is greater than the elevation data in the GRD file ("B"). If the Z value is less than the elevation in value in B, use the data value in A. The value of 0 is used in this example, but any value less than the Z minimum of the data can be used. Connect a *VolRender* module to the output of the *Math* module, and set the 0 value to be transparent in the **Colormap Editor** to complete the blanking procedure.

The math module can be used to blank a horizontal region to the left or right of an X value in a BLN file. Remove the BLN header, save to a DAT file, then grid the DAT file using the X column for the Z value. Attach a Gridder module and GRD file, and change the reference in the Math equation from Z to X, making the syntax X > B? 0:A.

Example 2: Math Functions on Multiple Lattices

The math module can be used to perform math functions, such as adding, subtracting, multiplying, or dividing, values from multiple lattices. For instance, if you have loaded two data files and gridded both, you can use an *Math* module with both *Gridders* as the input lattices. Then, set the expression to A+B to add the component values from the two grids.

Example 3: Combining Multiple Input from a Single Lattice

Sometimes, it is desirable to compute values based on multiple components. For instance, your drill data may have porosity, permeability, gamma, resistivity, and other components. To determine where to place the next drill hole, you may want to combine multiple components and map the

composite. To do this, attach the lattice to a *Math* module. You can then use an equation such as (A1+A2)*A3 to add two components and multiple that value by a third component. This new lattice can then be used to create a volrender showing the composite information.

Merge

The **Network | Computational | Merge** command adds a *Merge* module.

The *Merge* module combines two or more input lattices into a single uniform output lattice. The range and resolution of the output lattice may be specified. All input lattices should be in the same coordinate system. The maximum number of input lattices is currently limited to 5.

Inputs

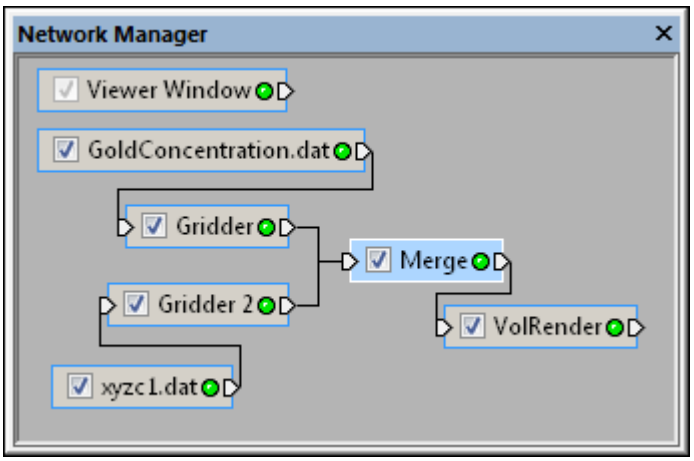
Lattice is the input type for the *Merge* module.

Outputs

The *Merge* module creates a uniform lattice. It may be connected to the *Graphics Output Modules* or the *Computational Modules*. An *Info Module* may also be connected to the output node.

Properties

The *Merge* module properties are described below.



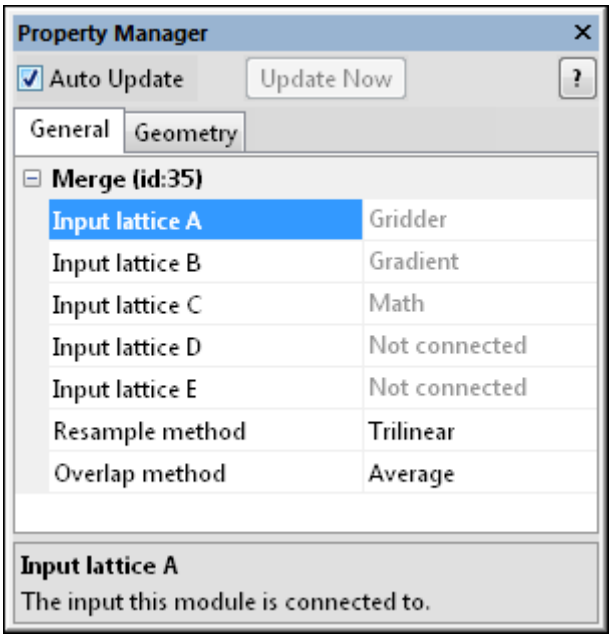
*Select the Merge module in the **Network Manager** to display its properties in the **Property Manager**.*

The *Merge* module contains the following tabs in the **Property Manager:**

General

Geometry

General Options



Choose the method used to merge the multiple values from two or more lattices.

Input Lattice

The *Input lattice A*, *Input lattice B*, *Input lattice C*, *Input lattice D*, and *Input lattice E* properties show the source to which the module is connected. This option cannot be changed in the **Property Manager**, but can be changed in the **Network Manager** by changing the module input.

Resample Method

The *Resample method* sets the method used to calculate the output values from the input components. To change the method, click on the existing method and select the desired method from the list. Available options are *Nearest neighbor* and *Trilinear*.

Nearest neighbor applies the closest grid node value on the original grid to the grid node value in the output grid. This is the fastest resampling method, though it can result in distorted output if the original grid and output grid differ in size.

Trilinear interpolation uses a weighted average of eight input nodes to interpolate a new value. Some lattice node values in the original grid may be applied to more than one output lattice node; conversely, some of the input lattice nodes may not be used at all.

Overlap Method

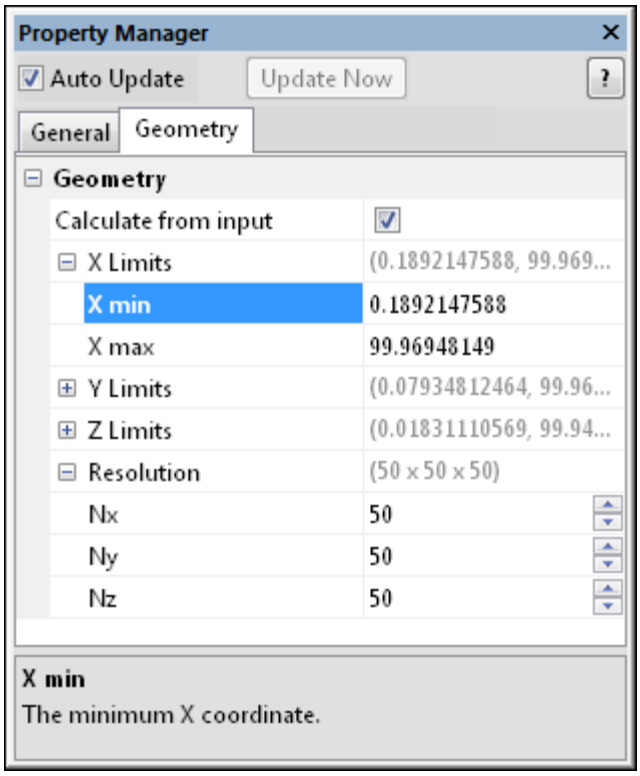
The *Overlap method* specifies the method used to combine multiple values from overlapping input lattices. To change the method, click on the existing method and select the desired method from the list. A node is interpolated from each input grid and then combined using one of the following available methods: *Average*, *Count*, *First*, *Last*, *Maximum*, *Median*, or *Minimum*.

- *Average* method averages the grid nodes and uses the average value.
- *Count* determines the number of data points used in determining the grid node value.
- *First* uses the first grid node value.
- *Last* uses the last grid node value.

- *Maximum* uses the maximum grid node value.
- *Median* uses the (unaveraged) middle grid node value.
- *Minimum* uses the minimum grid node value.

Merge Module - Geometry Page

The *Merge* and *Resample* modules **Geometry** page displays the extents and resolution of the output lattice. To open the **Geometry** page, click on the *Merge* or *Resample* modules in the **Network Manager**. In the **Property Manager**, click on the **Geometry** tab.



This is an example of the information displayed in the Property Manager on the Geometry tab for a Merge module.

Automatically Set Limits

Check the box next to *Calculate from input* to have **Voxler** automatically calculate the extents of the output lattice based on the combined extents of the input lattices.

X Limits

The *X Limits* item displays the limits in the X direction for the function lattice. Enter the *X min* and *X max* coordinates. These ranges are the limits of the lattice that is created. The values are in map coordinates. To change the values, highlight the existing value and type the desired minimum or maximum value.

Y Limits

The *Y Limits* item displays the limits in the Y direction for the function lattice. Enter the *Y min* and *Y max* coordinates. These ranges are the limits of the lattice that is created. The values are in map

coordinates. To change the values, highlight the existing value and type the desired minimum or maximum value.

Z Limits

The *Z Limits* item displays the limits in the Z direction for the function lattice. Enter the *Z min* and *Z max* coordinates. These ranges are the limits of the lattice that is created. The values are in map coordinates. To change the values, highlight the existing value and type the desired minimum or maximum value.

Resolution

Resolution is the number of nodes created in the lattice. By default, this value is 50 in each direction. Enter the number of nodes (*Nx*, *Ny*, and *Nz*) on each axis in the lattice. An increase in the number of lattice nodes provides more resolution, but also increases the amount of time needed to create the lattice and to draw output graphics. It also increases the amount of memory needed to display graphics on-screen and to save the graphic to a .VOXB file. To change the resolution, highlight the existing value and type a new value or click the  button to increase or decrease the values. Values can be between 1 and 10000.

Resample

The **Network | Computational | Resample** command adds a *Resample* module.

The *Resample* module allows the resolution of a lattice to be changed. This is performed by computing new data values at each output lattice node by interpolating the data values from the input lattice. The *Resample* module does NOT perform extrapolation. Any node outside the limits of the input lattice is blanked.

Inputs

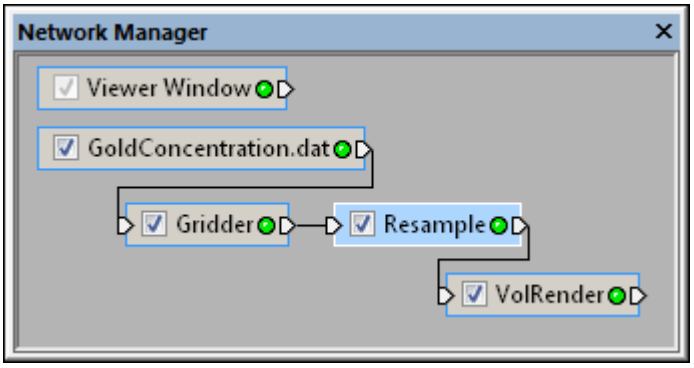
Lattice is the input type for the *Resample* module.

Outputs

The *Resample* module creates a uniform lattice. It may be connected to the *Graphics Output Modules* or the *Computational Modules*. An *Info Module* may also be connected to the output node.

Properties

The *Resample* module properties are described below.



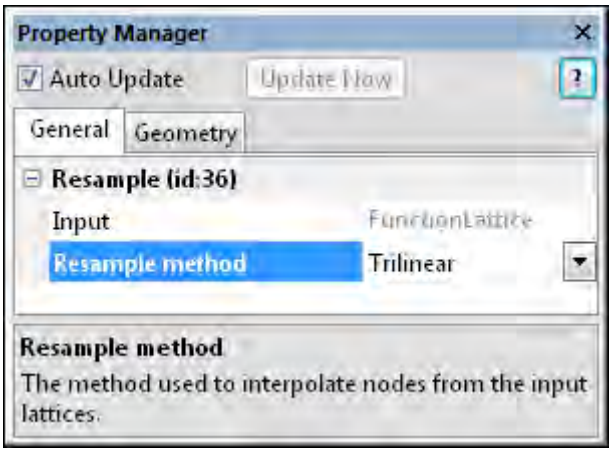
*Select the Resample module in the **Network Manager** to display its properties in the **Property Manager**.*

The *Resample* module contains the following tabs in the **Property Manager:**

General

Geometry

General Options



Use the Resample method to interpolate the input lattice data values in order to change the output lattice.

Input

The *Input* property shows the source to which the module is connected. This option cannot be changed in the **Property Manager**, but can be changed in the **Network Manager** by changing the module input.

Resample Method

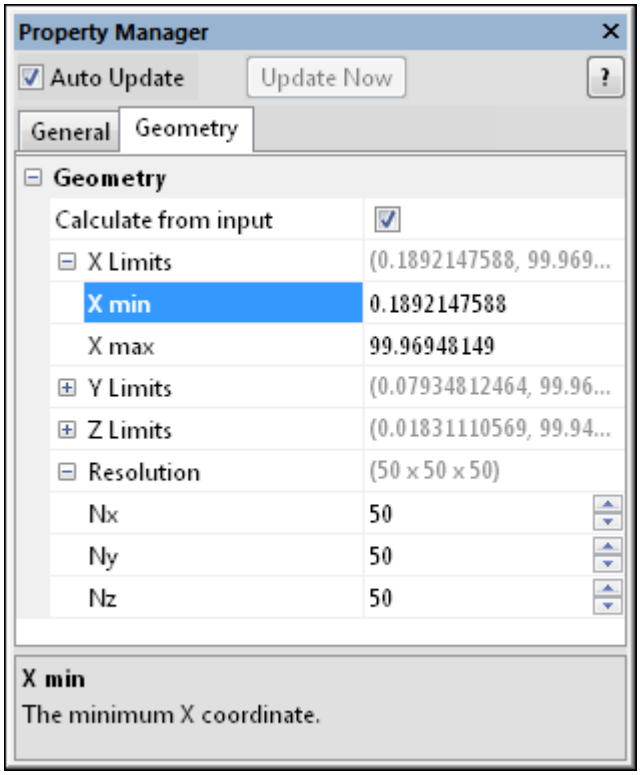
The *Resample method* sets the method used to calculate the output values from the input components. To change the method, click on the existing method and select the desired method from the list. Available options are *Nearest neighbor* and *Trilinear*.

Nearest neighbor applies the closest grid node value on the original grid to the grid node value in the output grid. This is the fastest resampling method, though it can result in distorted output if the original grid and output grid differ in size.

Trilinear interpolation uses a weighted average of eight input nodes to interpolate a new value. Some lattice node values in the original grid may be applied to more than one output lattice node; conversely, some of the input lattice nodes may not be used at all.

Resample Module - Geometry Page

The *Merge* and *Resample* modules **Geometry** page displays the extents and resolution of the output lattice. To open the **Geometry** page, click on the *Merge* or *Resample* modules in the **Network Manager**. In the **Property Manager**, click on the **Geometry** tab.



This is an example of the information displayed in the Property Manager on the Geometry tab for a Merge module.

Automatically Set Limits

Check the box next to *Calculate from input* to have **Voxler** automatically calculate the extents of the output lattice based on the combined extents of the input lattices.

X Limits

The *X Limits* item displays the limits in the X direction for the function lattice. Enter the *X min* and *X max* coordinates. These ranges are the limits of the lattice that is created. The values are in map coordinates. To change the values, highlight the existing value and type the desired minimum or maximum value.

Y Limits

The *Y Limits* item displays the limits in the Y direction for the function lattice. Enter the *Y min* and *Y max* coordinates. These ranges are the limits of the lattice that is created. The values are in map

coordinates. To change the values, highlight the existing value and type the desired minimum or maximum value.

Z Limits

The *Z Limits* item displays the limits in the Z direction for the function lattice. Enter the *Z min* and *Z max* coordinates. These ranges are the limits of the lattice that is created. The values are in map coordinates. To change the values, highlight the existing value and type the desired minimum or maximum value.

Resolution

Resolution is the number of nodes created in the lattice. By default, this value is 50 in each direction. Enter the number of nodes (*Nx*, *Ny*, and *Nz*) on each axis in the lattice. An increase in the number of lattice nodes provides more resolution, but also increases the amount of time needed to create the lattice and to draw output graphics. It also increases the amount of memory needed to display graphics on-screen and to save the graphic to a .VOXB file. To change the resolution, highlight the existing value and type a new value or click the  button to increase or decrease the values. Values can be between 1 and 10000.

Slice

The **Network | Computational | Slice** command adds a *Slice* module.

The *Slice* module creates a two-dimensional slice through a three-dimensional input lattice. The plane orientation may be preset to one of the local axis planes or in an arbitrary direction.

Inputs

Lattice is the input type for the *Slice* module.

Outputs

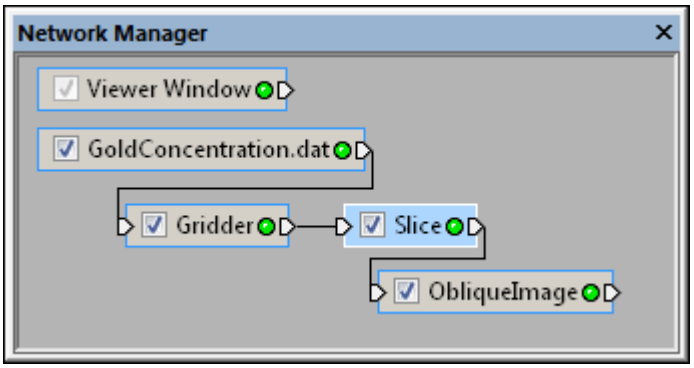
The *Slice* module creates a uniform 2D lattice. It may be connected to the *Graphics Output Modules* or the *Computational Modules*. An *Info Module* may also be connected to the output node.

Output Components

Two types of output modules— lattice and geometry— may be connected to the *Slice* module. To see this in action, click the output connection pad of the *Slice* module. A context menu appears with two choices: **Connect Output Lattice** and **Connect Output Geometry**. Choose the former to connect a module that contains transformed lattice or point data, e.g., *BoundingBox*. Choose the latter to connect a module that contains output geometry, e.g., *ClipPlane*. The output geometry consists of the dragger graphics and optional border.

Properties

The *Slice* module properties are described below.



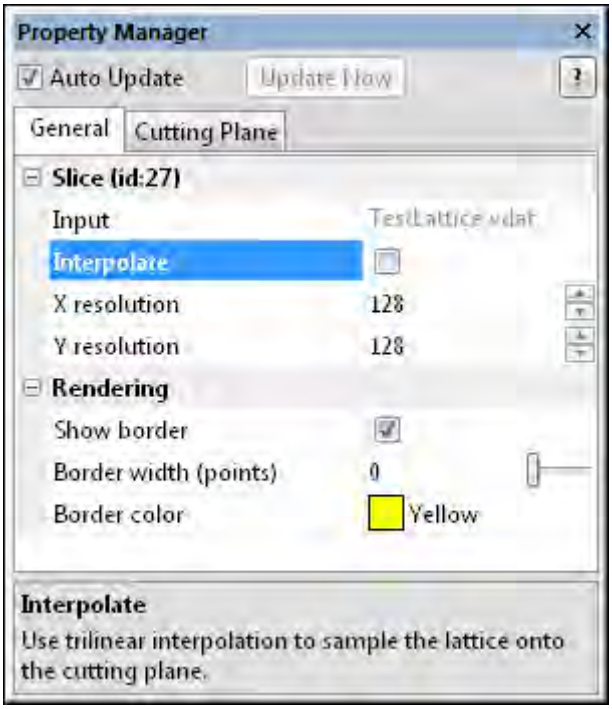
*Select the Slice module in the **Network Manager** to display its properties in the **Property Manager**.*

The *Info* module contains the following tabs in the **Property Manager:**

General

Cutting Plane

General Options



Uncheck the box next to Interpolate to use nearest neighbor interpolation to create a two-dimensional slice of a three-dimensional input lattice. Check the Interpolate box to use trilinear interpolation.

Input

The *Input* property shows the source to which the module is connected. This option cannot be changed in the **Property Manager**, but can be changed in the **Network Manager** by changing the module input.

Interpolate Output Lattice

Check the box next to the *Interpolate* command to use trilinear interpolation to resample the lattice onto the cutting plane. Uncheck the box next to the *Interpolate* command to use nearest neighbor interpolation, which applies the closest grid node value on the original grid to the grid node value in the slice.

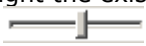
Output Lattice Resolution

The *X resolution* and *Y resolution* defines the number of nodes in the X and Y direction in the output grid. To change the resolution, highlight the existing value and type the desired value or click the  to increase or decrease the value. Type a value next to *X resolution* and *Y resolution* to specify the number of nodes in each direction of the output lattice. Resolution specifies the sampling frequency within the cutting plane. The maximum resolution is 10000 nodes in each direction. Increasing the resolution makes the resulting lattice smoother, but increases the computation time to create the lattice.

Show Border

Click the ⊞ next to *Rendering* to set the border properties for the slice. Check the box next to *Show border* to display the border on your map.

Border Width

To increase or decrease the thickness of the border, highlight the existing value next to *Border width (points)* and type a new value or click and drag the  to increase or decrease the width of the border line. The line width is displayed in points and ranges from zero to 4 points.

Border Color

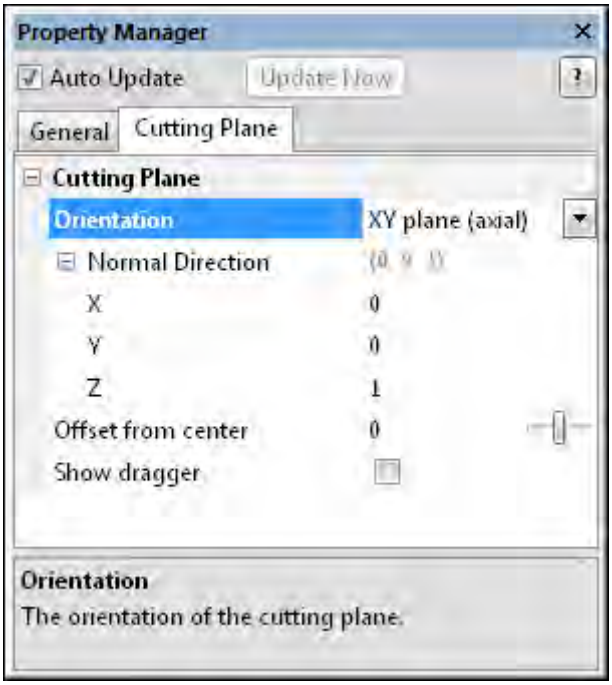
The *Border color* section specifies the color for the border line. To change the color, click the colored box and choose a new color from the color palette. If the basic colors in the palette do not meet your needs, click *Other* to create a custom color.

Cutting Plane

The *Cutting Plane,* a property of some modules, determines the orientation ("normal") of the cutting plane. You can choose one of the preset local axis plane normal, or you can choose an arbitrary direction by entering values for X, Y, and Z. The plane normal may be specified numerically with the *Normal Direction* property or graphically when the dragger is shown.

Properties

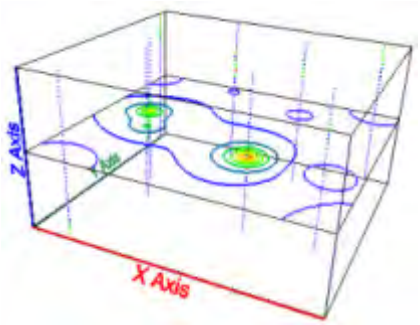
The *Cutting Plane* properties are described below. This feature may be accessed via the *Contours*, *ObliqueImage*, and *Slice* modules.



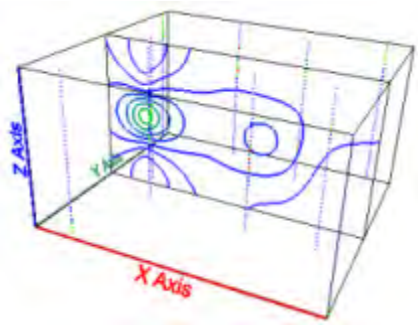
The Cutting Plane *properties may be changed to affect the orientation ("normal") of the cutting plane.*

Orientation

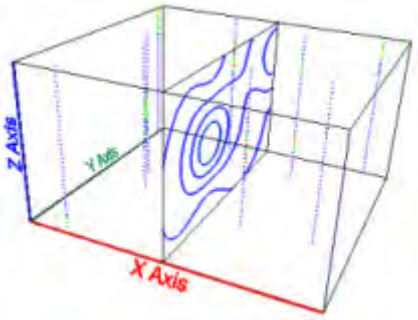
To change the *Orientation*, click on the existing option and select the desired option. Choose one of the preset local axis plane normals— *XY plane (axial)*, *XZ plane (coronal)*, or *YZ plane (sagittal)*— or choose *Custom* to enter custom *Normal Direction* values. *Axial* indicates the XY plane that travels horizontally. *Coronal* indicates the XZ plane that travels vertically. *Sagittal* indicates the YZ plane that travels vertically. The *Custom* setting lets you type a value under *Normal Direction* for each axis.



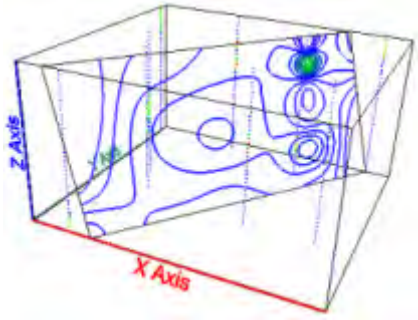
The contours Orientation is set to XY plane (axial).



The contours Orientation is set to XZ plane (coronal).



The contours Orientation is set to YZ plane (sagittal).



The contours Orientation is set to Custom.

Normal Direction

Enter a value under *Normal Direction* for the *X*, *Y*, and *Z* components of the plane normal. Changing these values changes the cutting plane orientation with respect to the normal for that axis. Changing the *X* value when the *Y* and *Z* values are zero creates a YZ cutting plane. Setting each of the *X*, *Y*, and *Z* values to 1 produces a plane with an oblique orientation. This vector is the normal to the cutting plane.

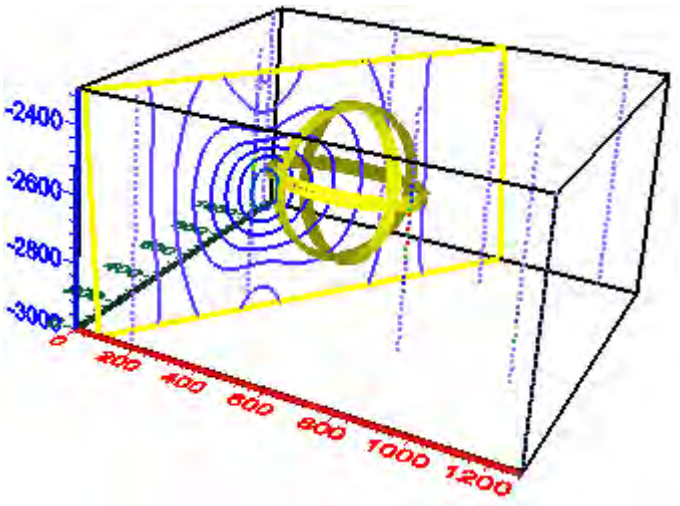
Offset from Center

The *Offset from center* property measures the distance of the cutting plane from the center of the lattice. To change the offset, highlight the existing value and type a new value or click and drag the  to increase or decrease the distance from the center. Units are in axis units. If the X axis goes from 0 to 60, a value of 0 for the *Offset from center* will place the plane at X= 30. A value of just less than 30 will place the plane at the maximum X value. A value of just greater than -30 will place the plane at the minimum X value.

Show Dragger

The *Dragger* allows interactive positioning and rotation of the plane.

Check the *Show dragger* box to show the dragger— a virtual, rotatable trackball— and allow interactive positioning and rotation of the plane. The dragger allows the orientation and offset of the cutting plane to be specified.



The Dragger is shown above in yellow and can be used to interactively rotate the cutting plane.

Drag one of the three bands to rotate around a principal axis in the direction of the ring. Drag anywhere on the ball (between the rings) to perform an unconstrained rotation in any direction.

To specify a user-defined rotation axis, press the SHIFT key while clicking the left mouse button and dragging. A new distinctively-colored axis is added.

To scale the size of the trackball, press the CTRL key and drag the trackball.

To offset the plane in the perpendicular direction, drag the cutting plane itself.

Subset

The **Network | Computational | Subset** command adds a *Subset* module.

The *Subset* module extracts a particular region of interest from an input lattice for further analysis. Change the available options that specify the geometric range of the subset, the sampling frequency, and the desired data components of the subset. This module is commonly used to reduce a large data set or to extract a particular region of interest for further analysis. This module is useful if you need to extract a particular region of interest for further analysis.

Inputs

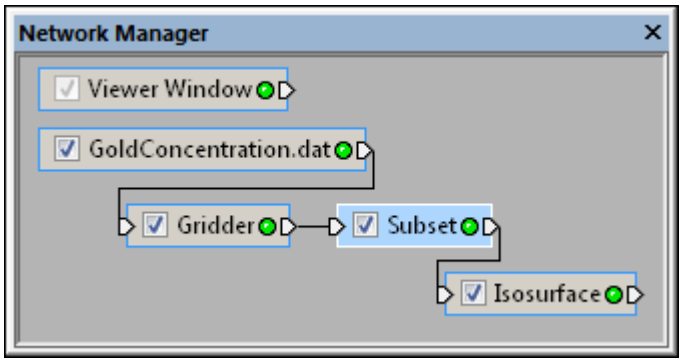
Lattice is the input type for the *Subset* module.

Outputs

The *Subset* module creates a lattice that is the same type as the input lattice. It may be connected to the *Graphics Output Modules* or the *Computational Modules*. An *Info Module* may also be connected to the output node.

Properties

The *Subset* module properties are described below.



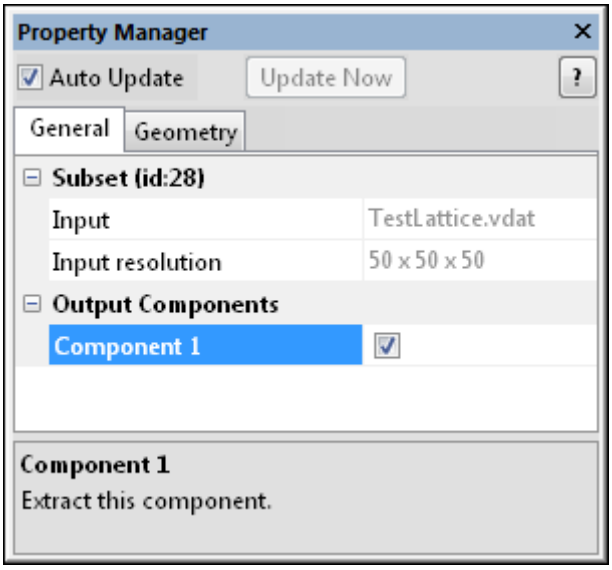
*Select the Subset module in the **Network Manager** to display its properties in the **Property Manager**.*

The *Subset* module contains the following tabs in the **Property Manager:**

General

Geometry

General Options



Use the Subset module to extract a region of interest from a lattice for further analysis.

Input

The *Input* property shows the source to which the module is connected. This option cannot be changed in the **Property Manager**, but can be changed in the **Network Manager** by changing the module input.

Input Resolution

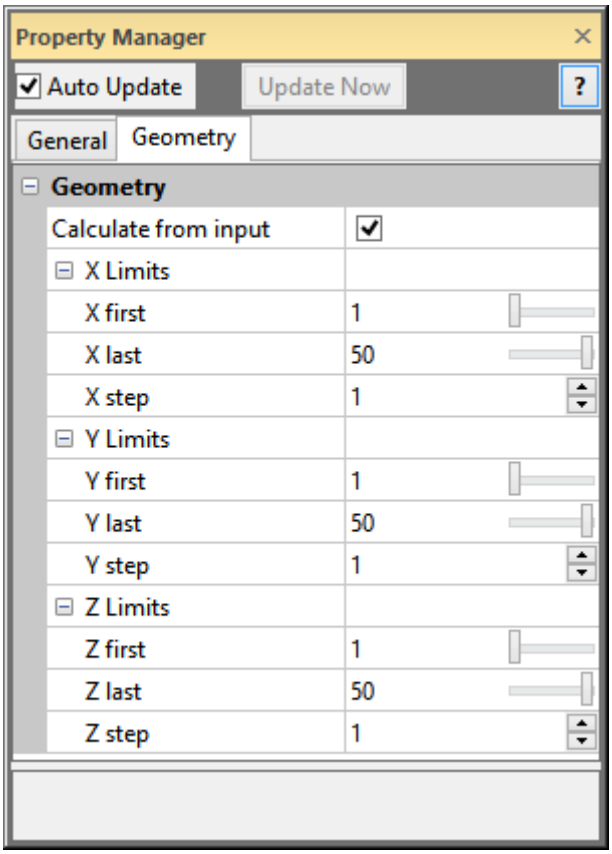
The *Input resolution* displays the number of grid nodes in the input lattice in X x Y x Z.

Components

Click the ⊞ next to *Output Components* to open the list of components. All components from the input grid are listed. Check the box next to a numbered component name to extract the chosen component(s).

Subset Module - Geometry Page

The *Subset* module **Geometry** page displays the extents and resolution of the output lattice. To open the **Geometry** page, click on the *Subset* module in the **Network Manager**. In the **Property Manager**, click on the **Geometry** tab.



This is an example of the information displayed in the Property Manager on the Geometry tab for a Subset module.

Automatically Set Limits

Check the box next to *Calculate from input* to have **Voxler** automatically calculate the extents of the output lattice based on the combined extents of the input lattices.

X Limits

The *X Limits* item displays the limits in the X direction for the lattice. Enter the *X first* and *X last* values. The *X first* value is the first grid node to be extracted. The *X last* value is the last grid node to be extracted. Values for *X first* and *X last* are index values for the X direction. The first node in the direction in the input grid is 1. The next node is 2, and so on. To change the value, highlight

the existing value and type the desired value or click and drag the  to increase or decrease the index grid node value.

The *X step* value indicates how many grid nodes from the original lattice to skip. If *X step* is set to 1, every grid node in the range from *X first* to *X last* is used in the output lattice. If *X step* is set to 2, every other grid node is extracted. If *X step* is set to 3, every third grid node is extracted, and so on. To change the *X step* value, highlight the existing value and type a new value or click the  button to increase or decrease the step value.

Y Limits

The *Y Limits* item displays the limits in the Y direction for the lattice. Enter the *Y first* and *Y last* values. The *Y first* value is the first grid node to be extracted. The *Y last* value is the last grid node to be extracted. Values for *Y first* and *Y last* are index values for the Y direction. The first node in the direction in the input grid is 1. The next node is 2, and so on. To change the value, highlight the existing value and type the desired value or click and drag the  to increase or decrease the index grid node value.

The *Y step* value indicates how many grid nodes from the original lattice to skip. If *Y step* is set to 1, every grid node in the range from *Y first* to *Y last* is used in the output lattice. If *Y step* is set to 2, every other grid node is extracted. If *Y step* is set to 3, every third grid node is extracted, and so on. To change the *Y step* value, highlight the existing value and type a new value or click the  button to increase or decrease the step value.

Z Limits

The *Z Limits* item displays the limits in the Z direction for the lattice. Enter the *Z first* and *Z last* values. The *Z first* value is the first grid node to be extracted. The *Z last* value is the last grid node to be extracted. Values for *Z first* and *Z last* are index values for the Z direction. The first node in the direction in the input grid is 1. The next node is 2, and so on. To change the value, highlight the existing value and type the desired value or click and drag the  to increase or decrease the index grid node value.

The *Z step* value indicates how many grid nodes from the original lattice to skip. If *Z step* is set to 1, every grid node in the range from *Z first* to *Z last* is used in the output lattice. If *Z step* is set to 2, every other grid node is extracted. If *Z step* is set to 3, every third grid node is extracted, and so on. To change the *Z step* value, highlight the existing value and type a new value or click the  button to increase or decrease the step value.

Transform

The **Network | Computational | Transform** command adds a *Transform* module.

The *Transform* module transforms the X, Y, and Z coordinates of an input data set or lattice using a standard 4x4 transformation matrix. The order of transformations is *Scale*, *Rotation*, *Translation*. Rotation and scaling are performed around the object's *Origin*.

Inputs

Lattice, point data, and well data are input types for the *Transform* module.

Outputs

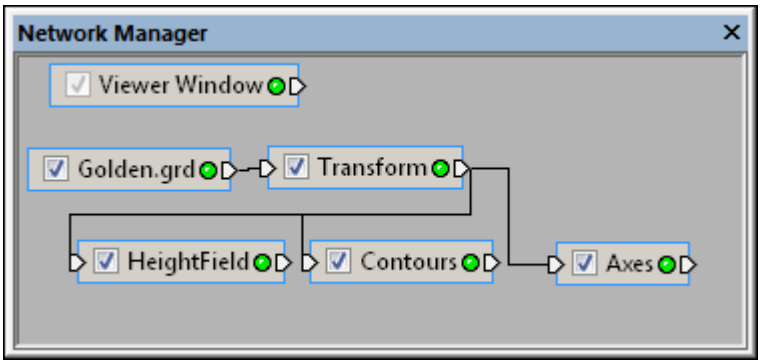
The Transform module creates lattice, point data, and well data depending on the input type. It may be connected to the *Graphics Output Modules* or the *Computational Modules*. An *Info Module* may also be connected to the output node.

Output Components

Two types of output modules— data and geometry— may be connected to the *Transform* module. To see this in action, click the output connection pad of the Transform module. A context menu appears with two choices: *Connect Output Data* and *Connect Output Geometry*. Choose *Connect Output Data* to connect a module that contains transformed lattice or point data, e.g., *Isosurface* or *ScatterPlot*. Choose *Connect Output Geometry* to connect a module that contains output geometry, e.g., ClipPlane. The output geometry is the dragger graphics. If the dragger is turned off, there is no output geometry.

Properties

The *Transform* module properties are described below.



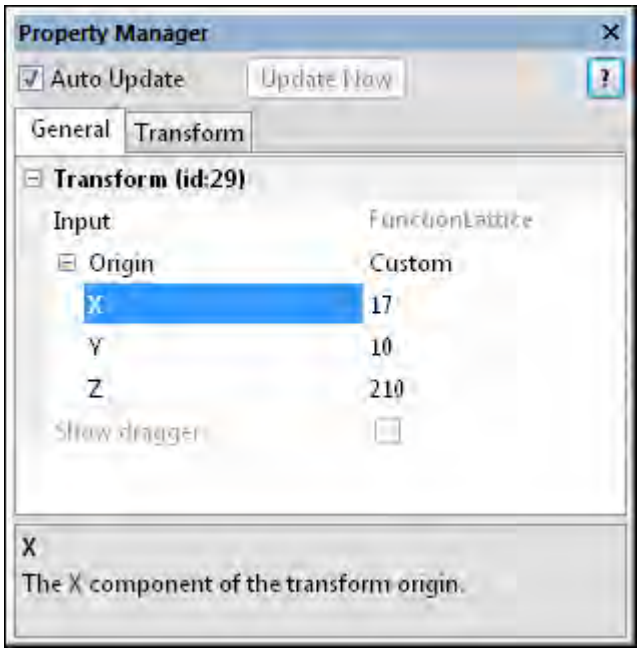
*Select the Transform module in the **Network Manager** to display its properties in the **Property Manager**.*

The *Transform* module contains the following tabs in the **Property Manager:**

General

Transform

General Options



*Change the Transform module properties in the **Property Manager**.*

Input

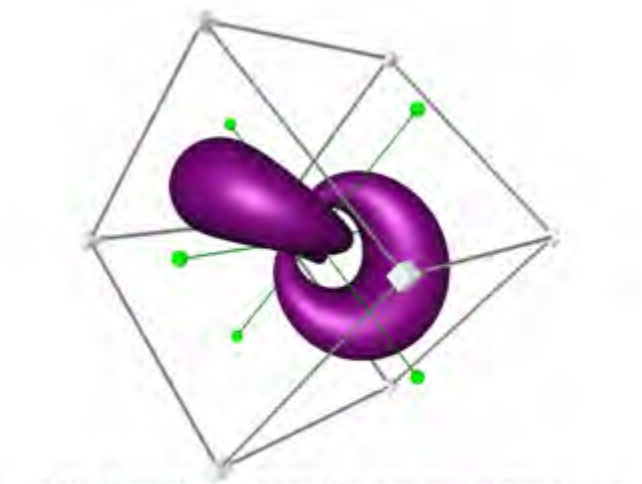
The *Input* property shows the source to which the module is connected. This option cannot be changed in the **Property Manager**, but can be changed in the **Network Manager** by changing the module input.

Origin

The *Origin* is the location about which the *Transform Scale* and *Rotation* are performed. To open the *Origin* section, click the ⊞ next to *Origin*. Available options are *Center*, *Lower left*, *Upper right*, and *Custom*. To change the *Origin*, click on the existing value and select the desired option from the list.

When the *Origin* is set to *Center*, all scaling and rotations are performed about the X, Y, and Z midpoints of the original data. This was the method used by **Voxler** 1 and 2. When the *Origin* is set to *Lower left*, scaling and rotations are performed about the X, Y, and Z minimum values. When the *Origin* is set to *Upper right*, scaling and rotations are performed about the X, Y, and Z maximum values. When the *Origin* is set to *Custom*, scaling and rotations are performed about the specified *X*, *Y*, and *Z* values. The *X*, *Y*, and *Z* value options are only editable when the *Origin* is set to *Custom*. To change the *X*, *Y*, and *Z* values, highlight the existing value and type the desired value. Values are in map coordinates.

Show Dragger



Use the Transform dragger (in gray and green above) to interactively change the position of the transform.

Check the box next to the *Show dragger* option to allow interactive adjustment of the transform. This option is only available when the *Origin* is set to *Center*. The dragger supports adjusting the scale, rotation, and translation.

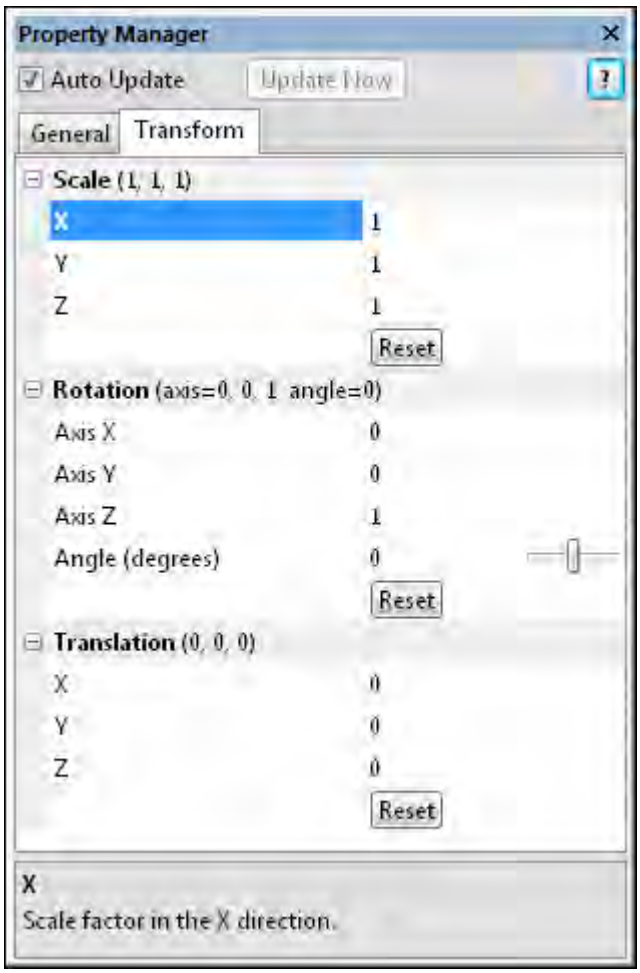
To use the dragger to change the scale, check the box next to *Show dragger*. In the **Viewer** window, drag the corner cubes of the dragger to change the scale. Uniform scaling is performed on the *Transform*, changing the X, Y, and Z values by the same amount. To change the *Scale* values by different amounts, press and hold the SHIFT key before selecting a corner cube. The dragger can then be dragged in various directions to change the scaling values. Hold down the CTRL key before selecting a corner cube to scale uniformly toward a corner point.

Click and drag any of the six green circle end markers of the dragger axis cross to change the rotation manually. The initial drag direction determines the orientation in which the rotation is performed. Hold down the SHIFT key before selecting the marker to rotate in a free form fashion around the sphere.

Click and drag any of the gray sides of the dragger to change the translation manually. Hold down the SHIFT key before selecting any side to lock translation to a single axis in the plane. Hold down the CTRL key before selecting a side to translate along the plane's normal vector. Translation is performed in the plane of the selected side.

Transform Module - Transform Page

The *Transform* module **Transform** page displays the *Scale*, *Rotation*, and *Translation* for the transform being applied. To open the **Transform** page, click on the *Transform* module in the **Network Manager**. In the **Property Manager**, click on the **Transform** tab.



*The **Transform** page contains the Scale, Rotation, and Translation options for a Transform module.*

Scale

The *Scale* transforms the input coordinates in the selected X, Y, or Z direction about the *Origin*. To open the *Scale* section, click the ⊞ next to *Scale*. To change the *Scale* values, highlight the existing value in the *X*, *Y*, and *Z* options and type a new value. Click the *Reset* button to reset the scale factor back to its original value.

Scaling is always done around the *Origin*. Changing the *Origin* changes the values that are calculated and displayed by output geometry. For example, consider a data set that has a *Transform* module with a *Z* scale of 0.1 applied. The table below lists the original data and the scaled new data values with various origins:

Origin	Z Min	Z Max	Z Range	Z Midpoint
Original Data	90	100	10	95
Center (Default)	94.5	95.5	1	95
Lower Left	90	91	1	90.5
Upper Right	99	100	1	99.5
Custom (0,0,0)	9	10	1	9.5
Custom (10,10,10)	18	19	1	18.5

This table shows how the original data changes with the same Z Scale transform applied from different Origin values.

The scale transform is calculated by subtracting the *Origin* base value from the *Original Data* value, multiplying by the scale value, and adding this to the *Original Data* value. The general form of the scale equation is: $Value_{New} = (Value_{Original} - Value_{Origin}) * Value_{Scale} + Value_{Origin}$

As an example, with the data above, the Z Min value for the Custom (10, 10, 10) value is calculated like:

$ZMin_{Custom} = (ZMin_{Original} - Z_{Origin}) * Z_{Scale} + Z_{Origin}$

$ZMin_{Custom} = (90 - 10) * 0.1 + 10$

$ZMin_{Custom} = (80) * 0.1 + 10$

$ZMin_{Custom} = 8 + 10 = 18$

Rotation

The *Rotation* transforms the input coordinates by the *Angle (degrees)* about the *Origin* in the selected axis direction. To open the *Rotation* section, click the ⊞ next to *Rotation.* To change the *Rotation*, highlight the existing value in the *Axis X, Axis Y,* or *Axis Z* direction and type a new value. The rotation direction is specified by the right hand rule: take your right hand and point your thumb in the direction of the positive rotation axis. The direction your fingers are curled is the direction of positive rotation angles. To change the *Angle (degrees)* value, highlight the existing *Angle (degrees)* value and type a new value. *Angle (degrees)* values are in degrees. Click the *Reset* button to reset the rotation back to its original setting.

Rotations are applied about the *Origin*. All points rotate about the origin value. The further away a point is from the *Origin*, the further away the point will be from its original location. Changing the *Origin* does change the location of where the rotation is applied and all of the locations of rotated points. The *Origin* does not have any rotation applied.

For example, consider a data set that has the following characteristics: *Lower left* value is 10, 10, 90. The *Upper right* is 50, 80, 100. The *Center* is 30, 45, 95. A point located at 50, 80, 95 is rotated with a *Transform* module with a *Rotation* where *Axis Z* is 1 and *Angle (degrees)* is 30 is applied. The table below shows the X, Y, Z value of the rotated point with each origin.

Origin	X	Y	Z
Original Data	50	80	95
Center (30, 45, 95)	29.821	85.311	95
Lower Left (10, 10, 90)	9.641	90.622	95
Upper Right (50, 80, 100)	50	80	95
Custom (0,0,0)	3.301	94.282	95
Custom (10,10,10)	9.641	90.622	95

This table shows how the original data changes with the same Z Rotation transform applied from different Origin values.

The rotation transform is calculated by subtracting the *Origin* base value from the *Original Data* value, multiplying by the rotation value, and adding this to the *Original Data* value. The general form of the scale equation is: $Value_{New} = (Value_{Original} - Value_{Origin}) * Value_{rotation} + Value_{Origin}$. The $Value_{rotation}$ is determined with a matrix that involves the Cosine, Sine, and X, Y, and Z positions of the *Origin* point and the *Original Data* value.

Translation

The *Translation* moves the input coordinates the specified amount in each of the X, Y, and Z. To open the *Translation* section, click the ⊞ next to *Translation.* To change the location of input data, highlight the existing *X*, *Y*, or *Z* value and type a new value that corresponds with the amount to translation in the X, Y, and Z directions. Click the *Reset* button to reset the translation back to the original values.

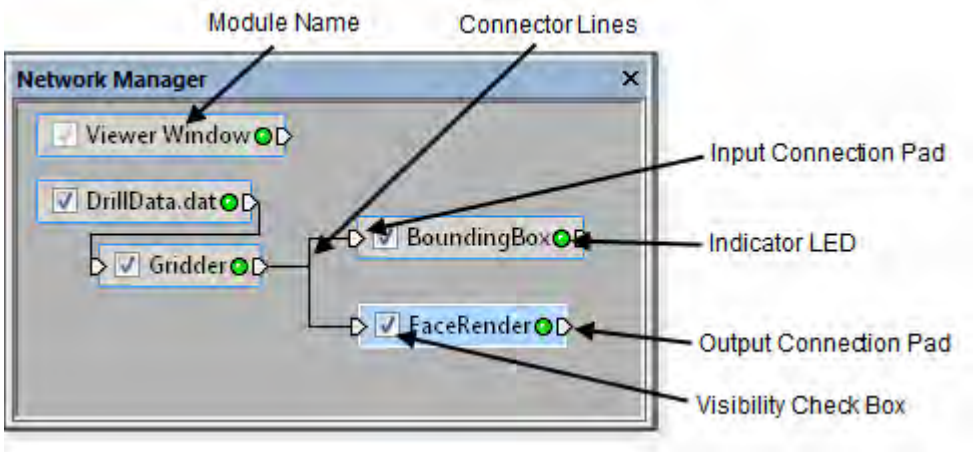
The *Translation* does not change when the *Origin* changes. As an example, if your data is at X Min = 40 and X Max = 70 and an *X Translation* of 10 is input, the output data will have X Min = 50 and X Max = 80. All data are shifted by the same 10 units, regardless of whether the *Origin* is *Center*, *Lower left*, *Upper right*, or *Custom*. The *Origin* has no affect on the translation calculation.

Chapter 6 - Axes, Bounding Boxes, and Clipping Planes

Introduction to Modules

A module is a data set or a process to be applied to a data set or process. Modules are the building blocks from which the final **Voxler** output is constructed. Modules accept data on their input connection pads ▷, modify the data, and pass it along through the output connection pad ▷.

Modules are displayed in the **Network Manager**. You can connect and disconnect modules to create a visualization network representing the flow of data. Modules need to be connected in order to generate an output. The current geometry output of the modules in the **Network Manager** are displayed in a three-dimensional view in the **Viewer** window. If the data is not connected to a graphics output module, nothing is displayed in the **Viewer** window.



*The **Network Manager** displays the visualization network, which includes all loaded modules and their connections.*

View All Modules

All modules are accessed using the Network menu commands. Alternatively, all modules are accessed in the **Module Manager** when the *Show all modules* button is selected .

View Applicable Modules

When a module is selected in the **Network Manager** and the *Show all modules* button  is not selected in the **Module Manager**, available modules that can be connected to the selected module output port are displayed in the **Module Manager**. Alternatively, right-click a module in the **Network Manager** to display only the applicable modules in the context menu.

On the module description pages, there are **Inputs** and **Outputs** sections to discuss the type of input and output modules that each module are compatible.

Import

Click **File | Import** to open the **Import** dialog. The type of data determines what type of module can be attached to it. **Voxler** supports several different data types. See the *File Format Chart* for a detailed list of supported file formats.

Viewer Window

The **Viewer Window** is a unique module that is automatically created when a new instance of **Voxler** is generated. The *Viewer Window* module appears in the **Network Manager**. The *Viewer Window* module cannot be deleted. The purpose of the *Viewer Window* module is to control the properties of the **Viewer** window for the current instance of **Voxler**. To change the **Viewer** window properties for future instances of **Voxler**, choose the **Tools | Options** command and adjust properties on the **Colors** page of the **Options** dialog.

Module Types

There are four types of modules: *computational*, *data source*, *general*, and *graphics output.* Each module type is discussed below.

Computational Modules

Computational modules alter the data by changing the data type, filtering, creating a gradient, gridding, performing mathematical transformations, merging, resampling, slicing, creating a subset, or transforming coordinates. Click on one of the following computational modules for detailed information on using the module and module properties.

ChangeType

DuplicateFilter

ExclusionFilter

ExtractPoints

Filter

Gradient

Gridder

Math

Merge

Resample

Slice

Subset

Transform

Data Source Modules

Data source modules serve as a source of raw data. The data can be imported or created from mathematical functions. Click on one of the following data source modules for detailed information on using the module and module properties.

Import (*Data Source*, *Point Source*, *Lattice Source*, *Geometry Source*)

FunctionLattice

TestLattice

WellData (combines multiple *Data Source* modules into a single output)

General Modules

General modules display module information and provide custom lighting in the **Viewer** window. Click on one of the following general modules for detailed information on using the module and module properties.

Info

Light

Graphics Output

Graphics output modules create graphics in the **Viewer** window. Typically, these modules require data input. Click on one of the following graphics output modules for detailed information on using the module and module properties.

Annotation

Axes

BoundingBox

ClipPlane

Contours

FaceRender

HeightField

Isosurface

ObliqueImage

OrthoImage

ScatterPlot

StreamLines

Text

VectorPlot

VolRender

WellRender

Graphics Output

Choose the **Network | Graphics Output** command to add data-dependent graphics to the scene.

Graphics output modules create graphics in the **Viewer** window. Typically, these modules require data input.

The modules are added to the menu in alphabetical order to make them easier to find. If a module is selected in the **Network Manager** when this command is chosen, then only those output modules compatible with the selected module are displayed. Choose a module from the list and **Voxler** automatically adds it to the selected module.

Graphics output modules include the following:

- *Annotation* displays a two-dimensional screen-aligned block of text
- *Axes* displays a set of axes for the input modules
- *BoundingBox* draws a three-dimensional bounding box around the input object's geometry
- *ClipPlane* limits the input geometry

- *Contours* draws one or more contour lines (isolines) in a plane through a three-dimensional lattice
- *FaceRender* displays cubes along the faces of a three-dimensional lattice
- *HeightField* displaces the "height" of a lattice slice according to a single component
- *Isosurface* computes an isosurface through a three-dimensional lattice and outputs the geometry
- *ObliqueImage* displays an arbitrary slice through a lattice
- *OrthoImage* displays an orthogonal slice through a lattice
- *ScatterPlot* displays a point set or lattice as a series of symbols
- *StreamLines* displays streamlines from a multiple component lattice
- *Text* allows the addition of an anchored text frame to the scene
- *VectorPlot* displays vectors on a three-dimensional lattice or point set
- *VolRender* performs direct volume rendering of a three-dimensional lattice
- *WellRender* displays well data as lines or tubes

Axes

The **Network | Graphics Output | Axes** command adds an *Axes* module.

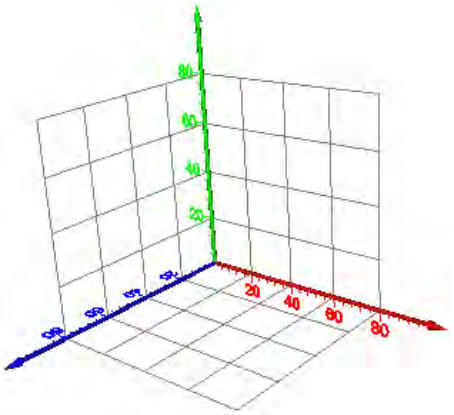
The *Axes* module creates a set of axes. The axes are attached to an input point set or lattice. The axis labels are planar by default, although the orientation of the plane can be changed in the **Property Manager**. A grid can also be displayed between any two axes. By default, the X axis is red, the Y axis is green, and the Z axis is blue. The colors of the axes can be changed in the **Property Manager** for the existing axes or in the **Tools | Options** dialog for default conditions for future axes.

Inputs

Lattice and point data are input types for the *Axes* module.

Outputs

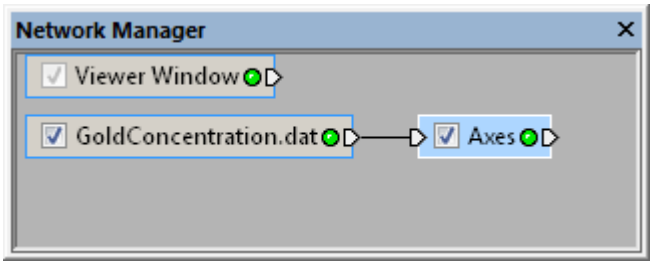
The *Axes* module creates an output geometry. It may be connected to other *Graphics Output Modules*. An *Info Module* may also be connected to the output node.



*Add a grid to your axes in the **Axes** section of the **Property Manager**.*

Properties

The *Axes* module properties are described below.



*Select the Axes module in the **Network Manager** to display its properties in the **Property Manager**.*

The *Axes* module contains the following tabs in the **Property Manager:**

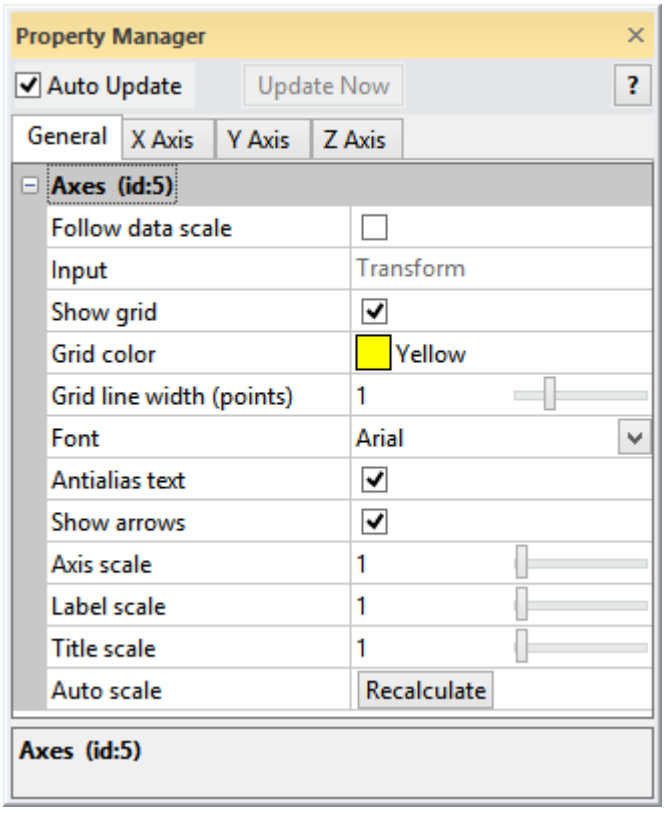
General

X Axis

Y Axis

Z Axis

General Options



Change the axes general properties to fit your needs.

Axes Section

Any changes made to these options automatically apply to the X, Y, and Z axes.

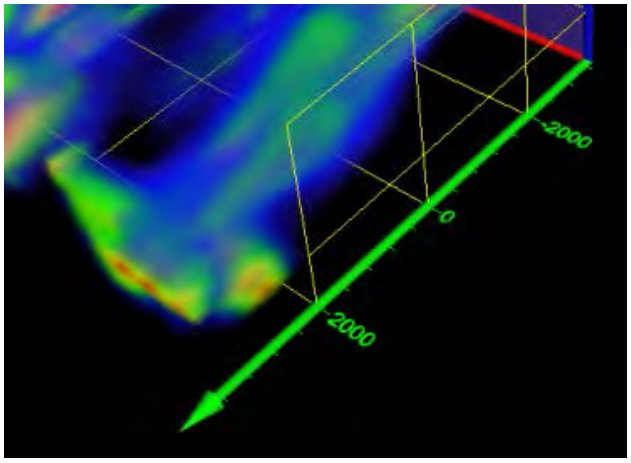
Follow Data Scale

The *Follow data scale* property determines if the axes are scaled by the original input data or the output from a *Transform* module. This property has no effect when a *Transform* module is not upstream of the *Axes* module. When the *Follow data scale* check box is checked, the original data is used to scale the axes. The axis values and labels will correspond to the transformed geometry coordinates.

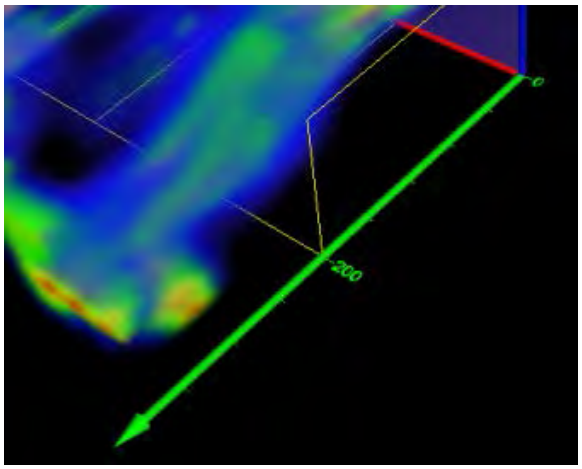
When the *Follow data scale* check box is not checked, the transformed data is used to scale the axes. The axis values and labels will correspond to the original data coordinates. Scaling the axis with the transformed data (i.e. the *Follow data scale* check box is NOT checked) keeps the axes and graphics outputs in the correct relative position when compared to a non-transformed graphics output.

It is generally necessary to adjust the axis scales, increments, and label settings after changing the *Follow data scale* state. You can manually change these settings on the **X Axis**, **Y Axis**, and **Z Axis** pages of the **Property Manager**. Alternatively, it is recommended that you click the *Recalculate* button in the *Auto scale* field after checking or unchecking the *Follow data scale* check box.

The following images highlight the difference between the *Follow data scale* property states. The original Y data range is 0 to 350, and the transformed Y data range is -3325 to 3675.



In this image the Dissolved Solids (VolRender).voxb sample file has been modified with the Follow data scale *check box checked.*



This image shows the Dissolved Solids (VolRender).voxb sample file with the Follow data scale *check box not checked.*

Input

The *Input* property shows the source to which the module is connected. This option cannot be changed in the **Property Manager**, but can be changed in the **Network Manager** by changing the module input.

Show Grid

Check the box next to the *Show grid* command to show the grid lines between the axes. Uncheck the box to turn off the display of the grid. The grid lines are drawn at the label locations.

Grid Color

Click the existing color box next to the *Grid color* command to change the color of the grid lines. The color palette opens to allow selection of a new color (or creation of a custom color). Click on the desired color or click the *Other* button to open the **Colors** dialog to set custom colors.

Grid Line Width

The *Grid line width (points)* is the thickness of the grid lines (in points). To change the line width, highlight the existing value and type a new value or drag the  to increase or decrease the line width. A line width of zero is equal to one pixel. The *Grid line width (points)* values vary between zero and 4.

Font

The *Font* command specifies the font that the axis text uses. To change the font, click the current font and choose a new font from the drop down menu.

Antialias Text

Check the box next to the *Antialias text* option to make the fonts appear smoother by slightly adjusting the colors. This usually makes the text appear somewhat more dim.

Show Arrows

Check the box next to the *Show arrows* command to show the three-dimensional arrows at the end of the axes. Uncheck the box to turn off the display of the arrows.

Axis Scale

Use the *Axis scale* option to scale the axis thickness. To change the thickness of the axis, highlight the existing value and type a new value, or drag the  to the desired value. The nominal thickness is defined as a fraction of the bounding box diagonal. When the axis thickness is set to 0, the main axis shaft is drawn as a single line. A thickness greater than 0 draws the axis as a 3D cylinder.

Label Scale

The *Label scale* command changes the label size. To change the label size, highlight the existing value and type a new value, or drag the  to the desired value.

Title Scale

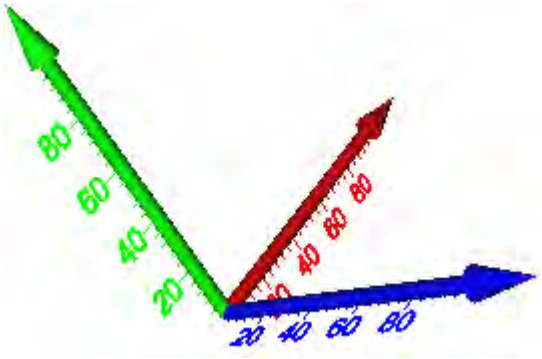
The *Title scale* command changes the title size. To change the title font size, highlight the existing value and type a new value, or drag the  to the desired value.

Auto Scale

Click the *Recalculate* button next to the *Auto scale* command to recalculate data-dependent default values for the current data set. This is useful when the input data has changed and you wish to rescale the axes to fit the data, or if manual scaling has resulted in axes that are not visible. This function replaces any user-specified axis scaling parameters.

Axes Module - X Axis Page

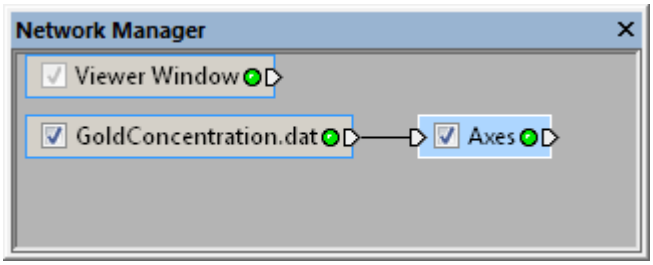
The *Axes* module **X Axis** page displays the properties of the X Axis. To open the **X Axis** page, click on the *Axes* module in the **Network Manager**. In the **Property Manager**, click on the **X Axis** tab.



*Change the axis properties on the **X Axis**, **Y Axis**, and **Z Axis** tabs of the **Property Manager**.*

Properties

The **X Axis** property sections of the *Axes* module are described below.



*Select the Axes module in the **Network Manager** to display its properties in the **Property Manager**.*

The *Axes* module contains the following tabs in the **Property Manager:**

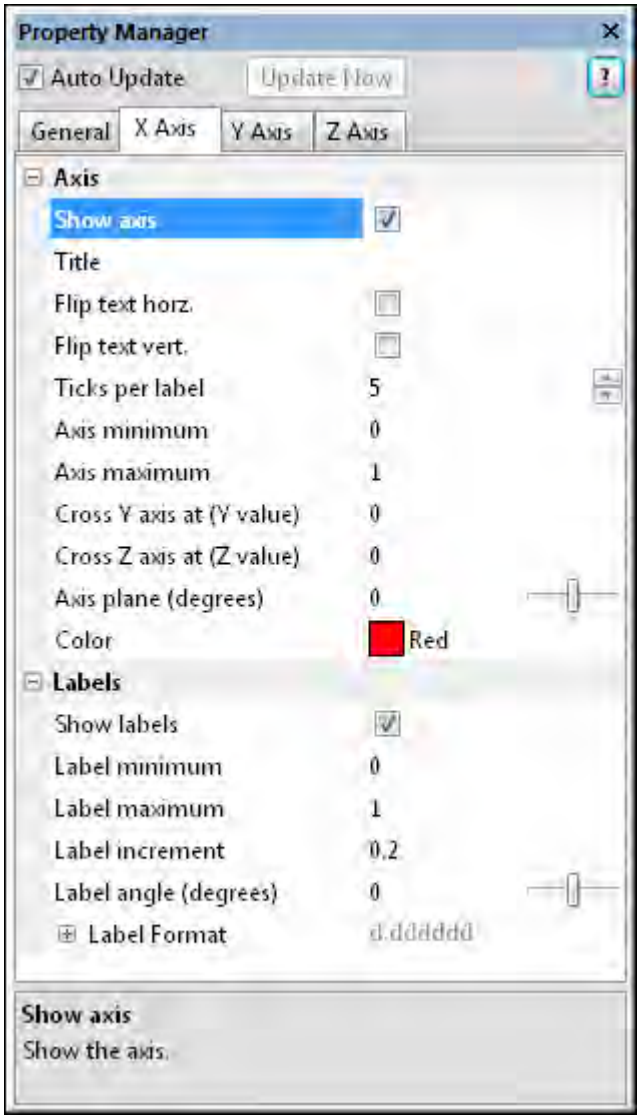
General

X Axis

Y Axis

Z Axis

General Options



*Customize the **X Axis** properties.*

Show Axis

Check the box next to the *Show axis* command to show the axis. Uncheck the box to turn the display off for the selected axis.

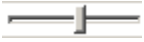
Title

Enter text in the *Title* section to give the axis a title.

Flip Text

Check the box next to the *Flip text horz.* or *Flip text vert.* commands to flip the label and title text horizontally or vertically for easier viewing after rotating the view.

Ticks per Label

The *Ticks per label* option controls how many tick mark divisions occur between each pair of labels. If the *Ticks per label* value is set to 2, one tick mark and two divisions will occur between labels. This value can be between 1 (no intermediate tick marks) and 100. To change the value, highlight the existing value and type a new number or click and drag the  to increase or decrease the number of ticks.

Axis Minimum and Maximum

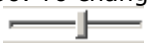
Enter a minimum data value next to *Axis minimum* for the starting axis value and a maximum data value next to *Axis maximum* for the ending axis value. Labels are drawn between the *Label minimum* and *Label maximum* values. If the *Label minimum* and *Label maximum* values are different than the *Axis minimum* and *Axis maximum* values, part of the axis will be displayed without labels.

Axes are drawn in the coordinate system of the input object and do not reflect any changes applied via the *Transform* module.

Cross Axis At

The *Cross ... axis at (... value)* value is the value at which the selected axis crosses the specified axis. Highlight the existing value and type a new value to change this. Axis values are in the coordinate system of the input object and do not reflect any changes applied via the *Transform* module.

Axis Plane

The *Axis plane (degrees)* contains the labels and tick marks, and can be specified as a rotation angle around the axis. The direction of rotation is specified by the right hand rule. Take your right hand and point your thumb in the direction of the positive rotation axis. The direction your fingers are curled is the direction of positive rotation angles. Angles are in degrees and range from -180 to +180. To change the rotation, highlight the existing value and type a new value or click and drag the  to increase or decrease the values.

Color

The *Color* section specifies the color of the axis and the axis labels and title text. To change the color, click the existing color box and select a new color from the color palette. Click on the desired color or click the *Other* button to open the **Colors** dialog to set custom colors.

Show Labels

Check the box next to the *Show labels* command to show tick marks and labels along the selected axis. Uncheck the box to hide the tick marks and labels.

Label Minimum and Maximum

Enter a minimum data value next to *Label minimum* for the first label and a maximum data value next to *Label maximum* for the last label. Labels are only drawn between the *Label minimum* and *Label maximum* values. The axis is drawn between the *Axis minimum* and *Axis maximum* values.

Axes are drawn in the coordinate system of the input object and do not reflect any changes applied via the *Transform* module.

Label Increment

The *Label increment* option sets the distance in axis units between adjacent labels. To change the spacing between labels, highlight the existing value and type a new value.

Label Angle

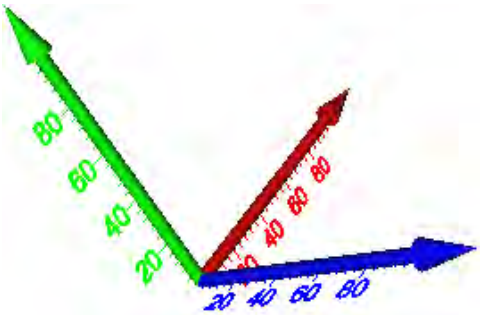
The *Label angle (degrees)* option controls the angle the label is displayed. To change the value, highlight the existing value and type a new value or click and drag the . Values are in degrees and range from -180 to + 180 degrees. Positive values rotate the label clockwise.

Label Format

Click the ⊞ next to the *Label Format* command to open the label format section. Set the *Type*, *Significant digits*, *Prefix*, and *Postfix* of axis labels.

Axes Module - Y Axis Page

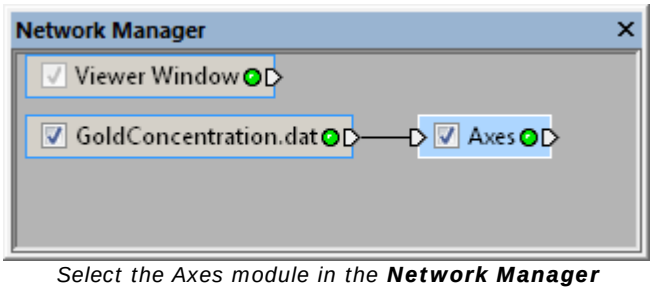
The *Axes* module **Y Axis** page displays the properties of the Y Axis. To open the **Y Axis** page, click on the *Axes* module in the **Network Manager**. In the **Property Manager**, click on the **Y Axis** tab.



*Change the axis properties in the **X Axis**, **Y Axis**, and **Z Axis** sections of the **Property Manager**.*

Properties

The **Y Axis** property sections of the *Axes* module are described below.



*Select the Axes module in the **Network Manager** to display its properties in the **Property Manager**.*

The *Axes* module contains the following tabs in the **Property Manager:**

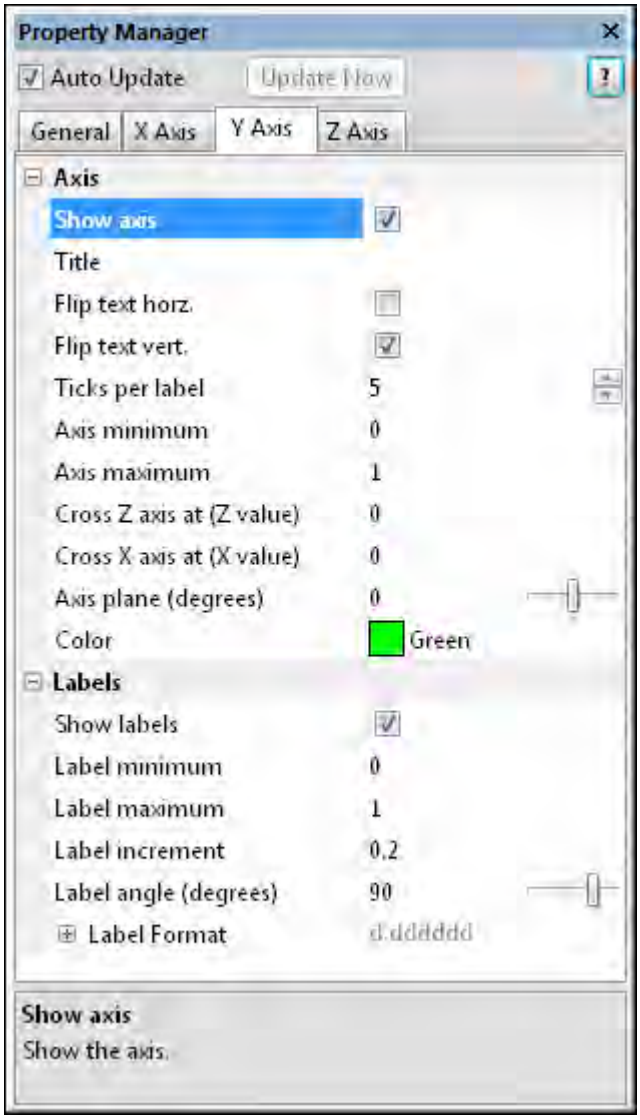
General

X Axis

Y Axis

Z Axis

General Options



*Customize the **Y Axis** properties.*

Show Axis

Check the box next to the *Show axis* command to show the axis. Uncheck the box to turn the display off for the selected axis.

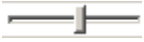
Title

Enter text in the *Title* section to give the axis a title.

Flip Text

Check the box next to the *Flip text horz.* or *Flip text vert.* commands to flip the label and title text horizontally or vertically for easier viewing after rotating the view.

Ticks per Label

The *Ticks per label* option controls how many tick mark divisions occur between each pair of labels. If the *Ticks per label* value is set to 2, one tick mark and two divisions will occur between labels. This value can be between 1 (no intermediate tick marks) and 100. To change the value, highlight the existing value and type a new number or click and drag the  to increase or decrease the number of ticks.

Axis Minimum and Maximum

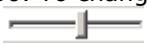
Enter a minimum data value next to *Axis minimum* for the starting axis value and a maximum data value next to *Axis maximum* for the ending axis value. Labels are drawn between the *Label minimum* and *Label maximum* values. If the *Label minimum* and *Label maximum* values are different than the *Axis minimum* and *Axis maximum* values, part of the axis will be displayed without labels.

Axes are drawn in the coordinate system of the input object and do not reflect any changes applied via the *Transform* module.

Cross Axis At

The *Cross ... axis at (... value)* value is the value at which the selected axis crosses the specified axis. Highlight the existing value and type a new value to change this. Axis values are in the coordinate system of the input object and do not reflect any changes applied via the *Transform* module.

Axis Plane

The *Axis plane (degrees)* contains the labels and tick marks, and can be specified as a rotation angle around the axis. The direction of rotation is specified by the right hand rule. Take your right hand and point your thumb in the direction of the positive rotation axis. The direction your fingers are curled is the direction of positive rotation angles. Angles are in degrees and range from -180 to +180. To change the rotation, highlight the existing value and type a new value or click and drag the  to increase or decrease the values.

Color

The *Color* section specifies the color of the axis and the axis labels and title text. To change the color, click the existing color box and select a new color from the color palette. Click on the desired color or click the *Other* button to open the **Colors** dialog to set custom colors.

Show Labels

Check the box next to the *Show labels* command to show tick marks and labels along the selected axis. Uncheck the box to hide the tick marks and labels.

Label Minimum and Maximum

Enter a minimum data value next to *Label minimum* for the first label and a maximum data value next to *Label maximum* for the last label. Labels are only drawn between the *Label minimum* and *Label maximum* values. The axis is drawn between the *Axis minimum* and *Axis maximum* values.

Axes are drawn in the coordinate system of the input object and do not reflect any changes applied via the *Transform* module.

Label Increment

The *Label increment* option sets the distance in axis units between adjacent labels. To change the spacing between labels, highlight the existing value and type a new value.

Label Angle

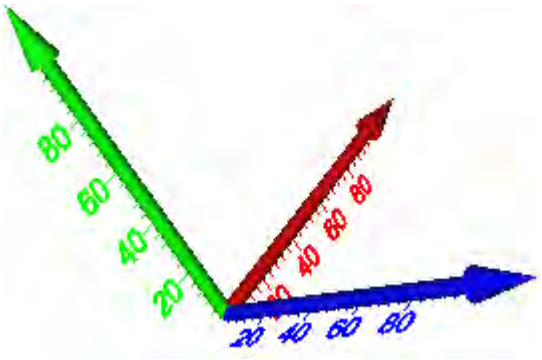
The *Label angle (degrees)* option controls the angle the label is displayed. To change the value, highlight the existing value and type a new value or click and drag the . Values are in degrees and range from -180 to + 180 degrees. Positive values rotate the label clockwise.

Label Format

Click the ⊞ next to the *Label Format* command to open the label format section. Set the *Type*, *Significant digits*, *Prefix*, and *Postfix* of axis labels.

Axes Module - Z Axis Page

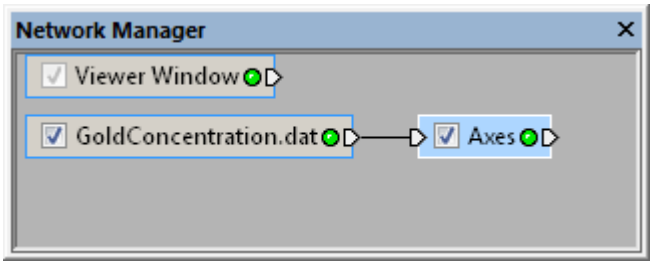
The *Axes* module **Z Axis** page displays the properties of the Z Axis. To open the **Z Axis** page, click on the *Axes* module in the **Network Manager**. In the **Property Manager**, click on the **Z Axis** tab.



*Change the axis properties in the **X Axis**, **Y Axis**, and **Z Axis** sections of the **Property Manager**.*

Properties

The **Z Axis** property sections of the *Axes* module are described below.



*Select the Axes module in the **Network Manager** to display its properties in the **Property Manager**.*

The *Axes* module contains the following tabs in the **Property Manager:**

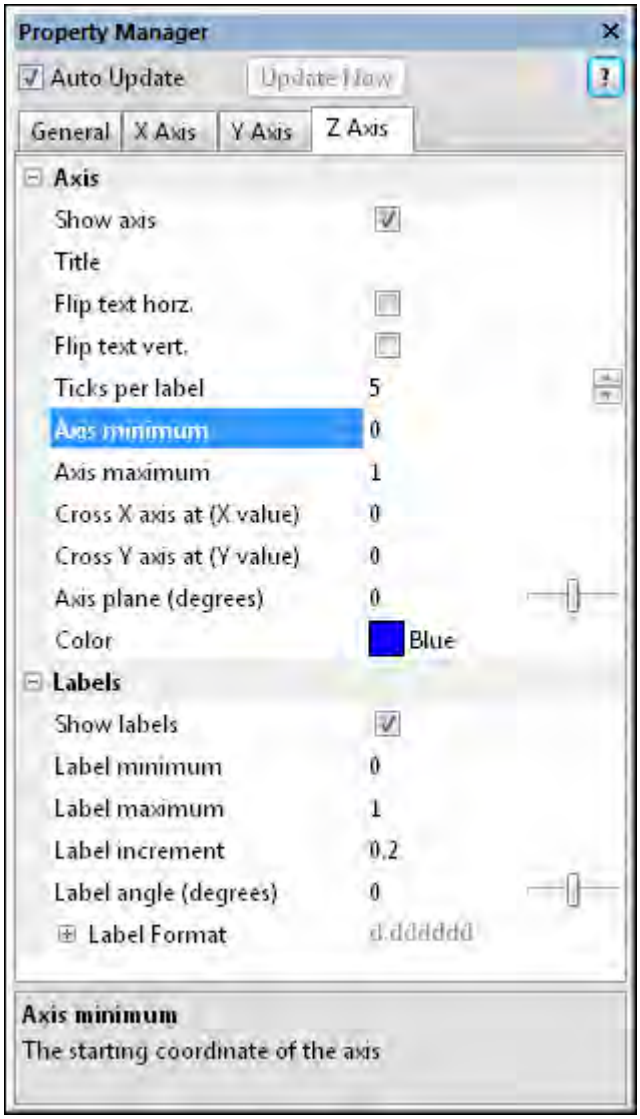
General

X Axis

Y Axis

Z Axis

General Options



*Customize the **Z Axis** properties.*

Show Axis

Check the box next to the *Show axis* command to show the axis. Uncheck the box to turn the display off for the selected axis.

Title

Enter text in the *Title* section to give the axis a title.

Flip Text

Check the box next to the *Flip text horz.* or *Flip text vert.* commands to flip the label and title text horizontally or vertically for easier viewing after rotating the view.

Ticks per Label

The *Ticks per label* option controls how many tick mark divisions occur between each pair of labels. If the *Ticks per label* value is set to 2, one tick mark and two divisions will occur between labels. This value can be between 1 (no intermediate tick marks) and 100. To change the value, highlight the existing value and type a new number or click and drag the  to increase or decrease the number of ticks.

Axis Minimum and Maximum

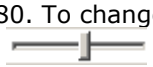
Enter a minimum data value next to *Axis minimum* for the starting axis value and a maximum data value next to *Axis maximum* for the ending axis value. Labels are drawn between the *Label minimum* and *Label maximum* values. If the *Label minimum* and *Label maximum* values are different than the *Axis minimum* and *Axis maximum* values, part of the axis will be displayed without labels.

Axes are drawn in the coordinate system of the input object and do not reflect any changes applied via the *Transform* module.

Cross Axis At

The *Cross ... axis at (... value)* value is the value at which the selected axis crosses the specified axis. Highlight the existing value and type a new value to change this. Axis values are in the coordinate system of the input object and do not reflect any changes applied via the *Transform* module.

Axis Plane

The *Axis plane (degrees)* contains the labels and tick marks, and can be specified as a rotation angle around the axis. The direction of rotation is specified by the right hand rule. Take your right hand and point your thumb in the direction of the positive rotation axis. The direction your fingers are curled is the direction of positive rotation angles. Angles are in degrees and range from -180 to +180. To change the rotation, highlight the existing value and type a new value or click and drag the  to increase or decrease the values.

Color

The *Color* section specifies the color of the axis and the axis labels and title text. To change the color, click the existing color box and select a new color from the color palette. Click on the desired color or click the *Other* button to open the **Colors** dialog to set custom colors.

Show Labels

Check the box next to the *Show labels* command to show tick marks and labels along the selected axis. Uncheck the box to hide the tick marks and labels.

Label Minimum and Maximum

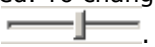
Enter a minimum data value next to *Label minimum* for the first label and a maximum data value next to *Label maximum* for the last label. Labels are only drawn between the *Label minimum* and *Label maximum* values. The axis is drawn between the *Axis minimum* and *Axis maximum* values.

Axes are drawn in the coordinate system of the input object and do not reflect any changes applied via the *Transform* module.

Label Increment

The *Label increment* option sets the distance in axis units between adjacent labels. To change the spacing between labels, highlight the existing value and type a new value.

Label Angle

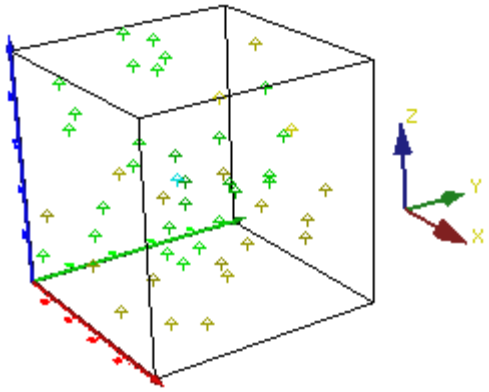
The *Label angle (degrees)* option controls the angle the label is displayed. To change the value, highlight the existing value and type a new value or click and drag the . Values are in degrees and range from -180 to + 180 degrees. Positive values rotate the label clockwise.

Label Format

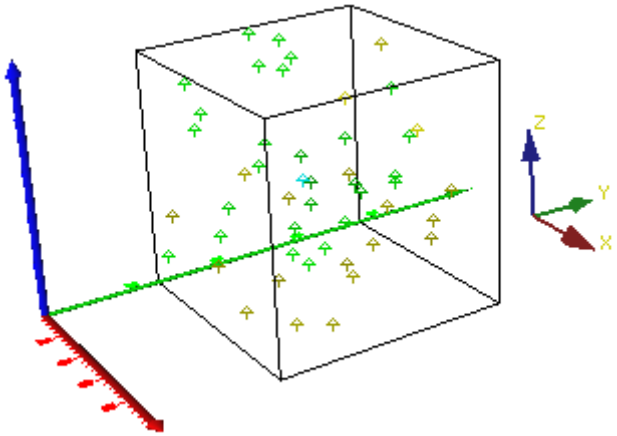
Click the ⊞ next to the *Label Format* command to open the label format section. Set the *Type*, *Significant digits*, *Prefix*, and *Postfix* of axis labels.

How do I change the scale of one of the axes?

Change the scale of an axis by inserting a *Transform module* between the input module and the *Graphics Output* module, then set the *Scale* of the *Transform* module X, Y, and/or Z directions.



The original Scale of XYZ 1:1:1.



The adjusted Scale of XYZ 1:2:1.

BoundingBox

The **Network | Graphics Output | BoundingBox** command adds a *BoundingBox* module.

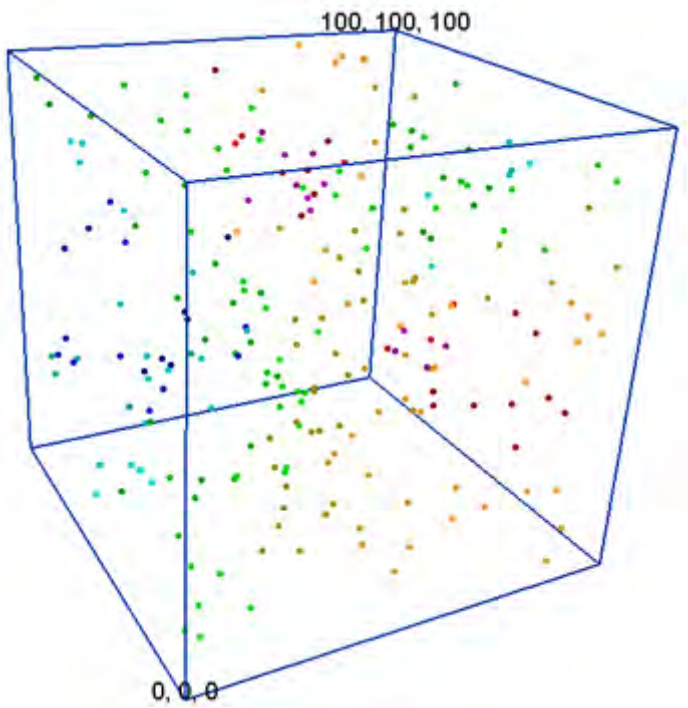
The *BoundingBox* module draws a bounding box around the input module extents. Additionally, labels can be displayed for the minimum and maximum corners. The labels are displayed as screen-aligned text centered on the minimum and maximum corners. The bounding box is useful to display the extent of your data.

Inputs

Lattice, point data, and geometry are input types for the *BoundingBox* module.

Outputs

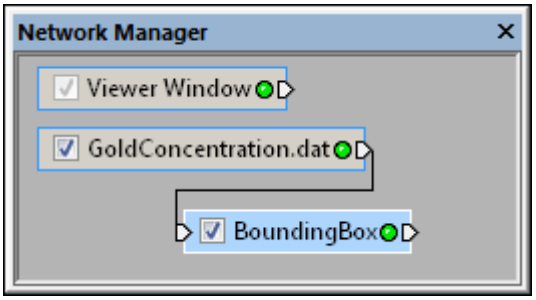
The *BoundingBox* module creates an output geometry. It may be connected to other *Graphics Output Modules*. An *Info Module* may also be connected to the output node.



The bounding box creates a box enclosing the entire area around the selected data.

Properties

The *BoundingBox* module properties are described below.



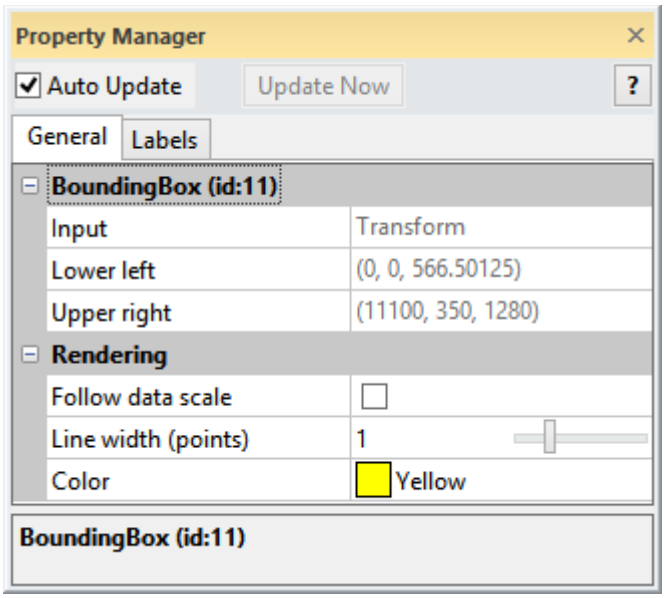
*Select the BoundingBox module in the **Network Manager** to display its properties in the **Property Manager**.*

The *Axes* module contains the following tabs in the **Property Manager:**

General

Labels

General Options



Customize the properties of the BoundingBox.

Input

The *Input* property shows the source to which the module is connected. This option cannot be changed in the **Property Manager**, but can be changed in the **Network Manager** by changing the module input.

Lower Left

The *Lower left* property shows the coordinates of the lower left corner of the bounding box. This value is for informational use only and cannot be changed.

Upper Right

The *Upper right* property shows the coordinates of the upper right corner of the bounding box. This value is for informational use only and cannot be changed.

Follow Data Scale

The *Follow data scale* property determines if the bounding box is scaled by the original input data or the output from a *Transform* module. This property has no effect when a *Transform* module is not upstream of the *BoundingBox* module. When the *Follow data scale* check box is checked, the original data is used to scale the bounding box. The bounding box label values will correspond to the transformed geometry coordinates.

When the *Follow data scale* check box is not checked, the transformed data is used to scale the bounding box. The bounding box label values will correspond to the original data coordinates. Scaling the bounding box with the transformed data (i.e. the *Follow data scale* check box is NOT checked) keeps the bounding box label values and graphics outputs in the correct relative position when compared to a non-transformed graphics output.

Line Width

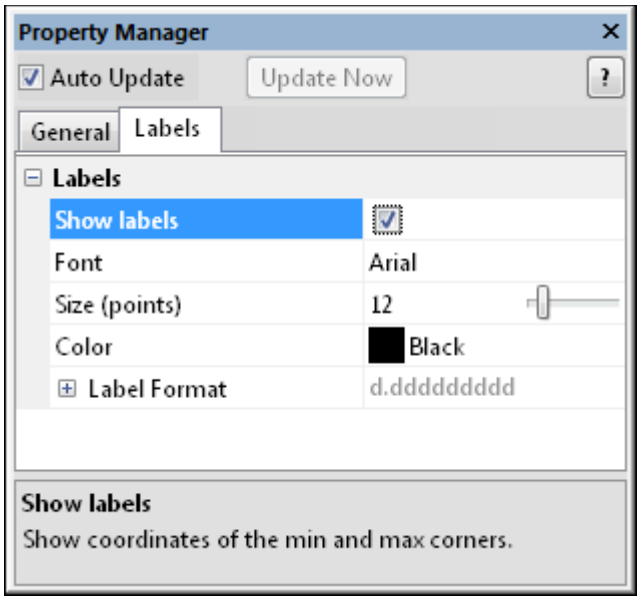
Use the *Line width (points)* option to scale the thickness of the lines in points. To change the line thickness, highlight the existing value and type a new one, or drag the  to the desired value. The thickness value is relative and can be any value between 0 (1 pixel wide) and 4.

Color

Click the existing color box next to the *Color* command to change the color of bounding box. The color palette opens to allow selection of a new color (or creation of a custom color). Click on the desired color or click the *Other* button to open the **Colors** dialog to set custom colors.

Bounding Box Module - Labels Page

The *BoundingBox* module **Labels** page displays the extents of the grid being created. To open the **Labels** page, click on the *BoundingBox* module in the **Network Manager**. In the **Property Manager**, click on the **Labels** tab.



*Set the label properties on the **Labels** tab in the Property Manager.*

Show Labels

Check the box next to the *Show labels* command to display the coordinates of the minimum and maximum corners of the bounding box. The labels are displayed as screen-aligned text centered on the minimum and maximum corners of the bounding box. Click the ⊞ next to the *Show labels* command to open the label properties section.

Font

The *Font* command specifies the text font for the bounding box labels. Click the current font and choose a new font from the drop down menu.

Size

The *Size (points)* command controls the size of the bounding box labels. The *Size (points)* is in points. Values range between 4 and 72. To change the label size, highlight the existing value and type a new value or click and drag the  to the desired value.

Color

Click the existing color box next to the *Color* command to change the color of labels. The color palette opens to allow selection of a new color (or creation of a custom color). Click on the desired color or click the *Other* button to open the **Colors** dialog to set custom colors.

Label Format

Click the ⊞ next to the *Label Format* command to open the label format section. Set the *Type*, *Significant digits*, *Prefix*, and *Postfix* of the bounding box labels.

ClipPlane

The *ClipPlane* module clips input geometry according to a user-defined clipping plane. All geometry on one side of the plane is clipped (not drawn). The side that is clipped and the location of the clipping can be altered in the **Property Manager**. Multiple input modules may be connected to the same *ClipPlane* so that each module is clipped to the same values.

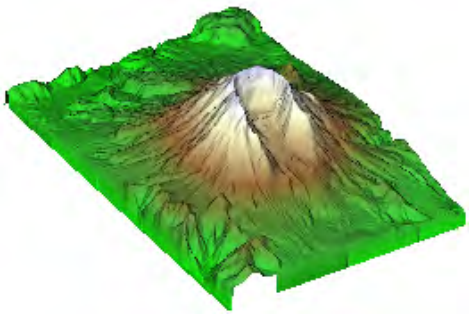
Multiple clip planes may be connected separately to the same module or connected in a series to a module. When connected separately to the same output geometry, effects like quarter cutouts, corner cutouts, etc can be accomplished. For example, one clip plane oriented along the X axis and a second clip plane oriented along the Y axis clips everything in three quarters of the plot.

Inputs

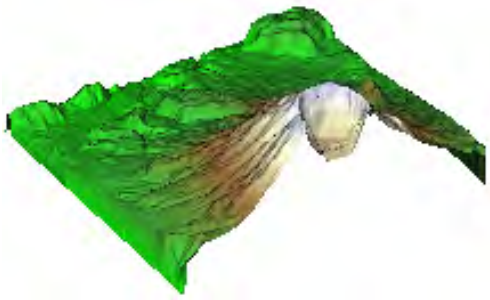
geometry

Outputs

The *ClipPlane* module creates an output geometry. It may be connected to the *Graphics Output Modules*. An *Info Module* may also be connected to the output node.



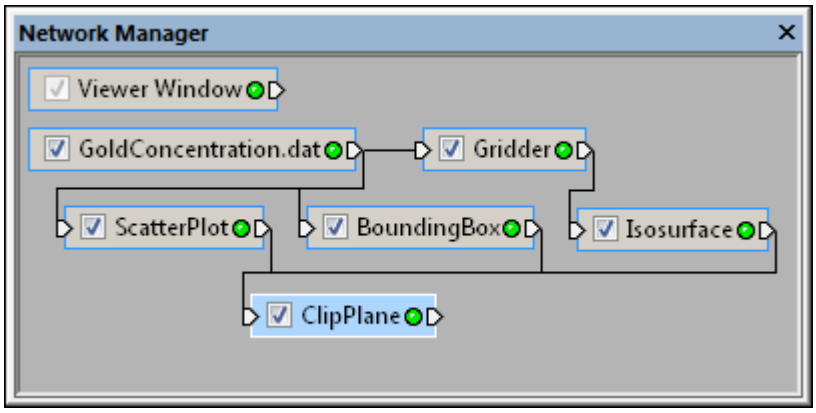
The example data for Mount St.Helens without a ClipPlane module.



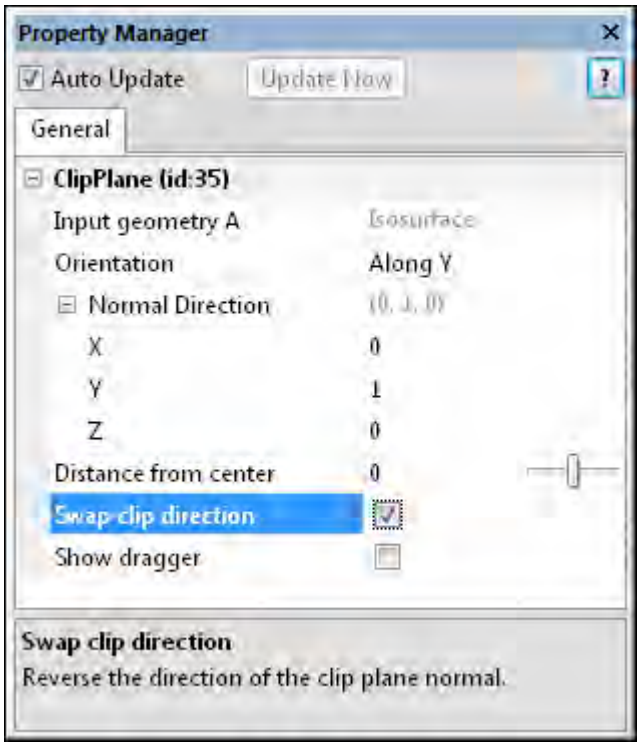
The example data for Mount St.Helens with a ClipPlane module. The ClipPlane module "clips" (does not draw) part of the geometry according to the user defined clipping plane.

Properties

The *ClipPlane* module properties are described below.



*Select the ClipPlane module in the **Network Manager** to display its properties in the **Property Manager**.*



Customize the ClipPlane properties.

Input Geometry

The *Input geometry* property shows the source to which the module is connected. This option cannot be changed in the **Property Manager**, but can be changed in the **Network Manager** by changing the module input. If multiple modules are connected, each module is listed on a separate *Input geometry* line.

Orientation

The *Orientation* is the direction of the clip plane normal. To change the orientation, click on the existing option and choose the desired option from the list. Options available are *Along X*, *Along Y*,

Along Z, and *Custom*. The *Along* option clips the modules so that a portion of the data normal to the axis plane is removed. *Custom* allows the clipping to occur along a user defined plane.

Normal Direction

The *Normal Direction* is the X, Y, and Z components of the normal to the clip plane. A plane is any flat, two-dimensional surface. A normal to a flat surface is a vector that is perpendicular to that surface. The *ClipPlane* is specified by the normal to the plane and an offset along the normal. The normal is specified by the X, Y, and Z components of a vector. The offset is the distance of the plane from the center of the input geometry. To change the *X*, *Y*, or *Z* component value, highlight the existing value and type a new value.

Distance from Center

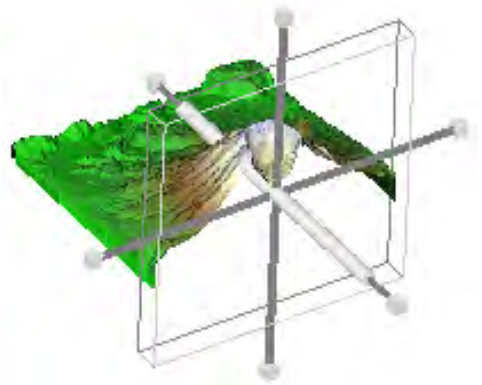
Distance from center is the distance of the clip plane from the center of the input geometry. A value of zero places the ClipPlane directly at the center of the input geometry. Larger positive values place the ClipPlane farther away from the center toward higher axis values. Smaller negative values place the ClipPlane farther away from the center toward the lower axis values. Units are in coordinate units. To change the distance, highlight the existing value and type a new value or click and drag the  to increase or decrease the distance location.

Swap Clip Direction

Check the box next to the *Swap clip direction* command to reverse the direction of the clip plane normal. This alters which portion of the area is displayed.

Show Dragger

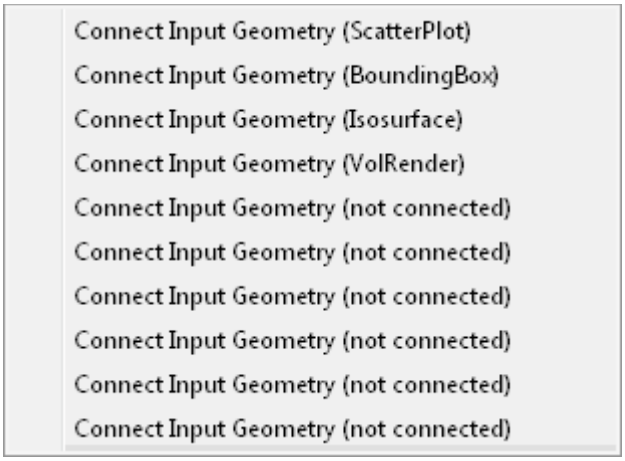
Check the *Show dragger* box to show the clip plane dragger and enable interactive rotation and offset. Invoke rotations by clicking and dragging the line parts of the three principal "axes" of the dragger geometry. Choose the solid middle part of the jack dragger axis to translate. See *Dragger* properties for more details.



Use the ClipPlane dragger (in gray and white above) to interactively change the position of the clip plane.

Connecting Multiple Modules to a Single ClipPlane

To connect multiple modules to a single ClipPlane, create each of the modules and add one ClipPlane to the network using the **Module Manager** or the **Network | Graphics Output | ClipPlane** command. Once the *ClipPlane* module appears in the **Network Manager**, you can connect your graphics modules to it. After dragging a connection from the graphics output to the ClipPlane, select which of the *Connect Input Geometry Nodes* to connect.



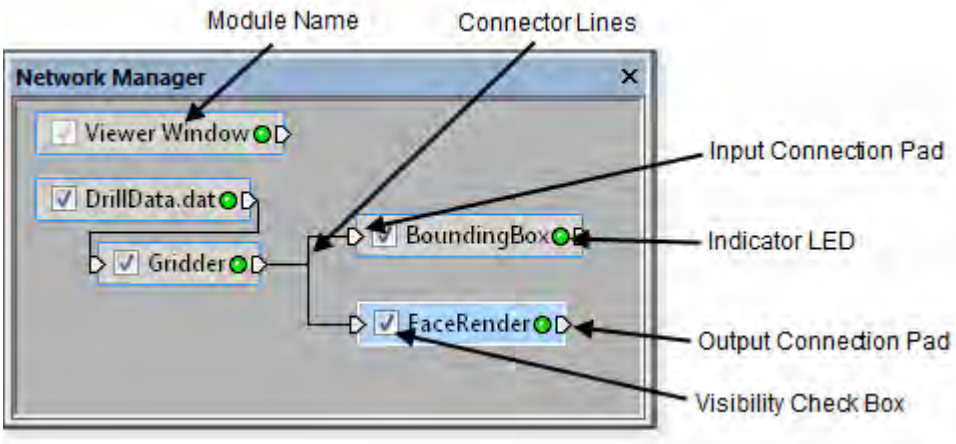
Select any of the available (not connected) Input Geometry nodes when attaching a new module to the ClipPlane.

Chapter 7 - Annotation and Text

Introduction to Modules

A module is a data set or a process to be applied to a data set or process. Modules are the building blocks from which the final **Voxler** output is constructed. Modules accept data on their input connection pads ▷, modify the data, and pass it along through the output connection pad ▷.

Modules are displayed in the **Network Manager**. You can connect and disconnect modules to create a visualization network representing the flow of data. Modules need to be connected in order to generate an output. The current geometry output of the modules in the **Network Manager** are displayed in a three-dimensional view in the **Viewer** window. If the data is not connected to a graphics output module, nothing is displayed in the **Viewer** window.



The **Network Manager** displays the visualization network, which includes all loaded modules and their connections.

View All Modules

All modules are accessed using the Network menu commands. Alternatively, all modules are accessed in the **Module Manager** when the *Show all modules* button is selected .

View Applicable Modules

When a module is selected in the **Network Manager** and the *Show all modules* button is not selected in the **Module Manager**, available modules that can be connected to the selected module output port are displayed in the **Module Manager**. Alternatively, right-click a module in the **Network Manager** to display only the applicable modules in the context menu.

On the module description pages, there are **Inputs** and **Outputs** sections to discuss the type of input and output modules that each module are compatible.

Import

Click **File | Import** to open the **Import** dialog. The type of data determines what type of module can be attached to it. **Voxler** supports several different data types. See the *File Format Chart* for a detailed list of supported file formats.

Viewer Window

The **Viewer Window** is a unique module that is automatically created when a new instance of **Voxler** is generated. The *Viewer Window* module appears in the **Network Manager**. The *Viewer Window* module cannot be deleted. The purpose of the *Viewer Window* module is to control the properties of the **Viewer** window for the current instance of **Voxler**. To change the **Viewer** window properties for future instances of **Voxler**, choose the **Tools | Options** command and adjust properties on the **Colors** page of the **Options** dialog.

Module Types

There are four types of modules: *computational*, *data source*, *general*, and *graphics output.* Each module type is discussed below.

Computational Modules

Computational modules alter the data by changing the data type, filtering, creating a gradient, gridding, performing mathematical transformations, merging, resampling, slicing, creating a subset, or transforming coordinates. Click on one of the following computational modules for detailed information on using the module and module properties.

ChangeType

DuplicateFilter

ExclusionFilter

ExtractPoints

Filter

Gradient

Gridder

Math

Merge

Resample

Slice

Subset

Transform

Data Source Modules

Data source modules serve as a source of raw data. The data can be imported or created from mathematical functions. Click on one of the following data source modules for detailed information on using the module and module properties.

Import (*Data Source*, *Point Source*, *Lattice Source*, *Geometry Source*)

FunctionLattice

TestLattice

WellData (combines multiple *Data Source* modules into a single output)

General Modules

General modules display module information and provide custom lighting in the **Viewer** window. Click on one of the following general modules for detailed information on using the module and module properties.

Info

Light

Graphics Output

Graphics output modules create graphics in the **Viewer** window. Typically, these modules require data input. Click on one of the following graphics output modules for detailed information on using the module and module properties.

Annotation

Axes

BoundingBox

ClipPlane

Contours

FaceRender

HeightField

Isosurface

ObliqueImage

OrthoImage

ScatterPlot

StreamLines

Text

VectorPlot

VolRender

WellRender

Graphics Output

Choose the **Network | Graphics Output** command to add data-dependent graphics to the scene.

Graphics output modules create graphics in the **Viewer** window. Typically, these modules require data input.

The modules are added to the menu in alphabetical order to make them easier to find. If a module is selected in the **Network Manager** when this command is chosen, then only those output modules compatible with the selected module are displayed. Choose a module from the list and **Voxler** automatically adds it to the selected module.

Graphics output modules include the following:

- *Annotation* displays a two-dimensional screen-aligned block of text
- *Axes* displays a set of axes for the input modules
- *BoundingBox* draws a three-dimensional bounding box around the input object's geometry
- *ClipPlane* limits the input geometry

- *Contours* draws one or more contour lines (isolines) in a plane through a three-dimensional lattice
- *FaceRender* displays cubes along the faces of a three-dimensional lattice
- *HeightField* displaces the "height" of a lattice slice according to a single component
- *Isosurface* computes an isosurface through a three-dimensional lattice and outputs the geometry
- *ObliqueImage* displays an arbitrary slice through a lattice
- *OrthoImage* displays an orthogonal slice through a lattice
- *ScatterPlot* displays a point set or lattice as a series of symbols
- *StreamLines* displays streamlines from a multiple component lattice
- *Text* allows the addition of an anchored text frame to the scene
- *VectorPlot* displays vectors on a three-dimensional lattice or point set
- *VolRender* performs direct volume rendering of a three-dimensional lattice
- *WellRender* displays well data as lines or tubes

Annotation

The **Network | Graphics Output | Annotation** command adds an *Annotation* module.

The *Annotation* module creates a text string that is always parallel to the scene. The text appears attached to the screen and is not part of the scene; as such, the output from this module cannot be clipped by a clipping plane. By default, the current date and time is used as the text string. You can enter your own text in the **Property Manager**. Use the *Text module* to anchor the text to the scene. This can be useful to add text to the screen that does not move with the graphical output.

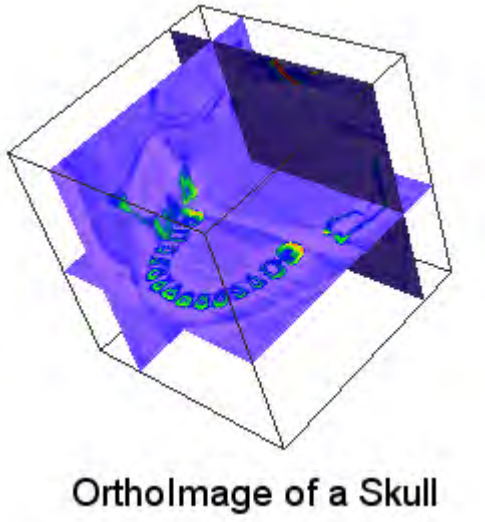
The *Annotation* module does not specify a bounding box for the text because the text is projected onto the screen and is not in the same coordinate system as the rest of the graph. If an *Info* module is attached, nothing is displayed for the bounding box.

Inputs

The *Annotation* module is not connected to any module on the input side.

Outputs

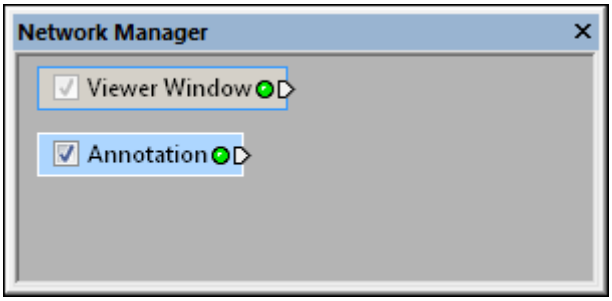
The *Annotation* module creates an output geometry. It may be connected to the *Graphics Output Modules*. An *Info Module* may also be connected to the output node.



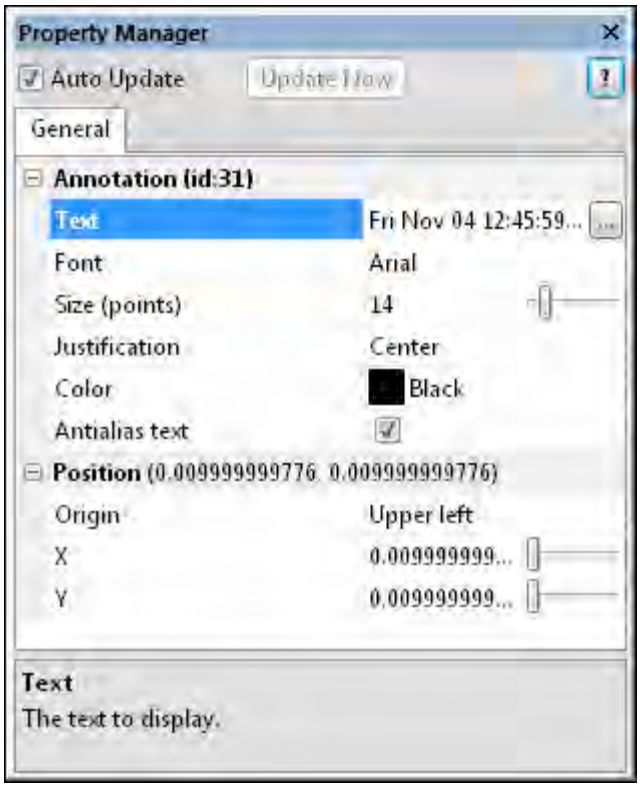
Use the Annotation module to add text to the screen that does not move with the graphical output, such as the text "OrthoImage of a Skull" in this example.

Properties

The *Annotation* module properties are described below.



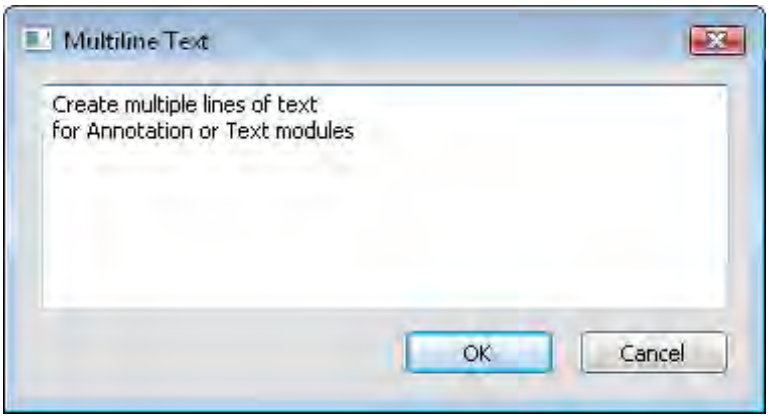
*Select the Annotation module in the **Network Manager** to display its properties in the **Property Manager**.*



*Change the Annotation settings in the **Property Manager** to move the text or change the text that appears.*

Text

The *Annotation* module displays an arbitrary screen-aligned text string. The text is always perpendicular to the viewer's line of sight. By default, the current date and time are used as the text string. Click the ... button to open the **Multiline Text** dialog to enter custom annotation text. To create multiple lines of text, press ENTER to start a new line. Click *OK* to make the text change and view it in the **Viewer** window. Click *Cancel* to return to the properties without making the change.

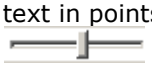


*Enter custom text in the **Multiline Text** dialog.*

Font

The *Font* section specifies the text font. To change the font, click the current font and choose a new font from the list.

Size

The *Size (points)* section determines the size of the text in points. To change the value, highlight the existing value and type a new value or drag the  to the increase or decrease the value.

Justification

The *Justification* indicates how multiple lines of text should appear. The *Justification* has no affect on a single line of text. The options available are *Left, Center*, and *Right.* To change the multiple line text *Justification*, click the existing option and select the desired option from the list. Choose *Left* to align the text to the left side of the text block. Choose *Right* to align the text to the right side of the text block. Choose *Center* to center all text in the text block.

Color

The *Color* section specifies the text color. All text in the *annotation* has the same color. To change the color, click the colored box and choose a new color from the color palette. If the basic colors in the palette do not meet your needs, click *Other* to create a custom color.

Antialias Text

Check the box next to the *Antialias text* option to make the fonts appear smoother by slightly adjusting the colors. This usually makes the text appear somewhat more dim.

Position

Click the  next to *Position* to open the *Position* section. This section locates the text on the screen.

Origin

The *Origin* is the **Viewer** window location where the annotation should be located. Available options are *Upper left*, *Upper right*, *Lower left*, and *Lower right.* To change the location, click on the existing option and select the desired location from the list. To further customize the location of the text, adjust the *X* and *Y* values. The *Origin* is relative to the *X* and *Y* values.

X and Y

The *X* and *Y* sections display the coordinates of the text in **Viewer** window coordinates ranging from 0.0 to 1.0. 0.0 places the *Origin* of the text at the left edge for *X* or the top edge for *Y* of the **Viewer** window. 1.0 places the *Origin* of the text at the right edge for *X* or bottom edge for *Y* of the **Viewer** window. To change either value, highlight the existing value and type a new value or drag the  to increase or decrease the value.

Text

The **Network | Graphics Output | Text** command adds a *Text* module.

The *Text* module allows addition of a two-dimensional text string aligned with the camera plane. The text has a three-dimensional anchor point that is transformed with the scene.

The text is not scaled according to the distance from the camera, nor is it influenced by rotation or scaling. It is, however, still obscured by graphics lying in front of it. The text is positioned according

to the current transformation: The X origin is the first pixel of the leftmost character of text and the Y origin is the baseline of the first line of text with the baseline being the imaginary line on which all upper case characters are standing.

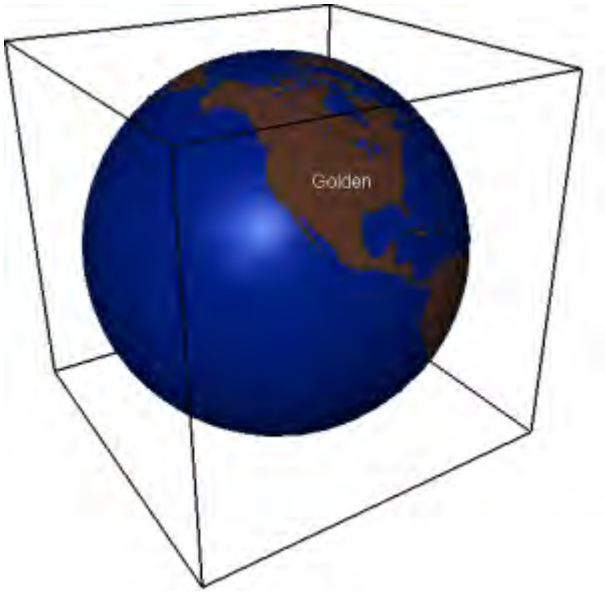
Use the *Annotation module* to create text that is not anchored to the scene. If multiple anchored text items need to be created, use a *ScatterPlot* and add the text as a label column

Inputs

The *Text* module is not connected to any module on the input side.

Outputs

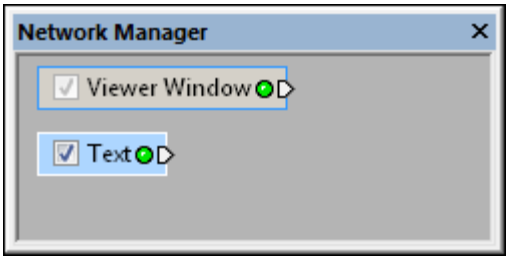
The *Text* module creates an output geometry. It may be connected to the *Graphics Output Modules*. An *Info Module* may also be connected to the output node.



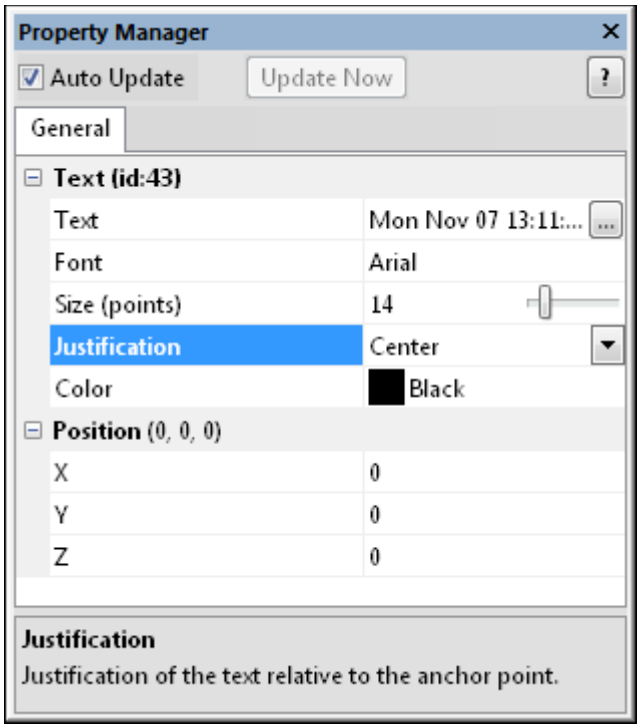
Use the Text module to add text to the scene that moves with the graphical output.

Properties

The *Text* module properties are described below.



*Select the Text module in the **Network Manager** to display its properties in the **Property Manager**.*

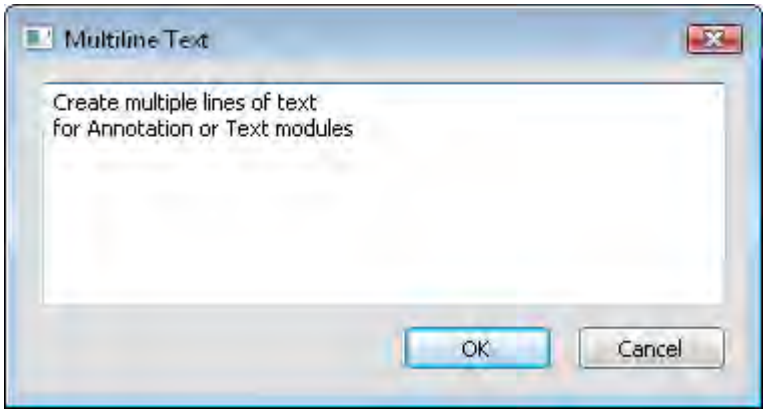


Customize the Text properties.

Text

The *Text* module displays a screen-aligned text string. The text is always perpendicular to the viewer's line of sight. By default, the current date and time are used as the text string. To change the text, highlight the existing text and type a new text string.

Click the ... button to open the **Multiline Text** dialog, where multiple lines of text can be entered. To enter a new line of text, press the ENTER key on the keyboard. To close the **Multiline Text** dialog, click *OK* to make the change or *Cancel* to return to the text module properties without making the change.

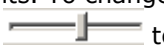


*Enter custom text in the **Multiline Text** dialog.*

Font

The *Font* option sets the font used for writing the text. All available TrueType fonts are listed as options. To change the font, click the current font name and select the desired font name from the list.

Size

The *Size (points)* section determines the size of the text in points. Values range from 4 to 288 points. To change the value, highlight the existing value and type the desired size or click and drag the  to the desired value.

Justification

The *Justification* determines how the anchor point is placed on the **Viewer** window. Available options are *Left*, *Center*, and *Right*. The justification is relative to the three-dimensional anchor point. *Left* anchors the left edge of the text to the anchor point, *Right* anchors the right edge of the text to the anchor point, and *Center* anchors the middle of the text to the anchor point. To change the justification, click on the existing option and select the desired option from the list.

Color

The *Color* section specifies the text color. To change the color, click the colored box and choose a new color from the color palette. If the basic colors in the palette do not meet your needs, click *Other* to create a custom color.

X, Y, and Z

The *X*, *Y*, and *Z* sections display the coordinates of the text in relation to the scene. To change a value, highlight the existing value and type the desired value.

Chapter 8 - ScatterPlots

ScatterPlot Module

The **Network | Graphics Output | ScatterPlot** command adds a *ScatterPlot* module.

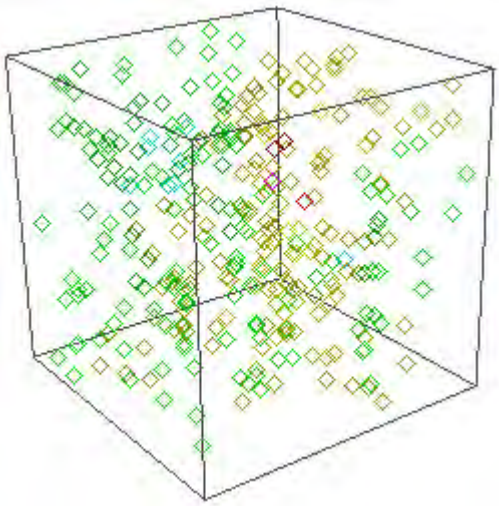
The *ScatterPlot* module displays a set of symbols at each point of a point set or each node of a lattice. The symbols are screen aligned and do not scale or "tilt" as the camera changes. The positions of the symbols, however, are maintained in three dimensions.

Inputs

Lattice and point data are input types for the *ScatterPlot* module.

Output

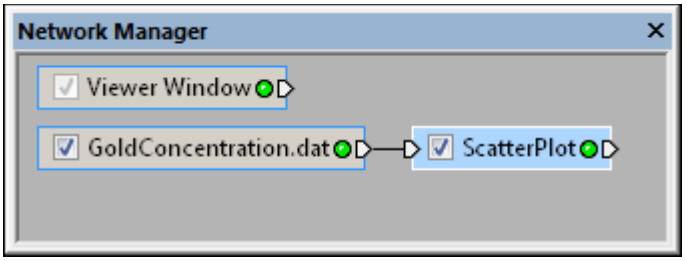
The *ScatterPlot* module creates an output geometry. It may be connected to the *Graphics Output Modules*. An *Info Module* may also be connected to the output node.



The Diamond Line symbol, shown above, is one of the many Symbol types that can be used by the ScatterPlot module.

Properties

The *ScatterPlot* module properties are described below.



*Select the ScatterPlot module in the **Network Manager** to display its properties in the **Property Manager**.*

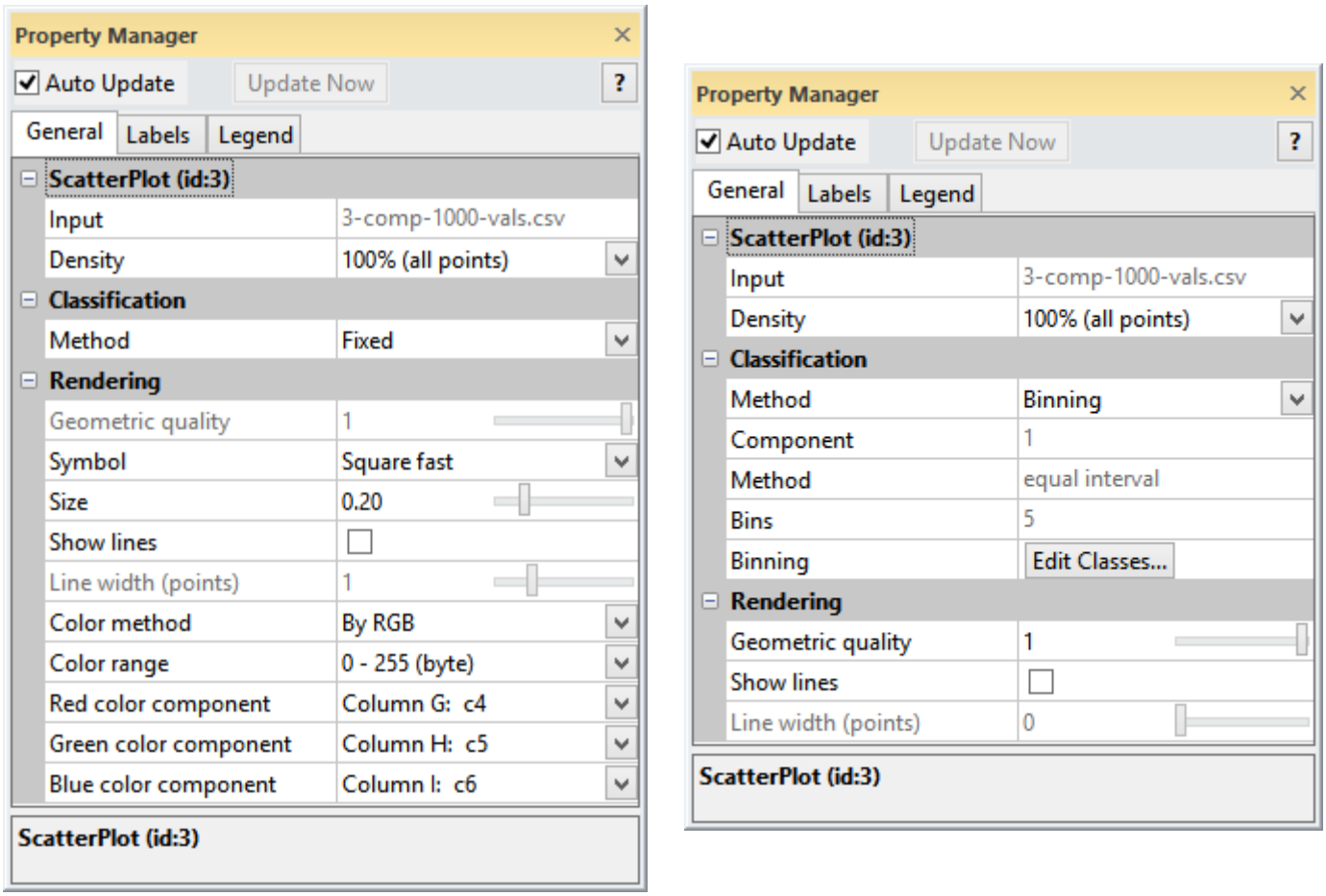
The *ScatterPlot* module contains the following tabs in the **Property Manager:**

General

Labels

Legend

General Options



*Customize the ScatterPlot properties in the **General** page of the **Property Manager**.*

Input

The *Input* property shows the source to which the module is connected. This option cannot be changed in the **Property Manager**, but can be changed in the **Network Manager** by changing the module input.

Density

The *Density* determines the number of points to plot as a percentage of the total number of points in the data file. Possible options are *100% (all points)*, *50% (every other point)*, *33% (every 3rd point)*, *25% (every 4th point)*, *20% (every 5th point)*, *10% (every 10th point)*, *5% (every 20th point)* and *1% (every 100th point)*. To change the density of plotted points, click on the existing option and select the desired value from the list.

Classification

The *Classification* section of the **General** page controls the classification method and displays information about the classes when the *Binning* method is selected.

Method

Select the *Fixed* or *Binning* classification method from the *Method* list in the *Classification* section. The *Fixed* method uses one symbol and size for the points in the scatterplot. The symbol color can be fixed, mapped by component value to a colormap, or specified by RGB color columns with the *Fixed* method. The *Binning* method separates points of the scatterplot into classes or bins. The symbol, size, and color for each bin is specified separately in the Edit Classes dialog.

Component

The value displayed in the *Component* field is the component used to separate the scatterplot points into bins. Change the *Component* in the **Edit Classes** dialog. The *Component* property is only displayed when the classification *Method* is set to *Binning.*

Method

The *Method* property displays the binning method. It is either *equal interval* or *equal bins*. Change the binning *Method* in the **Edit Classes** dialog. The binning *Method* property is only displayed when the classification *Method* is set to *Binning.*

Bins

The *Bins* property displays the number of bins. Change the number of *Bins* in the **Edit Classes** dialog. The *Bins* property is only displayed when the classification *Method* is set to *Binning*.

Binning

Click the *Edit Classes* button to specify the binning options in the Edit Classes dialog.

Rendering

The *Rendering* section of the **General** page contains options for the scatterplot display. The properties displayed in the *Rendering* section are dependent on the classification *Method*.

Geometric Quality

The *Geometric quality* property controls the 3D quality for the 3D symbols, i.e. the *3D Cube*, *3D Cone*, *3D Cylinder*, and *3D Sphere* symbols. The *Geometric quality* value can be between 0 and 1. 3D symbols are rendered with higher quality and smoothness as the *Geometric quality* value is increased. Increasing the value also increases the relative drawing time. The *Geometric quality* property is only enabled when a 3D symbol is used.

Symbol

Choose the type of symbol to draw at each point when the classification *Method* is set to *Fixed*. The symbol *Square fast* is drawn using a native OpenGL primitive and can be drawn faster than the other symbols. The *Symbol* list includes the following symbol types: *3D Cone*, *3D Cube*, *3D Cylinder*, *3D Sphere*, *Backslash*, *Bar*, *Caution filled*, *Caution line*, *Circle filled*, *Circle line*, *Cross*, *Diamond filled*, *Diamond line*, *Hourglass filled*, *Hourglass lined*, *Lightning*, *Minus*, *Pine tree filled*, *Pine tree line*, *Plus*, *Rhombus filled*, *Rhombus line*, *Satellite filled*, *Satellite line*, *Ship filled*, *Ship line*, *Slash*, *Square fast*, *Square filled*, *Square line*, *Star*, *Triangle filled*, *Triangle line*, *Well*, and *Y .* To change the symbol, click on the existing option and select the desired option in the list.

Size

The *Size* controls the size of the *Symbol*. The size ranges from 0 to 1 . To change the size, highlight the existing value and type a new value or click and drag the  to increase or decrease the size. The larger the value, the larger the symbol appears.

Show Lines

Check the box next to the *Show lines* command to draw a connecting line between sequential points. Lines are drawn in the order that points appear in the worksheet. Line color is determined by the *Color method*.

Line Width

The *Line width (points)* is the thickness of the line that connects points. Values are in points and range from zero to four. When the width is set to zero, the line is one pixel wide. The larger the value, the thicker the line. To change the value, highlight the existing value and type the desired width or click and drag the  to increase or decrease the value.

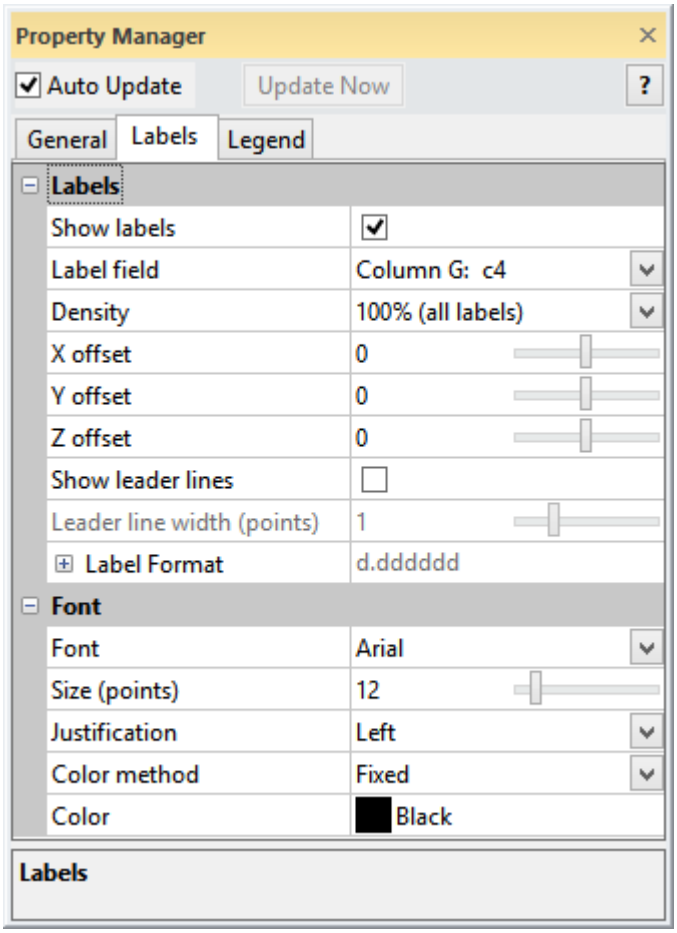
Color Method

The *Color method* determines the label and label line color. Options are *Fixed*, *By colormap*, and *By RGB.*

- When set to *Fixed,* the *Color* option is available. To change the symbol color, click the current *Color* selection and select the desired color from the color palette. Click *Other* in the color palette to select or create a color in the **Colors** dialog.
- When set to *By colormap*, the symbol color is determined by the *Colormap* selection. The *By colormap* property is available only if the input data has at least one data component. To change the colormap, click the current *Colormap* selection and select the desired colormap from the list or click the  button to select or edit the colormap in the **Colormap Editor**.
- When set to *By RGB* the symbol color is specified by columns in the data file. The *Color range* property specifies if the color data is *0 - 255 (byte)* or *0.0 - 1.0 (double)* format. Select the appropriate worksheet columns in the *Red color component*, *Green color component*, and *Blue color component* lists.

ScatterPlot Module - Labels Page

The *ScatterPlot* module **Labels** page displays the options for showing labels for scatter plot points. To open the **Labels** page, click on the *ScatterPlot* module in the **Network Manager**. In the **Property Manager**, click on the **Labels** tab.



*Edit the label properties on the **Labels** tab for a ScatterPlot.*

Show Labels

Check the box next to the *Show labels* command to display labels at each point on the *ScatterPlot*. The field to use to display labels is set in the *Label field* option.

Label Field

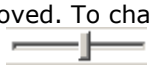
The *Label field* is the value displayed as the label on the *ScatterPlot*. To change the label displayed, click on the existing *Label field* option and select the desired option from the list. Available options are *X*, *Y*, *Z*, *XYZ*, and all *Label columns* specified in the *Data Source* properties. A *Component* column can be specified as a *Label* column if you wish to plot the *Component* value as a label. If multiple label columns are specified in the *Data Source* module properties, each label column will be listed as a separate option. When a label column is selected, the value in the column is displayed next to the symbol. When *X*, *Y*, *Z*, or *XYZ* is selected, the location value is displayed. For *X*, *Y*, and *Z*, only the selected value is displayed. When *XYZ* is selected, the labels are displayed as *X*, *Y*, *Z*.

Density

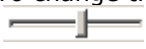
The *Density* determines the number of labels to plot as a percentage of the total number of points displayed on the *ScatterPlot*. Due to this relationship, the percentage of labels compared to the number of data points in the data file is the *ScatterPlot* point *Density* (set on the **General** page) times the label *Density* (e.g. 50% point density x 50% label density = 25%). Also because of this, labels are only plotted for visible *ScatterPlot* points.

The possible Density values are 100% (all labels), 50% (every other label), 33% (every 3rd label), 25% (every 4th label), 20% (every 5th label), 10% (every 10th label), 5% (every 20th label) and 1% (every 100th label). To change the density of plotted labels, click on the existing option and select the desired value from the list.

X Offset

The *X offset* moves the screen position of the label away from the symbol along the X axis. Values are in pixels and range from -100 pixels to +100 pixels. From the top view for example, a negative value moves the label to the left, and a positive value moves the label to the right. The label stays aligned with the screen regardless of how it is moved. To change the offset, highlight the existing value and type a new value or click and drag the  to the desired value.

Y Offset

The *Y offset* moves the screen position of the label away from the symbol along the Y axis. Values are in pixels and range from -100 pixels to +100 pixels. From the top view for example, a negative value moves the label below the symbol, and a positive value moves the label above the symbol. The label stays aligned with the screen regardless of how it is moved. To change the offset, highlight the existing value and type a new value or click and drag the  to the desired value.

Z Offset

The *Z offset* moves the screen position of the label away from the symbol along the Z axis. Values are in pixels and range from -100 pixels to +100 pixels. From the top view for example, a negative value moves the label behind the symbol, and a positive value moves the label in front of the symbol. The label stays aligned with the screen regardless of how it is moved. To change the offset, highlight the existing value and type a new value or click and drag the  to the desired value.

Display Leader Lines

Check the box next to the *Show leader lines* option to display lines connecting the label and *ScatterPlot* symbol.

Line Width

Use the *Leader line width (points)* option to scale the thickness of the line that connects the point to the label. Values are in points and range from zero to four points. To change the line thickness, highlight the existing value and type a new one, or drag the  to the desired value.

Label Format

Click the ⊞ next to *Label Format* to open the label format section. The *Type*, *Significant digits*, *Prefix*, and *Postfix* can be set for the *ScatterPlot* labels.

Font

The *Font* option sets the font used for writing the labels. All available TrueType fonts are listed as options. To change the font, click the current font name and select the desired font name from the list.

Size

The *Size (points)* section sets the size of the labels in points. Values range from 4 to 288 points. The larger the value, the larger the text appears on the screen. To change the value, highlight the existing value and type a new value or click and drag the  to the desired size.

Justification

Choose *Left*, *Right*, or *Center* justification for the label. The justification is relative to the three-dimensional anchor point. *Left* anchors the left edge of the text to the anchor point, *Right* anchors the right edge of the text to the anchor point, and *Center* anchors the middle of the text to the anchor point.

Color Method

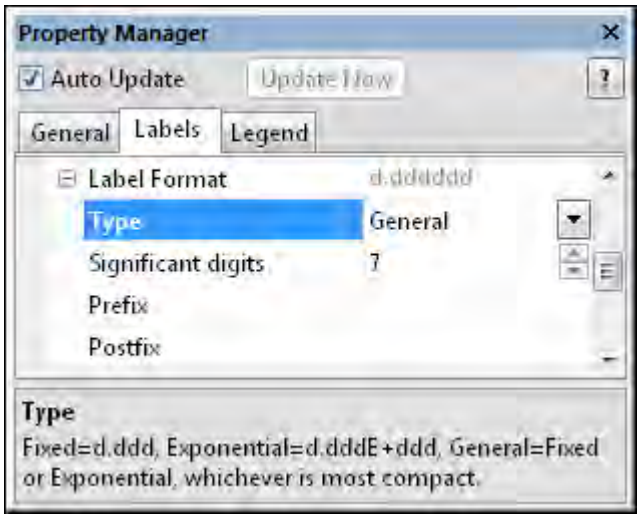
The method used to generate the symbol colors. The color of the symbols may be set to a *Fixed* color or to *By data*, which matches a colormap with the component value. To change the *Color method*, click on the existing option and select the desired option from the list.

When set to *Fixed*, the *Color* option becomes available. The *Color* is the color of the symbol. To change the color, click on the existing color and select the desired color from the list. Alternatively, click the *Other* box to open the **Colors** dialog, where you can set custom colors.

When the *Color method* is set to *By data*, the *Colormap* for the symbols is used for the labels.

Label Format

Axis, *ScatterPlot* labels, and *legend* label formats may be changed. Several options exist for changing label format properties:

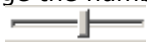


*Customize the Label Format in the **Property Manager**.*

Label Type

The *Type* display the numeric format to use for the labels. Available options are *Fixed*, *Exponential*, or *General*. To change the label type, click on the existing option and select the desired option from the list. Numbers displayed as *Fixed* appear as d.ddd; *Exponential* appear as d.dddE+ddd; and *General* appears in the format that uses less characters.

Significant Digits

The *Significant digits* specifies the number of digits that should appear to the right of the decimal point for *Fixed* or *Exponential* label types, or how many total digits should be considered significant for *General* labels. To change the number of digits, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.

As an example, consider the numbers 7.45 and 15500:

Label Type	Significant digits	7.45 displays as	15500 displays as
Fixed	2	7.45	15500
Fixed	0	7	15500
Exponential	2	7.45E+000	1.55E+004
Exponential	0	7E+000	2E+004
General	2	7.4	1.5E+004
General	1	7	2E+004

Prefix

The *Prefix* is any text that appears displayed before each label. For example, a prefix of *GS* would change the labels 1, 2, 3, 4 ,5 to GS1, GS2, GS3, GS4, GS5. To insert a prefix, click in the empty box next to *Prefix* and type the desired text.

Postfix

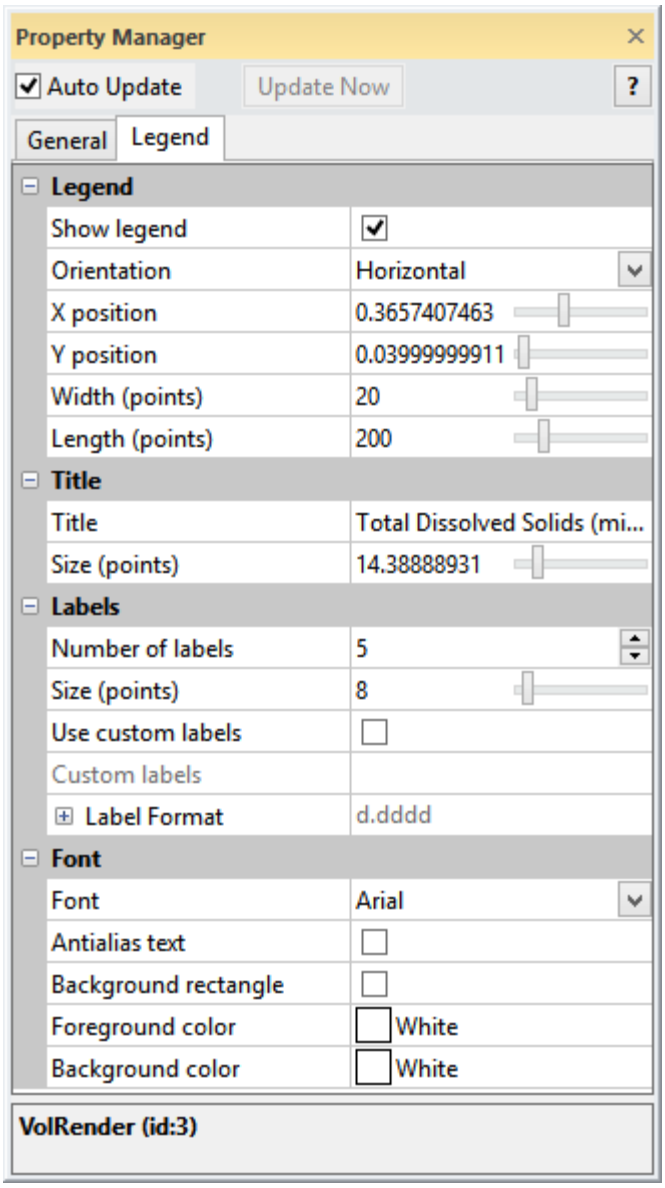
The *Postfix* is any text that appears displayed after each label. For example, a postfix of *ppm* would change the labels 1, 2, 3, 4 ,5 to 1ppm, 2ppm, 3ppm, 4ppm, 5ppm. To insert a postfix, click in the empty box next to *Postfix* and type the desired text.

Legend

A legend is a key to a plot that labels data by color, pattern, or symbol. Legends can be added to many modules, including *FaceRender*, *HeightField*, *Isosurface*, *ScatterPlot*, *StreamLines*, *VectorPlot*, *VolRender*, and *WellRender*.

Properties

Select a module that supports legends. In the **Property Manager**, click the *Show* button next to *Legend* to display the legend in the **Viewer** window. The legend properties are described below.



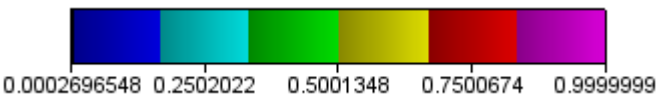
Customize the Legend properties in the ***Property Manager****.*

Show Legend

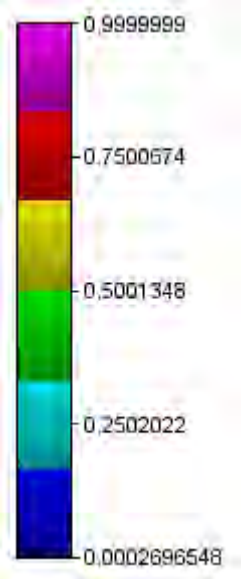
Check the box next to the *Show legend* option to display the legend for the selected module.

Orientation

Orientation specifies the direction of the "length" dimension. Available options are *Horizontal* and *Vertical* length dimension. To change the *Orientation*, click on the existing option and select the desired option from the list.



This legend has a Horizontal Orientation.



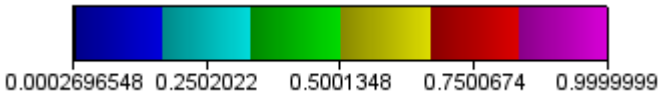
This legend has a Vertical Orientation.

X and Y Position

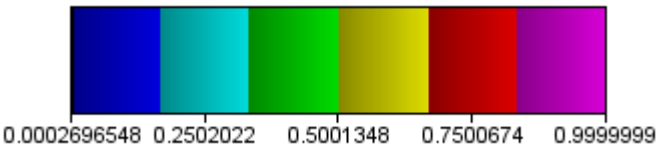
The *X position* and *Y position* control the location of the legend in the **Viewer** window. Adjust the X and/or Y position value to move the legend location. Values range from 0.0 to 1.0. 0.0 places the left bottom edge of the legend at the left edge for *X position* or the bottom edge for *Y position* in the **Viewer** window. 1.0 places the left bottom edge of the legend at the right edge for *X position* or top edge for *Y position* in the **Viewer** window. To change either value, highlight the existing value and type a new value or drag the  to increase or decrease the value.

Width

The *Width (points)* displays the width of the legend. For a vertical legend, the *Width (points)* is the distance from the left to the right of the legend. For a horizontal legend, the *Width (points)* is the distance from the top to the bottom of the legend. Values range between zero and 200 points. To change the width, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.



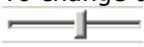
This Horizontal legend has a Width (points) set to a value of 20.

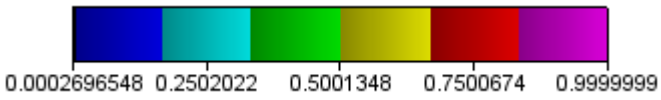


This Horizontal legend has a Width (points) set to a value of 40.

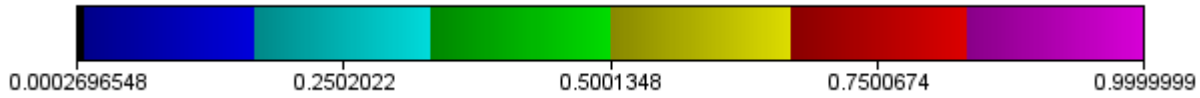
Length

The *Length (points)* displays the length of the legend. For a vertical legend, the *Length (points)* is the distance from the top to the bottom of the legend. For a horizontal legend, the *Length (points)* is the distance from the left to the right of the legend. Values range between zero and 1024 points.

To change the length, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.



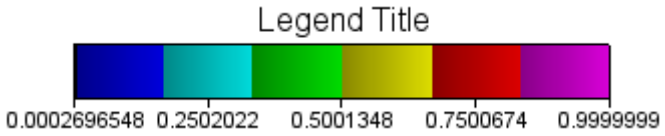
This Horizontal legend has a Length (points) set to a value of 200.



This Horizontal legend has a Length (points) set to a value of 400.

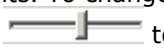
Title

Click the ⊞ next to the *Title* section to set the legend title properties. Enter an optional title to display above the legend. To enter the *Title*, click in the empty spot next to *Title*. Type the desired text. If text already appears next to *Title*, highlight this text and type new text.



This legend has a title.

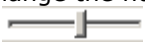
Title Size

The *Size (points)* property is the height of the title text, in points. The size can range from 4 to 72 points. To change the height, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.

Labels

Click the ⊞ next to *Labels* to open the labels section for the legend. Set the number of labels, label height, label format, or set custom labels in this section.

Number of Labels

The *Number of labels* displays the number of text labels displayed next to the legend in the **Viewer** window. A legend can have zero labels up to 100 labels. To change the number of labels, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.

Label Size

The *Size (points)* of the label text is displayed in points. To change the label size, highlight the existing value and type a new value or click and drag the  to increase or decrease the value. The height can range between 4 and 72 points.

Use Custom Labels

Check the box next to *Use custom labels* to set user defined text next to the label. When this box is unchecked, automatic labels are created.

Custom Labels

When the box next to *Use custom labels* is checked, custom labels can be defined. Only labels in the proper format will be displayed. The format of the label string is "value:string." Use quotes if the label has spaces. Separate multiple labels with a comma or space. For example, the label string *0:Low* displays the text *Low* at the value 0. The label string *0:Low, 1.5: Medium, 3:0: High* would display the labels *Low*, *Medium*, and *High* at data values of 0.0, 1.5, and 3.0, respectively. To enter new custom labels, click in the empty box next to *Custom labels* and type the desired *value:label* text.

Label Format

Click the ⊞ next to *Label Format* to open the label format section and set the label format.

Font

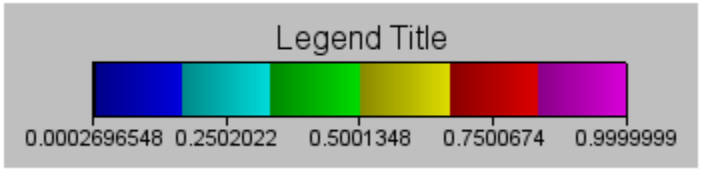
Click the ⊞ next to *Font* to open the legend font section. The *Font* displays the font to use to display the labels. To change the font, click on the existing font name and select the desired font from the list. All of the fonts installed on your computer are displayed in the list.

Antialias Text

Check the box next to *Antialias text* to make the font look smoother by slightly adjusting the colors along curves. For a crisper text appearance, uncheck the box.

Background Rectangle

Check the box next to *Background rectangle* to draw a rectangle around the legend. The rectangle is filled with the *Background color*. Uncheck the box to remove the background rectangle.



This legend has a gray background rectangle.

Foreground Color

The *Foreground color* displays the color to use as an outline for the legend and as the label color. To change the color, click on the current color and select the desired color in the color palette. Click *Other* to open the **Colors** dialog and specify a custom color.

Background Color

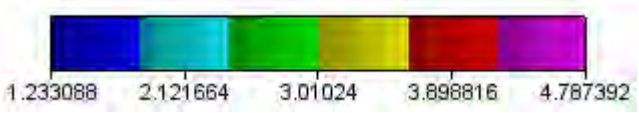
The *Background color* displays the color to use to fill the background rectangle. To change the color, click on the current color and select the desired color in the color palette. Click *Other* to open the **Colors** dialog and specify a custom color.

How can I change the data min and data max in the legend?

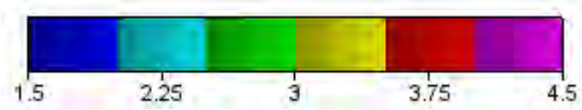
To change the data minimum and data maximum values that are displayed on a legend, use the **Colormap Editor** dialog.

1. Select a module that has a legend in the **Network Manager** to display its properties in the **Property Manager**.

2. In the **Property Manager**, click on the ellipsis button to the right of the *Colormap* item.
3. In the **Colormap Editor** dialog, specify a new *Data Min* and/or *Data Max* value. Click the *Close* button. The data values on the legend are updated.



This legend has default data min and max values.



This legend has custom data min and max values.

Chapter 9 - Contours

Contours Module

The **Network | Graphics Output | Contours** command adds a *Contours* module.

The *Contours* module generates contour lines for a two-dimensional data set or for slices of a three-dimensional data set. Contour lines represent the boundary between data less than a given level (threshold) and data greater than the level. For three-dimensional data sets, the *Contours* module creates a planar slice through the lattice and contours the two-dimensional slice. Contour lines are colored by mapping data values to colors through a Colormap.

Contours are calculated by locating points along with the lattice lines where the values are equal to the contour level by using linear interpolation between the lattice nodes. The contour line is drawn by connecting the points with straight line segments. The more dense a lattice is, the more line segments are generated and the smoother the contour lines appear. This method is sometimes called "marching squares."

Contours can be draped over the *HeightField* by loading the contours from a grid file as a curvilinear lattice.

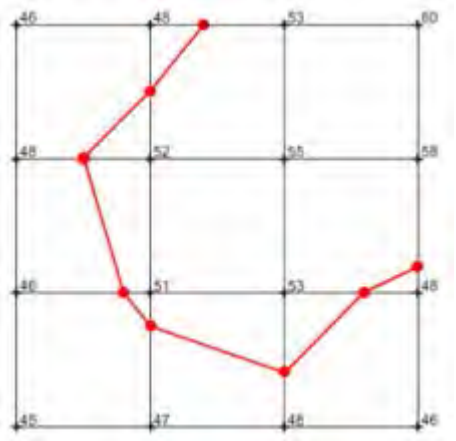
A *Contours* module can be exported to different file types with the **File | Save Data** command, including IV, 3D DXF, and XYZC data files in the following data file formats: CSV, DAT, SLK, TXT, XLS, and XLSX.

Inputs

Lattice is the input type for the *Contours* module.

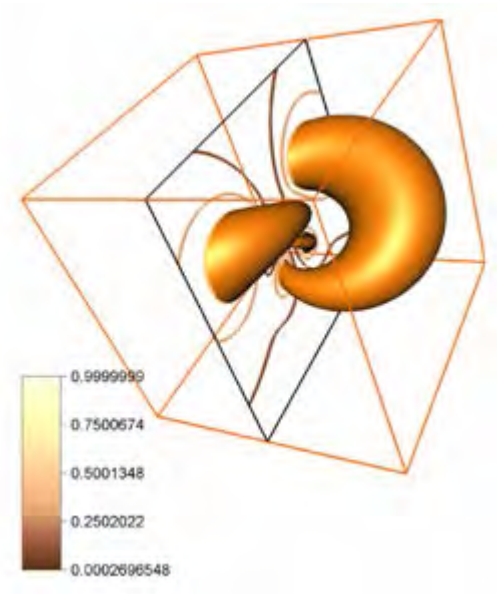
Outputs

The *Contours* module creates an output geometry. It may be connected to the *Graphics Output Modules*. An *Info Module* may also be connected to the output node.



A contour with a value of 50 (shown in red) is drawn with straight line segments between points along lattice lines (shown in black) connecting lattice nodes.

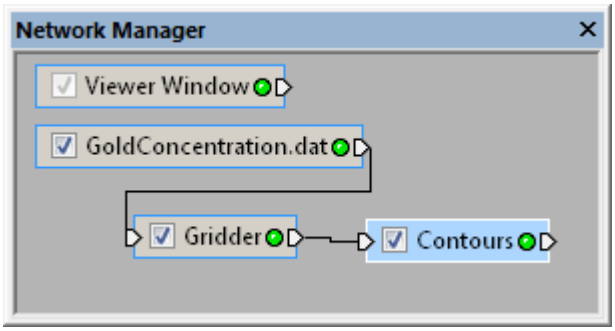
Red dots illustrate the ends of the straight line segments that form the contour.



The Contours module contours a slice created through a three dimensional data set.

Properties

The *Contours* module properties are described below.



*Select the Contours module in the **Network Manager** to display its properties in the **Property Manager**.*

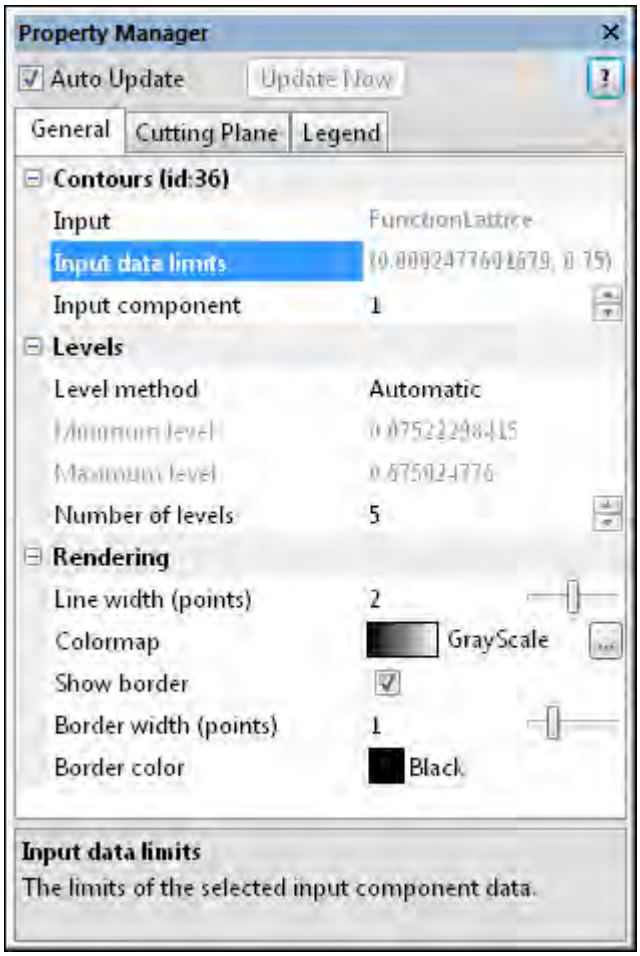
The *Contours* module contains the following tabs in the **Property Manager:**

General

Cutting Plane

Legend

General Options



Customize the Contours properties.

Input

The *Input* property shows the source to which the module is connected. This option cannot be changed in the **Property Manager**, but can be changed in the **Network Manager** by changing the module input.

Input Data Limits

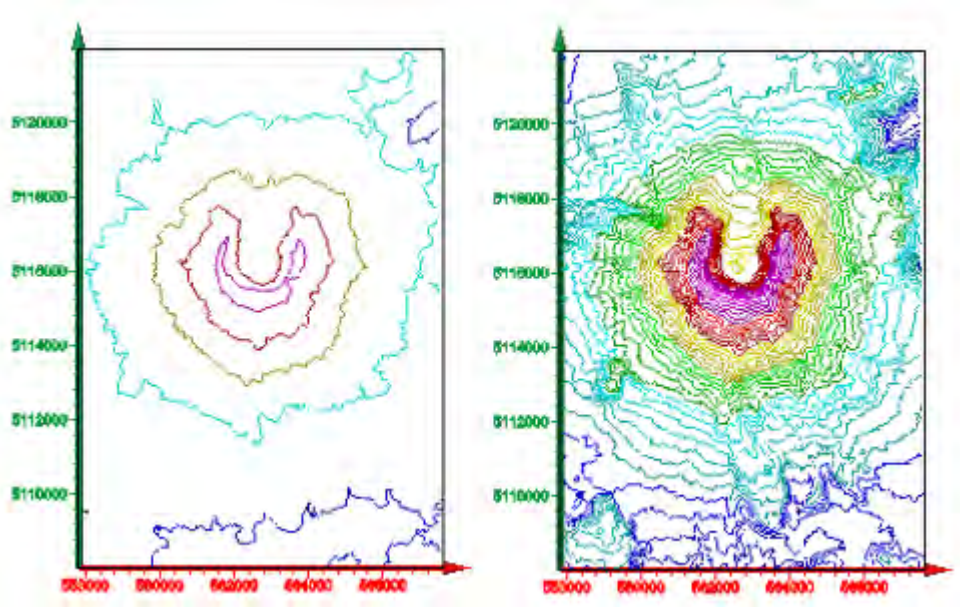
The *Input data limits* option displays the minimum and maximum value of the selected input component data. If the input data is a three-dimensional lattice, these values are for the entire lattice and not just for the specified two-dimensional cutting plane through the lattice.

Input Component

The *Input component* specifies which of the input components contains the data used to compute the contours when there is more than one component. When there is not more than one component in the input lattice, this option is not shown. To change the *Input component*, highlight the existing value and type a new value or click the  to increase or decrease the component value.

Level Method

The *Level method* specifies the method to compute the values of the contour levels. Increase the number of contour levels to provide more detail. To change the level method, click on the existing option and select the new value from the list. Available options are *Automatic; Min, max, count; Min, max, interval;* and *Explicit*.



Add more contour levels to define additional details. The default Automatic method (shown on the left) displays five levels. The Min, max, interval method (shown on the right) is used to display contours at 50-meter intervals.

Automatic uses the data limits to display the specified number of contour levels. The minimum and maximum levels are offset from the data minimum and maximum by an amount equal to the data range divided by twice the number of levels. When attaching a new *Contours* module to a lattice, the default number of levels is 5, so the offset is 10% of the data range. When attaching a *Contours* module to a lattice, the offset depends on the number of levels specified in the *Contours* module. If the minimum or maximum level values are not present in the specified cutting plane, not all of the specified number of contours are displayed.

For example, the data minimum and maximum for the lattice in helens2.grd are 684 and 2547 and the range is 1863. The default = 186.3. The minimum level is 684 + 186.3 = 870.3. The maximum level is 2547 - 186.3 = 2360.7. To calculate the default contour interval, divide the levels range by the number of levels minus 1, or (2360.7 - 870.3)/4 = 372.6. The resulting value for the five levels are 870.3, 1242.9, 1615.5, 1988.1, and 2360.7.

Min, max, count displays the specified number of contour levels with equal spacing from the specified minimum to the specified maximum values. Use the calculations in the example for the *Automatic* method to calculate the contour interval and values of the contour levels. If the *Minimum level* or *Maximum level* values are not present in the specified cutting plane, not all of the specified number of levels are displayed.

Min, max, interval displays contour levels with the specified interval ranging from specified minimum and maximum. If the *Minimum level* or *Maximum level* values are not present in the specified cutting plane, not all of the specified number of levels are displayed.

Explicit allows you to enter a space-separated list of contour levels to be displayed. If a level is outside the data range, it is not displayed.

Minimum Level

The *Minimum level* is the smallest data value to use for displaying a contour line. If the specified *Minimum level* value is smaller than the minimum value in the lattice slice, no contour is displayed for that level. If the specified value is larger than the maximum value in the lattice slice, no contours are displayed for any of the levels. This option is available for each *Level method*, except for the *Explicit* option. To change the *Minimum level* value, highlight the existing value and type a new value. The option is for information only when the *Level method* is set to *Automatic*.

Maximum Level

The *Maximum level* is the largest data value to use for displaying a contour line. If the specified value is larger than the maximum value in the lattice slice, no contour is displayed for that level. If the specified value is smaller than the minimum value in the lattice slice, none of the levels are displayed. This option is available for each *Level method*, except for the *Explicit* option. To change the *Minimum level* value, highlight the existing value and type a new value. The option is for information only when the *Level method* is set to *Automatic*.

Number of Levels

The *Number of levels* is the number of contour levels. To change the number of levels, highlight the existing value and type a new value or click the  to increase or decrease the number of levels. If the minimum and maximum level values are not present in the specified cutting plane, there are fewer contours displayed than the number specified by the *Number of levels*. This option is available when the *Level method* is set to *Automatic* or *Min, max, count.*

Level Interval

The *Level interval* is the contour interval or the difference between the values of two adjacent contours. To change the level interval, highlight the existing value and type a new value. If the *Level interval* is larger than the range from *Minimum level* to *Maximum level,* no contour lines are shown. This option is only available when the *Level method* is set to *Min, max, interval*.

Level Values

The *Level values* option controls the exact values that should display contour lines. If no values are typed next to *Levels values*, no lines are displayed on the contour plot. To set the *Level values*, click in the box next to *Level values*, and type a number. For multiple values, include a space between contour levels. For example, to display contours with a logarithmic interval, enter the following:

0.1 0.5 1 5 10 50 100 500 1000 5000

Line Width

The *Line width (points)* is the thickness of the contour lines (in points). To change the line width, highlight the existing value and type a new value or drag the  to increase or decrease the line width. A line width of zero is equal to one pixel. The *Line width (points)* values vary between zero and 4.

Colormap

The *Colormap* maps scalar values to colors. To change the colormap used by the contours, click the existing color bar to the right of the *Colormap* command to select a different colormap from the list. Click on the desired colormap and the contour map updates. Alternatively, click the  button to open the **Colormap Editor** dialog. The **Colormap Editor** dialog allows you to create a custom colormap and to change the mapping of color to data values. By default, the minimum and maximum values are mapped to the data minimum and maximum, and a smaller range may not be present in the cutting plane.

Draw Border

Check the box next to the *Show border* command to show or hide the border around the edge of the contour plane.

Border Width

The *Border width (points)* controls the thickness of the border (in points). To change the line width, highlight the existing value and type a new value or drag the  to increase or decrease the line width. A line width of zero is equal to one pixel. The *Border width (points)* values vary between zero and 4.

Border Color

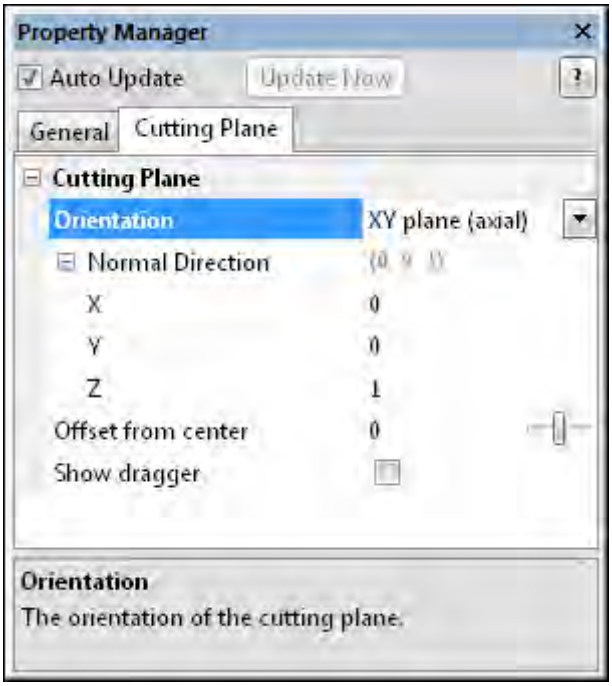
The *Border color* option controls the color of the border line. Click the color next to *Border color* to open the color palette. Click on the desired new color to update the map. Alternatively, click the *Other* button at the bottom of the list to open the **Colors** dialog, where you can select a custom color.

Contours - Cutting Plane Page

The *Cutting Plane,* a property of some modules, determines the orientation ("normal") of the cutting plane. You can choose one of the preset local axis plane normal, or you can choose an arbitrary direction by entering values for X, Y, and Z. The plane normal may be specified numerically with the *Normal Direction* property or graphically when the dragger is shown.

Properties

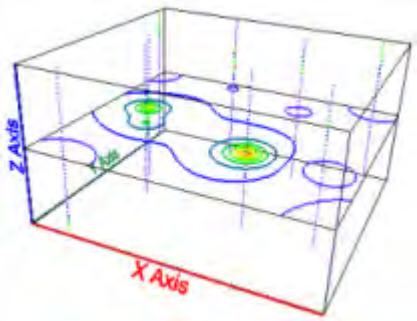
The *Cutting Plane* properties are described below. This feature may be accessed via the *Contours*, *ObliqueImage*, and *Slice* modules.



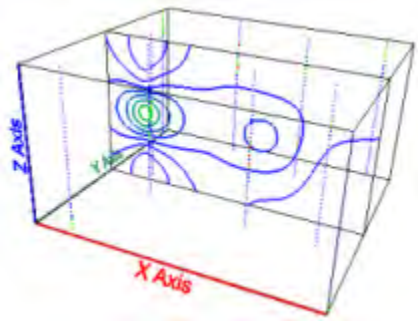
The Cutting Plane properties may be changed to affect the orientation ("normal") of the cutting plane.

Orientation

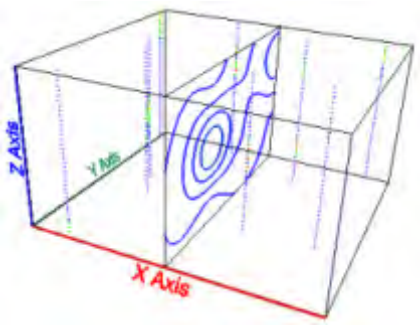
To change the *Orientation*, click on the existing option and select the desired option. Choose one of the preset local axis plane normals— *XY plane (axial)*, *XZ plane (coronal)*, or *YZ plane (sagittal)*— or choose *Custom* to enter custom *Normal Direction* values. *Axial* indicates the XY plane that travels horizontally. *Coronal* indicates the XZ plane that travels vertically. *Sagittal* indicates the YZ plane that travels vertically. The *Custom* setting lets you type a value under *Normal Direction* for each axis.



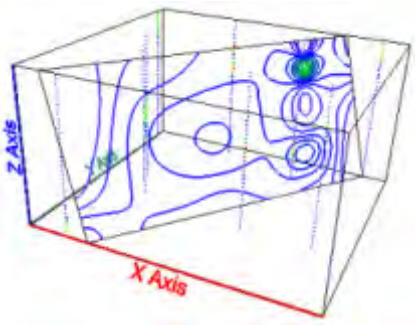
The contours Orientation is set to XY plane (axial).



The contours Orientation is set to XZ plane (coronal).



The contours Orientation is set to YZ plane (sagittal).



The contours Orientation is set to Custom.

Normal Direction

Enter a value under *Normal Direction* for the *X*, *Y*, and *Z* components of the plane normal. Changing these values changes the cutting plane orientation with respect to the normal for that axis. Changing the *X* value when the *Y* and *Z* values are zero creates a YZ cutting plane. Setting each of the *X*, *Y*, and *Z* values to 1 produces a plane with an oblique orientation. This vector is the normal to the cutting plane.

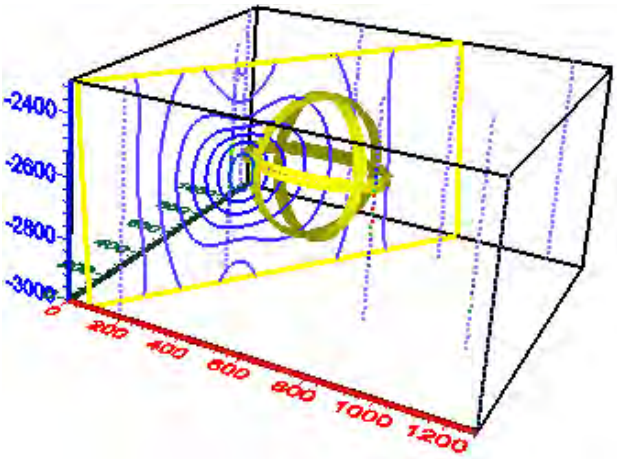
Offset from Center

The *Offset from center* property measures the distance of the cutting plane from the center of the lattice. To change the offset, highlight the existing value and type a new value or click and drag the  to increase or decrease the distance from the center. Units are in axis units. If the X axis goes from 0 to 60, a value of 0 for the *Offset from center* will place the plane at X= 30. A value of just less than 30 will place the plane at the maximum X value. A value of just greater than -30 will place the plane at the minimum X value.

Show Dragger

The *Dragger* allows interactive positioning and rotation of the plane.

Check the *Show dragger* box to show the dragger— a virtual, rotatable trackball— and allow interactive positioning and rotation of the plane. The dragger allows the orientation and offset of the cutting plane to be specified.



The Dragger is shown above in yellow and can be used to interactively rotate the cutting plane.

Drag one of the three bands to rotate around a principal axis in the direction of the ring. Drag anywhere on the ball (between the rings) to perform an unconstrained rotation in any direction.

To specify a user-defined rotation axis, press the SHIFT key while clicking the left mouse button and dragging. A new distinctively-colored axis is added.

To scale the size of the trackball, press the CTRL key and drag the trackball.

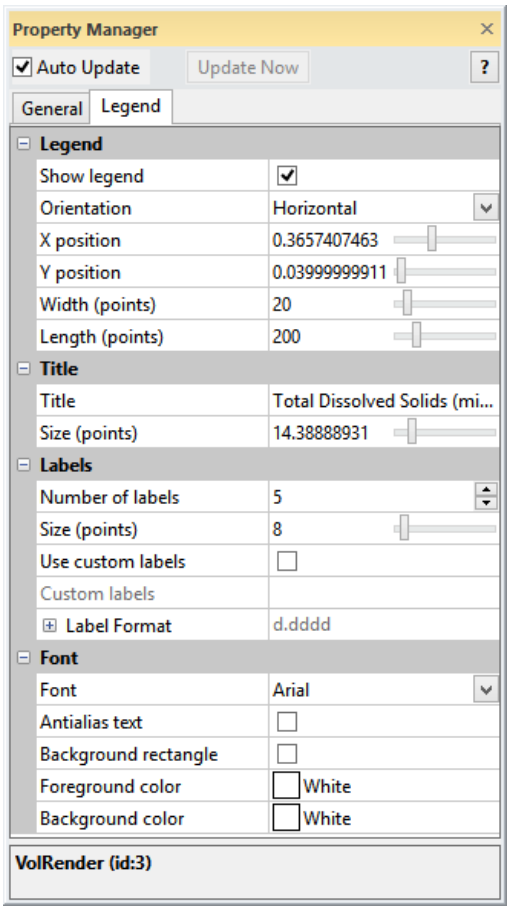
To offset the plane in the perpendicular direction, drag the cutting plane itself.

Contours - Legend Page

A legend is a key to a plot that labels data by color, pattern, or symbol. Legends can be added to many modules, including *FaceRender*, *HeightField*, *Isosurface*, *ScatterPlot*, *StreamLines*, *VectorPlot*, *VolRender*, and *WellRender*.

Properties

Select a module that supports legends. In the **Property Manager**, click the *Show* button next to *Legend* to display the legend in the **Viewer** window. The legend properties are described below.



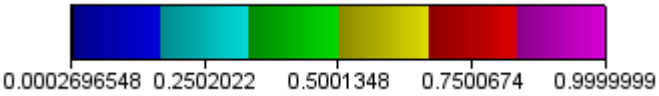
*Customize the Legend properties in the **Property Manager**.*

Show Legend

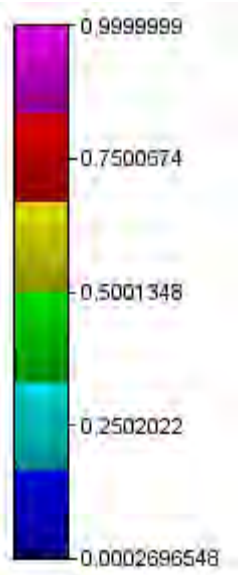
Check the box next to the *Show legend* option to display the legend for the selected module.

Orientation

Orientation specifies the direction of the "length" dimension. Available options are *Horizontal* and *Vertical* length dimension. To change the *Orientation*, click on the existing option and select the desired option from the list.

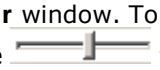


This legend has a Horizontal Orientation.

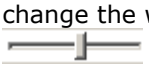


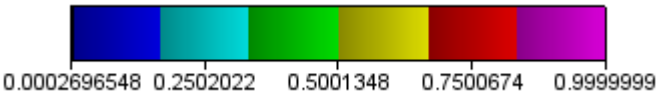
This legend has a Vertical Orientation.

X and Y Position

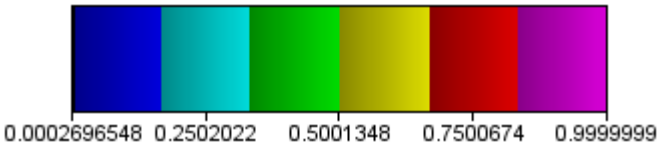
The *X position* and *Y position* control the location of the legend in the **Viewer** window. Adjust the X and/or Y position value to move the legend location. Values range from 0.0 to 1.0. 0.0 places the left bottom edge of the legend at the left edge for *X position* or the bottom edge for *Y position* in the **Viewer** window. 1.0 places the left bottom edge of the legend at the right edge for *X position* or top edge for *Y position* in the **Viewer** window. To change either value, highlight the existing value and type a new value or drag the  to increase or decrease the value.

Width

The *Width (points)* displays the width of the legend. For a vertical legend, the *Width (points)* is the distance from the left to the right of the legend. For a horizontal legend, the *Width (points)* is the distance from the top to the bottom of the legend. Values range between zero and 200 points. To change the width, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.

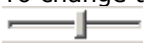


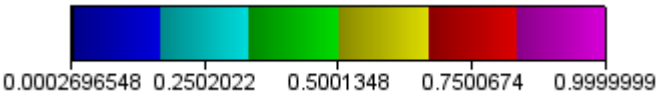
This Horizontal legend has a Width (points) set to a value of 20.



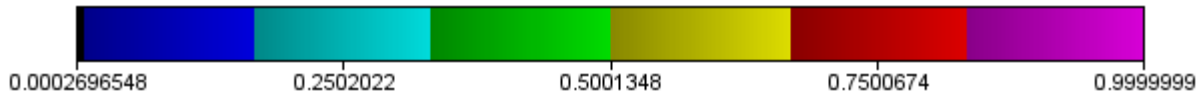
This Horizontal legend has a Width (points) set to a value of 40.

Length

The *Length (points)* displays the length of the length of the legend. For a vertical legend, the *Length (points)* is the distance from the top to the bottom of the legend. For a horizontal legend, the *Length (points)* is the distance from the left to the right of the legend. Values range between zero and 1024 points. To change the length, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.



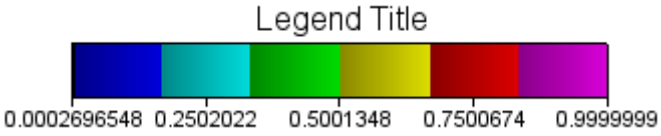
This Horizontal legend has a Length (points) set to a value of 200.



This Horizontal legend has a Length (points) set to a value of 400.

Title

Click the ⊞ next to the *Title* section to set the legend title properties. Enter an optional title to display above the legend. To enter the *Title*, click in the empty spot next to *Title*. Type the desired text. If text already appears next to *Title*, highlight this text and type new text.



This legend has a title.

Title Size

The *Size (points)* property is the height of the title text, in points. The size can range from 4 to 72 points. To change the height, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.

Labels

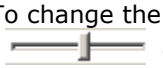
Click the ⊞ next to *Labels* to open the labels section for the legend. Set the number of labels, label height, label format, or set custom labels in this section.

Number of Labels

The *Number of labels* displays the number of text labels displayed next to the legend in the **Viewer** window. A legend can have zero labels up to 100 labels. To change the number of labels, highlight

the existing value and type a new value or click and drag the  to increase or decrease the value.

Label Size

The *Size (points)* of the label text is displayed in points. To change the label size, highlight the existing value and type a new value or click and drag the  to increase or decrease the value. The height can range between 4 and 72 points.

Use Custom Labels

Check the box next to *Use custom labels* to set user defined text next to the label. When this box is unchecked, automatic labels are created.

Custom Labels

When the box next to *Use custom labels* is checked, custom labels can be defined. Only labels in the proper format will be displayed. The format of the label string is "value:string." Use quotes if the label has spaces. Separate multiple labels with a comma or space. For example, the label string *0:Low* displays the text *Low* at the value 0. The label string *0:Low, 1.5: Medium, 3:0: High* would display the labels *Low*, *Medium*, and *High* at data values of 0.0, 1.5, and 3.0, respectively. To enter new custom labels, click in the empty box next to *Custom labels* and type the desired *value:label* text.

Label Format

Click the ⊞ next to *Label Format* to open the label format section and set the label format.

Font

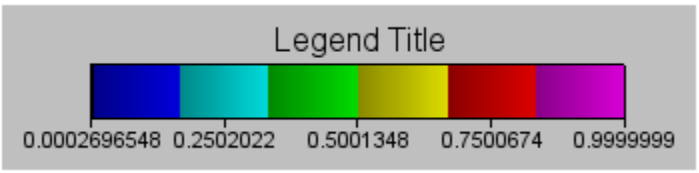
Click the ⊞ next to *Font* to open the legend font section. The *Font* displays the font to use to display the labels. To change the font, click on the existing font name and select the desired font from the list. All of the fonts installed on your computer are displayed in the list.

Antialias Text

Check the box next to *Antialias text* to make the font look smoother by slightly adjusting the colors along curves. For a crisper text appearance, uncheck the box.

Background Rectangle

Check the box next to *Background rectangle* to draw a rectangle around the legend. The rectangle is filled with the *Background color*. Uncheck the box to remove the background rectangle.



This legend has a gray background rectangle.

Foreground Color

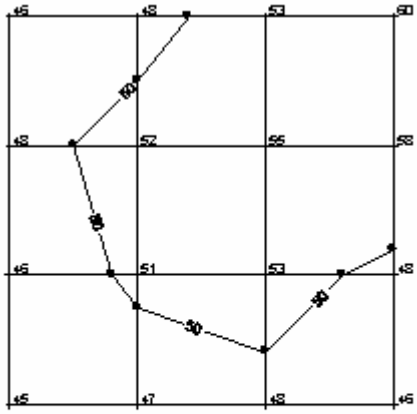
The *Foreground color* displays the color to use as an outline for the legend and as the label color. To change the color, click on the current color and select the desired color in the color palette. Click *Other* to open the **Colors** dialog and specify a custom color.

Background Color

The *Background color* displays the color to use to fill the background rectangle. To change the color, click on the current color and select the desired color in the color palette. Click *Other* to open the **Colors** dialog and specify a custom color.

Drawing Contours

Use the *Contours module* to create a contour map. When **Voxler** creates a contour map, the contour lines are drawn as a series of straight line segments between adjacent grid lines in the grid or lattice slice. The point where a contour line intersects a grid line is determined by interpolation between values at adjacent grid nodes.

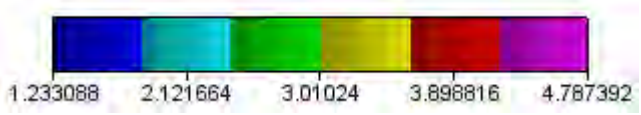


The path of a contour line through a grid is defined by interpolating between grid nodes. This example shows the straight-line segments that define the contour line. The dots are added to show the ends of each line segment.

How can I change the data min and data max in the legend?

To change the data minimum and data maximum values that are displayed on a legend, use the **Colormap Editor** dialog.

1. Select a module that has a legend in the **Network Manager** to display its properties in the **Property Manager**.
2. In the **Property Manager**, click on the ellipsis button to the right of the *Colormap* item.
3. In the **Colormap Editor** dialog, specify a new *Data Min* and/or *Data Max* value. Click the *Close* button. The data values on the legend are updated.



This legend has default data min and max values.



This legend has custom data min and max values.

Chapter 10 - Isosurfaces

Isosurface Module

The **Network | Graphics Output | Isosurface** command adds a *Isosurface* module.

The *Isosurface* module creates an isosurface through an input lattice. An isosurface is a surface of constant value in a three-dimensional volume. The isosurface value is set in the *Isovalue* property in the **Property Manager**. The isosurface separates regions less than the selected isovalue from regions greater than the selected isovalue. All points on the isosurface have the same value (the isovalue).

This module provides a very quick method for reconstructing polygonal surface models from a lattice. The algorithm computes lattice cell interactions and combines them into triangle meshes for rendering.

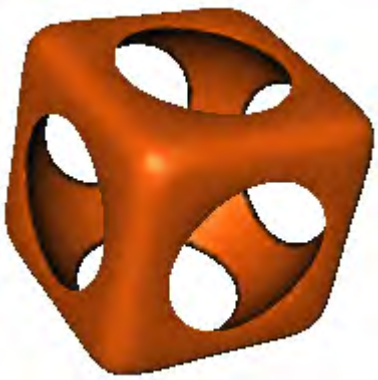
An *Isosurface* module can be exported to different file types with the **File | Save Data** command, including IV, 3D DXF, and XYZC data files in the following data file formats: CSV, DAT, SLK, TXT, XLS, and XLSX. Note the component value is the same for every point in the isosurface, and is the *Isovalue* specified in the **Property Manager**.

Inputs

3D lattice is the input type for the *Isosurface* module.

Outputs

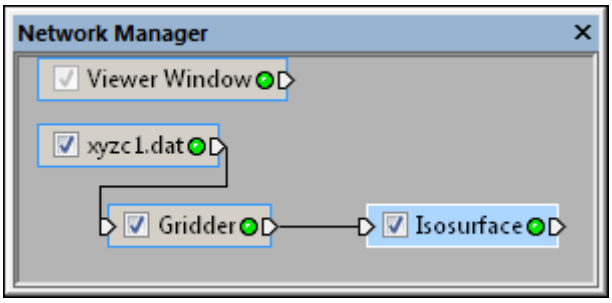
The *Isosurface* module creates an output geometry. It may be connected to the *Graphics Output Modules*. An *Info Module* may also be connected to the output node.



The sample TestLattice wiffle ball with an Isosurface module.

Properties

The *Isosurface* module properties are described below.



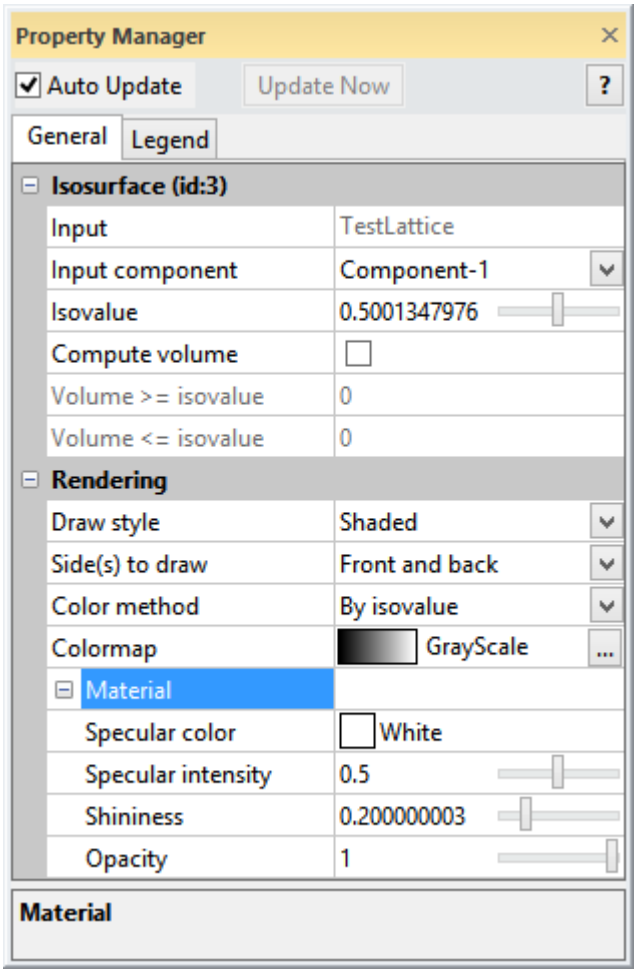
*Select the Isosurface module in the **Network Manager** to display its properties in the **Property Manager**.*

The *Isosurface* module contains the following tabs in the **Property Manager:**

General

Legend

General Options



Customize the Isosurface properties.

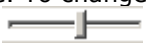
Input

The *Input* property shows the source to which the module is connected. This option cannot be changed in the **Property Manager**, but can be changed in the **Network Manager** by changing the module input.

Input Component

The *Input component* specifies which of the input components contains the data used to compute the isosurface. To change the *Input component*, click the current selection and select the desired component from the list.

Isovalue

The threshold value of the surface to generate. The generated surface separates data less than this value from data greater than this value. To change the *Isovalue*, highlight the existing value and type a new value or click and drag the  to increase or decrease the value. The image in the **Viewer** window updates as the value changes.

Volume

Check the box next to *Compute volume* to calculate the volume displayed by all enclosed areas of the *Isosurface*. The volume greater than the isosurface is displayed next to the *Volume >= isovalue* option. Only areas with isosurface values greater than the isovalue are included in this calculation. Areas with values less than the isovalue are shown in the *Volume<=isovalue* option.

Volume calculations are generated from all voxels either partially or fully included in the isosurface. A voxel is fully contained if all of the eight corner lattice points have a component value greater than or equal to the *Isovalue*. A voxel is partially contained if one or more corner lattice points have a component value greater than or equal to the *Isovalue*. The total *Volume >= isovalue* is the sum of the individual volumes from these voxels. Results are provided in cubic units based on the units of the input grid file. To make volumetric sense, the X, Y, and Z units should be the same.

To increase accuracy in the volume calculations, increase the lattice resolution by regridding the data or increase the number of cells in the X, Y, and Z directions (Nx, Ny, and Nz) or *Gridder*.

Leaving the *Compute volume* box checked can result in an increase in the time needed to redraw the *Isosurface*. It is recommend that the box next to *Compute volume* be left unchecked unless the volume calculation is needed.

Draw Style

The *Draw style* is the style that is used to draw the isosurface in the **Viewer** window. Available options are *Shaded*, *Lines,* and *Points*. To change the value, click on the existing option and select the desired option from the list. Choose *Shaded* to render all polygonal geometry as shaded surfaces. Choose *Lines* to render all geometry as a wireframe connection of border lines. Choose *Points* to render all geometry as vertex points.

Side to Draw

the *Side(s) to Draw* controls which portion of the isosurface is visible. Available options are *Front and back*, *Front only*, and *Back only*. To change the value, click on the existing option and select the desired option from the list. Choose *Front and back* to draw both the front and back of the isosurface. When *Draw style* is set to *Lines* or *Points*, the resulting image may be difficult to determine which values are in the front or back., *Front only*, and *Back only* draws only a single side

of the isosurface. The side drawn is relative to the camera's current position. The closer side is the *Front* and the side furthest from the camera is *Back*.

Color Method

The method used to generate the surface colors. The color of the isosurface may be set to a *Fixed* color or to *By isovalue*, which matches a colormap with the *Isovalue*. Mapping the *Isovalue* through a Colormap can be useful when displaying multiple isosurfaces. To change the *Color method*, click on the existing option and select the desired option from the list.

When set to *Fixed*, the *Color* option becomes available. The *Color* is the color of the isosurface. To change the color, click on the existing color and select the desired color from the list. Alternatively, click the *Other* box to open the **Colors** dialog, where you can set custom colors.

When the *Color method* is set to *By isovalue*, the *Colormap* option becomes available. To change the colormap used by the isosurface, click the existing color bar to the right of the *Colormap* command to select a different colormap from the list. Click on the desired colormap and the isosurface updates. Alternatively, click the  button to open the **Colormap Editor** dialog. The **Colormap Editor** dialog allows you to create a custom colormap and to change the mapping of color to data values.

Material

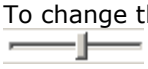
Click the  next to *Material* to open the material options settings. The *Material* options control the color, intensity, shininess, and opacity of specular highlights, which are bright spots of light that appear on illuminated objects.

Specular Color

Specular color and *Specular intensity* control the amount of directional light reflected off of the surface. "Specular" refers to light coming from a particular direction and bouncing off of the surface in a preferred direction. A shiny surface such as metal has a high specular component whereas a surface like carpet has almost no specular component. Increasing the specular intensity results in strong shadow effects and more pronounced "shiny" or glare spots.

Specular color details the color of specular highlights. This helps control the amount of directional light reflected off the surface. To change the color, click on the existing color and select the desired color from the list. Alternatively, click the *Other* box to open the **Colors** dialog, where you can set custom colors. White is the default *Specular color*.

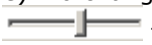
Specular Intensity

Specular intensity is the intensity of specular highlights. This also helps control the amount of directional light reflected off the surface. The value ranges from 0.0 (none) to 1.0 (full intensity). To change the value, highlight the existing value and type a new value or click and drag the  to increase or decrease the light intensity.

Shininess

Shininess controls the size and brightness of specular highlights. The larger the value, the smaller and brighter (more focused) the highlight. The value ranges from 0.0 (dull, diffuse surface) to 1.0 (highly polished surface). To change the shininess, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.

Opacity

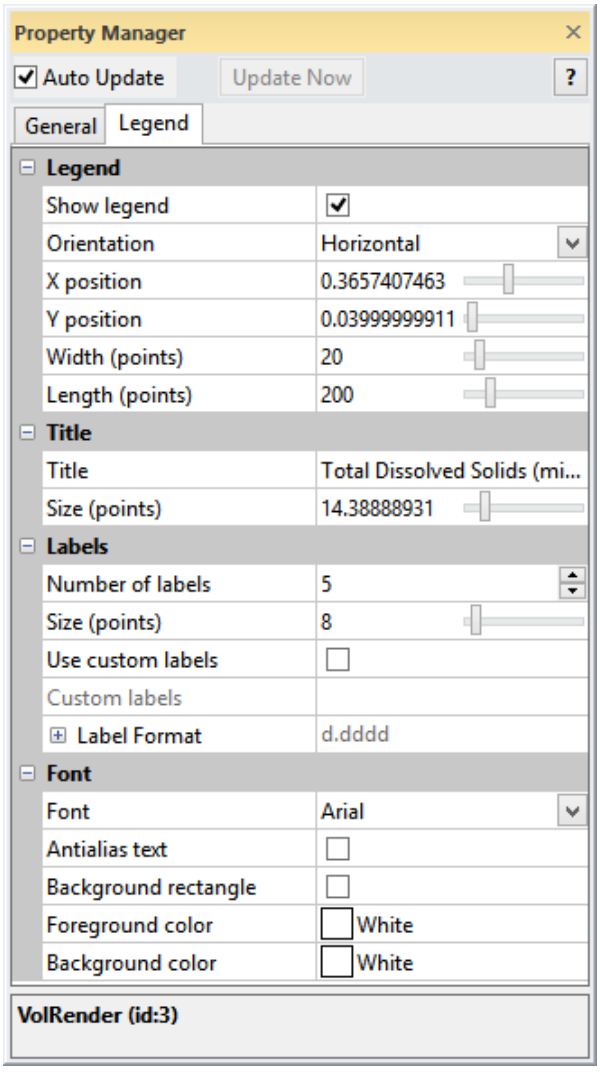
The opacity of the surface is a combined function of the opacity settings in the *Colormap* and the material *Opacity* value. The *Opacity* value combines with the alpha value in the *Colormap* to determine the transparency of the surface. Values range from 0.0 (completely transparent) to 1.0 (completely opaque). To change the opacity, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.

Isosurface - Legend Page

A legend is a key to a plot that labels data by color, pattern, or symbol. Legends can be added to many modules, including *FaceRender*, *HeightField*, *Isosurface*, *ScatterPlot*, *StreamLines*, *VectorPlot*, *VolRender*, and *WellRender*.

Properties

Select a module that supports legends. In the **Property Manager**, click the *Show* button next to *Legend* to display the legend in the **Viewer** window. The legend properties are described below.



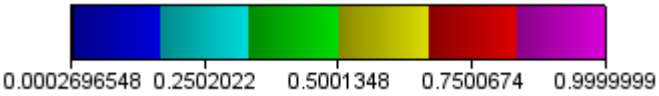
*Customize the Legend properties in the **Property Manager**.*

Show Legend

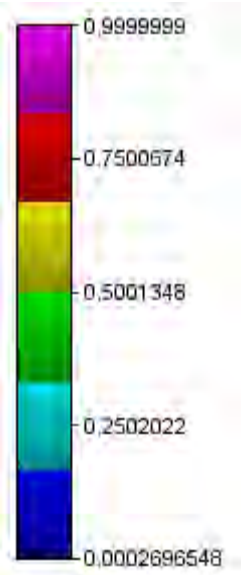
Check the box next to the *Show legend* option to display the legend for the selected module.

Orientation

Orientation specifies the direction of the "length" dimension. Available options are *Horizontal* and *Vertical* length dimension. To change the *Orientation*, click on the existing option and select the desired option from the list.

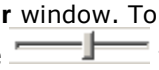


This legend has a Horizontal Orientation.

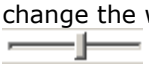


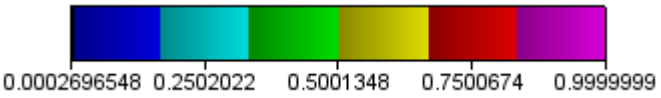
This legend has a Vertical Orientation.

X and Y Position

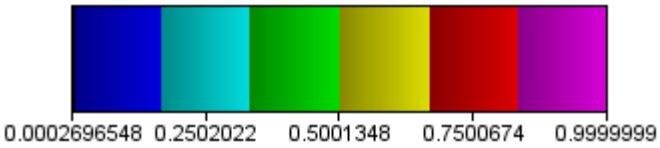
The *X position* and *Y position* control the location of the legend in the **Viewer** window. Adjust the X and/or Y position value to move the legend location. Values range from 0.0 to 1.0. 0.0 places the left bottom edge of the legend at the left edge for *X position* or the bottom edge for *Y position* in the **Viewer** window. 1.0 places the left bottom edge of the legend at the right edge for *X position* or top edge for *Y position* in the **Viewer** window. To change either value, highlight the existing value and type a new value or drag the  to increase or decrease the value.

Width

The *Width (points)* displays the width of the legend. For a vertical legend, the *Width (points)* is the distance from the left to the right of the legend. For a horizontal legend, the *Width (points)* is the distance from the top to the bottom of the legend. Values range between zero and 200 points. To change the width, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.



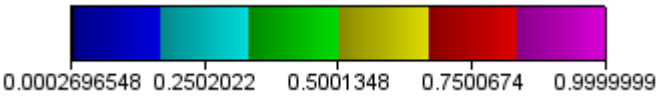
This Horizontal legend has a Width (points) set to a value of 20.



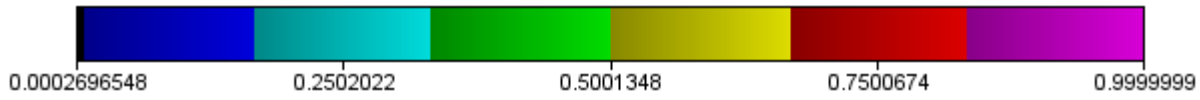
This Horizontal legend has a Width (points) set to a value of 40.

Length

The *Length (points)* displays the length of the length of the legend. For a vertical legend, the *Length (points)* is the distance from the top to the bottom of the legend. For a horizontal legend, the *Length (points)* is the distance from the left to the right of the legend. Values range between zero and 1024 points. To change the length, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.



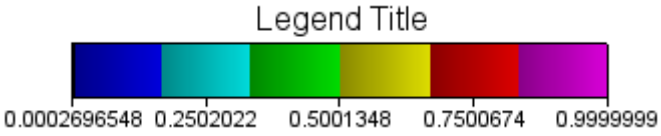
This Horizontal legend has a Length (points) set to a value of 200.



This Horizontal legend has a Length (points) set to a value of 400.

Title

Click the ⊞ next to the *Title* section to set the legend title properties. Enter an optional title to display above the legend. To enter the *Title*, click in the empty spot next to *Title*. Type the desired text. If text already appears next to *Title*, highlight this text and type new text.



This legend has a title.

Title Size

The *Size (points)* property is the height of the title text, in points. The size can range from 4 to 72 points. To change the height, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.

Labels

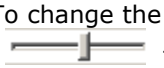
Click the ⊞ next to *Labels* to open the labels section for the legend. Set the number of labels, label height, label format, or set custom labels in this section.

Number of Labels

The *Number of labels* displays the number of text labels displayed next to the legend in the **Viewer** window. A legend can have zero labels up to 100 labels. To change the number of labels, highlight

the existing value and type a new value or click and drag the  to increase or decrease the value.

Label Size

The *Size (points)* of the label text is displayed in points. To change the label size, highlight the existing value and type a new value or click and drag the  to increase or decrease the value. The height can range between 4 and 72 points.

Use Custom Labels

Check the box next to *Use custom labels* to set user defined text next to the label. When this box is unchecked, automatic labels are created.

Custom Labels

When the box next to *Use custom labels* is checked, custom labels can be defined. Only labels in the proper format will be displayed. The format of the label string is "value:string." Use quotes if the label has spaces. Separate multiple labels with a comma or space. For example, the label string *0:Low* displays the text *Low* at the value 0. The label string *0:Low, 1.5: Medium, 3:0: High* would display the labels *Low*, *Medium*, and *High* at data values of 0.0, 1.5, and 3.0, respectively. To enter new custom labels, click in the empty box next to *Custom labels* and type the desired *value:label* text.

Label Format

Click the ⊞ next to *Label Format* to open the label format section and set the label format.

Font

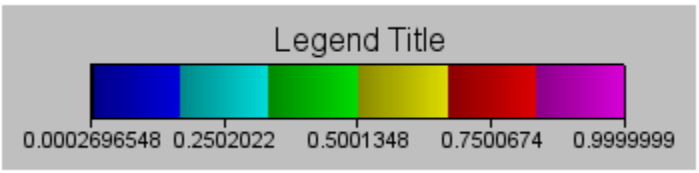
Click the ⊞ next to *Font* to open the legend font section. The *Font* displays the font to use to display the labels. To change the font, click on the existing font name and select the desired font from the list. All of the fonts installed on your computer are displayed in the list.

Antialias Text

Check the box next to *Antialias text* to make the font look smoother by slightly adjusting the colors along curves. For a crisper text appearance, uncheck the box.

Background Rectangle

Check the box next to *Background rectangle* to draw a rectangle around the legend. The rectangle is filled with the *Background color*. Uncheck the box to remove the background rectangle.



This legend has a gray background rectangle.

Foreground Color

The *Foreground color* displays the color to use as an outline for the legend and as the label color. To change the color, click on the current color and select the desired color in the color palette. Click *Other* to open the **Colors** dialog and specify a custom color.

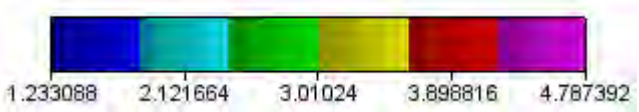
Background Color

The *Background color* displays the color to use to fill the background rectangle. To change the color, click on the current color and select the desired color in the color palette. Click *Other* to open the **Colors** dialog and specify a custom color.

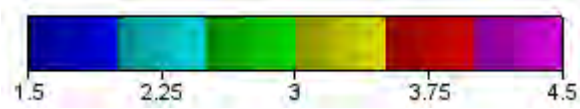
How can I change the data min and data max in the legend?

To change the data minimum and data maximum values that are displayed on a legend, use the **Colormap Editor** dialog.

1. Select a module that has a legend in the **Network Manager** to display its properties in the **Property Manager**.
2. In the **Property Manager**, click on the ellipsis button to the right of the *Colormap* item.
3. In the **Colormap Editor** dialog, specify a new *Data Min* and/or *Data Max* value. Click the *Close* button. The data values on the legend are updated.



This legend has default data min and max values.



This legend has custom data min and max values.

How can I calculate volume?

Voxler can calculate the volume of an *Isosurface* or *FaceRender* module. To display volumes, data should be in the same units in the X, Y, and Z directions. For instance, all directions should contain meters or feet. Using latitude and longitude or mixing units is not recommended as the volume will not have meaning.

To display the volume:

1. Attach an *Isosurface* or *FaceRender* module to a *Gridder* or lattice.
2. In the **Property Manager**, click on the **General** tab.
3. Check the box next to the *Compute volume* option.
4. The *Approx. volume* or the *Volume >= isovalue* and *Volume <=isovalue* is displayed in the **Property Manager**. This is in X, Y, Z units.

Alternatively, export lattice slices to the **Surfer** .GRD file format.

In **Voxler**, save data as .GRD:

1. Select a uniform lattice module in the **Network Manager** and choose the **File | Save Data** command to export a uniform lattice to the *GRD Surfer Grid format*. The exported lattice has a "Z" dimension of one. This means the lattice is two-dimensional. If an input lattice with a Z dimension of greater than one is exported, then **Voxler** prompts you to save multiple slices into separate .GRD files.
2. If a multi-slice uniform lattice is output to a .GRD file format, it is output as a series of slices in the Z direction. The **Select Slices** dialog appears and the user is prompted for the range of slices to output and a file name template that adds the slice number to each generated file name.

In **Surfer**, calculate volume:

1. Choose the **Grid | Volume** command to calculate the area above a certain level, and multiply it by the thickness of the slice to get the volume.
2. Repeat for each slice.

How do I create an Isosurface with different colors above and below a GRD surface?

The color of an *Isosurface* is controlled solely by its value.

One method to produce different colors is to use two *Isosurface* modules and a *Math* module to modify the value of the *Isosurface* lattice based on the Z value in the GRD file of the surface.

If the lattice controlling the *Isosurface* is input "A" and the GRD of the geologic surface is input "B", then the following equations change the value of the lattice to the negative value based on the geologic surface GRD.

IF Z > B, A, -A

Or

Z > B? A:-A

How can I model geologic surfaces?

One way to create a geological model in **Voxler** is to fill the interval between the top and bottom of each layer with a numerical value corresponding to the layer. Attach a *Gridder* module to interpolate data throughout the rectangular volume, and attach an *Isosurface* module for each layer, specifying the value that is the just below each layer value. Attach a *VolRender* module to fill the layers with color. Attach a *ScatterPlot* module to show the borehole points.

Example

For example, consider the following borehole data.

Borehole 1 at 456000 easting and 4123000 northing

Ground Elevation and top of alluvium: 5280

Depth to base of alluvium and top of Dakota formation: 280

Depth to base of Dakota and top of Morrison formation: 780

Total Depth of borehole: 1000

First convert the depths to elevations.

Ground Elevation and topo of alluvium: 5280

Top of Dakota: 5000

Top of Morrison: 4500

Total Depth: 4280

Fill the intervals with numeric values for each formation. For this example, use 1 for alluvium, 2 for the Dakota, and 3 for the Morrison.

456000, 4123000, 5280, 1
456000, 4123000, 5200, 1
456000, 4123000, 5100, 1
456000, 4123000, 5000, 2
456000, 4123000, 4900, 2
456000, 4123000, 4800, 2
456000, 4123000, 4700, 2
456000, 4123000, 4600, 2
456000, 4123000, 4500, 3
456000, 4123000, 4400, 3
456000, 4123000, 4300, 3
456000, 4123000, 4280, 3

Repeat for the other boreholes in the area and save all the points in a single data file. Load the data file, attach a *Gridder* module and grid the data. You may need to apply gridding and search anisotropy if the X, Y range is much larger than the Z range. Attach an *Isosurface* module and specify a value of 1.99 to show the top of the Dakota. Attach another *Isosurface* module and specify a value of 2.99 to show the top of the Morrison. Attach a *VolRender* module to fill the layers with color.

In this example a depth interval of 100 is used. You may wish to use a smaller interval if the features you wish to depict are smaller. When you grid the data specify a smaller Z interval for more grid lines in the Z direction and better resolution.

How do I use interval data for formation tops or ore grade?

One way to create a geological model in **Voxler** is to fill the interval between the top and bottom of each layer with a numerical value corresponding to the layer. Attach a *Gridder* module to interpolate data throughout the rectangular volume, and attach an *Isosurface* module for each layer, specifying the a value that is the just below each layer value. Attach a *VolRender* module to fill the layers with color. Attach a *ScatterPlot* module to view the borehole points in 3D.

Example

For example, consider the following borehole data.

Borehole 1 at 456000 easting and 4123000 northing

Ground Elevation and top of alluvium: 5280

Depth to base of alluvium and top of Dakota formation: 280

Depth to base of Dakota and top of Morrison formation: 780

Total Depth of borehole: 1000

First convert the depths to elevations.

Ground Elevation and top of alluvium: 5280

Top of Dakota: 5000

Top of Morrison: 4500

Total Depth: 4280

Fill the intervals with numeric values for each formation. Use 1 for alluvium, 2 for the Dakota, and 3 for the Morrison.

456000, 4123000, 5280, 1
456000, 4123000, 5200, 1
456000, 4123000, 5100, 1
456000, 4123000, 5000, 2
456000, 4123000, 4900, 2
456000, 4123000, 4800, 2
456000, 4123000, 4700, 2
456000, 4123000, 4600, 2
456000, 4123000, 4500, 3
456000, 4123000, 4400, 3
456000, 4123000, 4300, 3
456000, 4123000, 4280, 3

Repeat for the other boreholes in the area and save all the points in a single data file. Load the data file, attach a *Gridder* module and grid the data. You may need to apply gridding and search anisotropy if the X, Y range is much larger than the Z range. Attach an *Isosurface* module and specify a value of 1.99 to show the top of the Dakota. Attach another *Isosurface* module and specify a value of 2.99 to show the top of the Morrison. Attach a *VolRender* module to fill the layers with color.

This example used a depth interval of 100. You may wish to use a smaller interval if the features you wish to depict are smaller. When you grid the data specify a smaller Z interval for more grid lines in the Z direction and better resolution.

Chapter 11 - Heightfields

HeightField Module

The **Network | Graphics Output | HeightField** command adds a *HeightField* module.

The *HeightField* module displays a lattice slice in three dimensions. The slice is scaled in the direction perpendicular to the lattice slice by the value of the data component specified in the lattice data import module and the *HeightField* scale factor. The surface is colored by mapping the data values through a Colormap, which is controlled in the **Colormap Editor**. The *HeightField* module will ignore the **Surfer** blanking value in the lattice (1.70141e+038). Blanked regions are not rendered.

Images can be draped over the HeightField by loading an image and connecting it as an *Input Image Overlay*. Vector files can be draped over the *HeightField* by importing the vector file and connecting it as an *Input Vector Overlay*. Contours can be draped over the *HeightField* by loading the contours from a grid file as a curvilinear lattice or as a vector overlay.

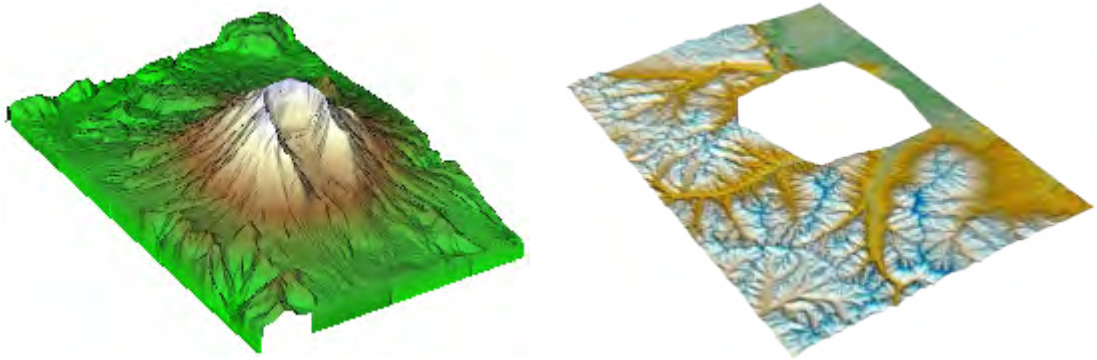
A *HeightField* module can be exported to different file types with the **File | Save Data** command, including IV, 3D DXF, and XYZC data files in the following data file formats: CSV, DAT, SLK, TXT, XLS, and XLSX.

Inputs

Lattice and image are the input type for the *HeightField* module.

Outputs

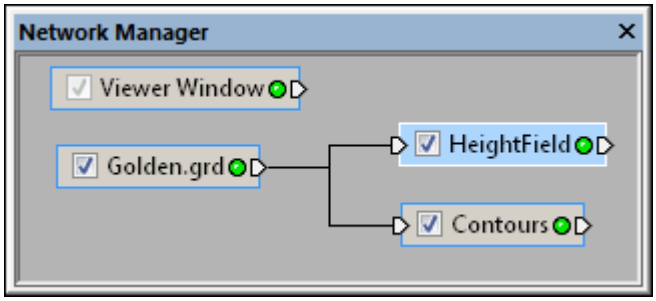
The *HeightField* module creates an output geometry. It may be connected to the *Graphics Output Modules*. An *Info Module* may also be connected to the output node.



The image on the left shows a HeightField module with a sample data set and uses the Land Colormap. The image on the right shows a HeightField module with a data set that includes a blanked region.

Properties

The *HeightField* module properties are described below, followed by an example that illustrates how to use the module.



*Select the HeightField module in the **Network Manager** to display its properties in the **Property Manager**.*
*The Contours module contains the following tabs in the **Property Manager:***

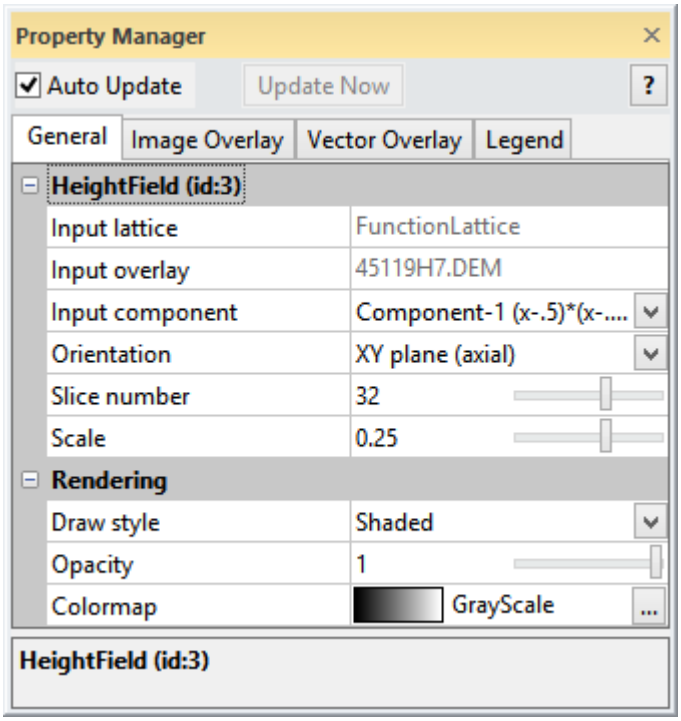
General

Image Overlay

Vector Overlay

Legend

General Options



Customize the HeightField properties.

Input

The *Input lattice* property shows the source to which the module is connected. The *Input lattice* is the geometry source and defines the heights.

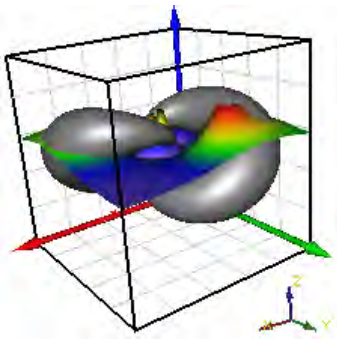
The *Input overlay* is the image source that defines the image overlay. Only 2D RGBA images can be connected to the *Input overlay*.

Input Component

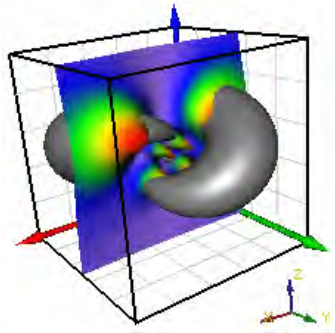
The *Input component* specifies which of the input components contains the data used to compute the *HeightField*. To change the *Input component*, click the current component select the desired component from the list.

Orientation

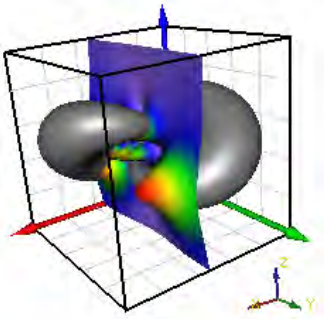
Choose the direction of the *HeightField* slice through the volume. Choices include *XY plane (axial)*, *XZ plane (coronal)*, or *YZ Plane (sagittal)*. To change the *Orientation*, click on the existing option and select the desired option. Choose one of the preset local axis plane normals. *Axial* indicates the XY plane that travels horizontally. *Coronal* indicates the XZ plane that travels vertically. *Sagittal* indicates the YZ plane that travels vertically.



The HeightField Orientation is set to XY plane (axial).



The HeightField Orientation is set to XZ plane (coronal).



The HeightField Orientation is set to YZ plane (sagittal).

Slice Number

The *Slice number* is the index of the slice to display. The *Slice number* ranges from 1 to the maximum number of nodes in the specified direction. The slice number resets to the middle of the volume when the *Orientation* is changed. It is possible to specify an orientation so that the slice is exactly on edge, in which case nothing is drawn. Rotate the camera slightly to see the image. This option is not available if a single slice exists. To change the slice displayed, highlight the existing value and type a new value or click and drag the  to increase or decrease the slice value.

Scale

Scale represents the scale factor when computing the heights (0.0 = no height). By default, the scale factor is set to 0.25 for a uniform lattice and 0.0 for a curvilinear lattice. To change the scale, highlight the existing value and type a new value or click and drag the  to the desired scale factor. The scale is any value between -1 and +1.

When importing a curvilinear lattice, it is recommended that the *Scale* be set to 0.0. In most cases, **Surfer** .GRD files will be imported into **Voxler** as curvilinear lattices. Curvilinear lattices have the **Surfer** *Z* value imported as both the Z and C values in **Voxler**. This results in excessive stretching when changing the *Scale* from zero.

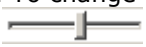
With uniform lattices, the more the scale factor deviates from 0.0, the farther the surface projects in the perpendicular direction. The scale factor is normalized by a percentage of the volume diagonal. The scale is computed using the following equation:

scale = ScaleFromPropertiesWindow * HalfTheLatticeDiagonal / (DataMax - DataMin)

Draw Style

The *Draw style* is the method used to display the HeightField. Available options are *Shaded*, *Lines*, and *Points*. To change the style, click on the existing option and select the desired option from the list. *Shaded* renders all polygonal geometry as shaded surfaces. *Lines* renders all geometry as border lines. Lines are drawn between each lattice node and those surrounding it. *Points* renders all geometry as vertex points. Each lattice node is displayed as a small square.

Opacity

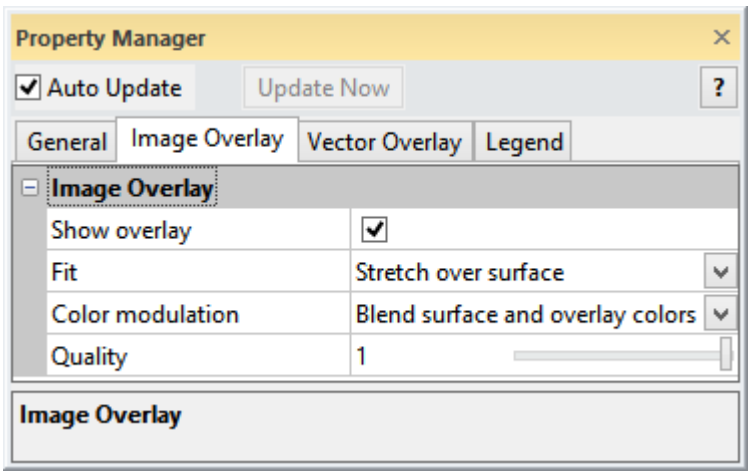
Choose the level of opacity. This value ranges from 0.0 for completely transparent images to 1.0 for completely opaque images. To change the opacity, highlight the existing value and type a new value or click and drag the  to the desired value. See *Transparency Type* for more information about transparency.

Colormap

The *HeightField* is colored by mapping the data values using the specified *Colormap*. To change the colors, click on the existing colormap and select the desired colormap from the list. Or, click on the  to set custom colormap options. The Colormap is not displayed if an RGBA color lattice is used for the *Input lattice* since the colors are obtained directly from the lattice.

HeightField - Image Overlay Page

The *HeightField* module **Image Overlay** page displays options for overlaying an image onto a *HeightField*. To open the **Image Overlay** page, click on the *HeightField* module in the **Network Manager**. In the **Property Manager**, click on the **Image Overlay** tab.



*The **Image Overlay** options allow images to be stretched or clipped to a HeightField.*

Overlay

The *Overlay* options are available when an image is connected as an *Input Image Overlay*. The image file name is shown in the *Input overlay* on the **General** page.

Show Overlay

Check the box next to the *Show overlay* command to show or hide the overlay display on the *HeightField*.

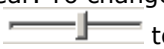
Overlay Fit

The *Fit* option controls how the overlay is placed on the *HeightField*. Available options are *Stretch over surface* and *Clip to surface*. To change how the overlay fits on the *HeightField*, click on the existing option and select the desired option from the list. *Stretch over surface* ignores the X, Y, and Z coordinates for the image and extends the image to the extents of the *HeightField*. *Clip to surface* uses the X, Y, and Z coordinates from the image and overlays the image onto the *HeightField* at the correct coordinates. If the image extends beyond the *HeightField*, the image is clipped at the extents of the *HeightField*.

Overlay Color Modulation

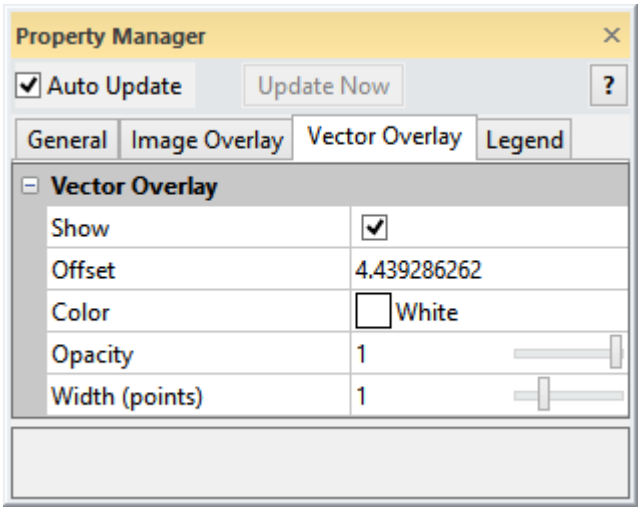
The *Color modulation* option refers to the method used to combine the HeightField colormap with the image colors. Available options are *Blend surface and overlay colors* and *Use overlay color only*. To change how the overlay and HeightField are combined, click on the existing option and select the desired option from the list. When *Blend surface and overlay colors* is selected, the HeightField colormap and the image color are combined to create a new color that is a mix of both. When *Use overlay color only* is selected, the HeightField colormap does not appear. The color is defined by only the image overlay.

Overlay Quality

The *Quality* option determines how detailed the image appears as the overlay on the *HeightField*. Values range from zero to one. When set to zero, the image does not appear. As the value gets larger, the image quality increases with 1 being the best quality. As the value increases, the redraw time slows. With large images, reducing the *Quality* will result in faster redraw, though a less accurate image on the HeightField will appear. To change the *Quality*, highlight the existing value and type a new value or click and drag the  to increase or decrease the quality.

HeightField - Vector Overlay Page

The *HeightField* module **Vector Overlay** page displays options for overlaying an vector file onto a *HeightField*. To open the **Vector Overlay** page, click on the *HeightField* module in the **Network Manager**. In the **Property Manager**, click on the **Vector Overlay** tab.



*The **Vector Overlay** options allow vector files to be overlaid on a HeightField.*

Overlay

The *Vector Overlay* options are available when an vector file is connected as the *Input Vector Overlay*.

Show Overlay

Check the box next to the *Show overlay* command to show or hide the overlay display on the *HeightField*.

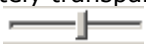
Offset

The *Offset* value offsets the vector lines above the *HeightField* surface. Specify the *Offset* value by highlighting the current value and typing a new *Offset* value into the box.

Color

The *Color* property specifies the color of the vector lines. To change the color, click the current color next to *Color* and select a new color from the color palette. Click the *Other* button in the color palette to select or create a color in the **Colors** dialog.

Opacity

The *Opacity* value controls the vector line transparency. The *Opacity* value is any number from 0 to 1, where 1 is completely opaque and 0 is completely transparent. Highlight the current *Opacity* value and type a new value or click and drag the  to change the *Opacity*.

Width

The *Width (points)* property specifies the vector line width in points. The line with can be from 0 to 4 points. A *Width (points)* value of 0 renders the line as one pixel wide. Highlight the current *Width*

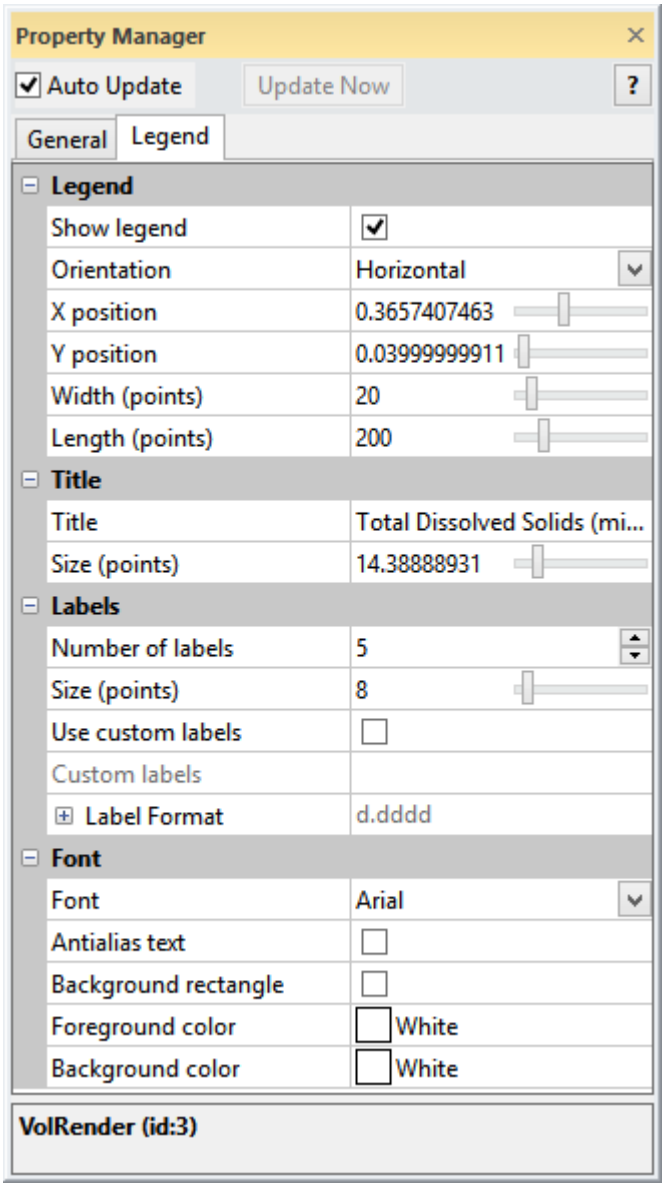
(points) value and type a new value in points or click and drag the  to change the line *Width*.

HeightField - Legend Page

A legend is a key to a plot that labels data by color, pattern, or symbol. Legends can be added to many modules, including *FaceRender*, *HeightField*, *Isosurface*, *ScatterPlot*, *StreamLines*, *VectorPlot*, *VolRender*, and *WellRender*.

Properties

Select a module that supports legends. In the **Property Manager**, click the *Show* button next to *Legend* to display the legend in the **Viewer** window. The legend properties are described below.



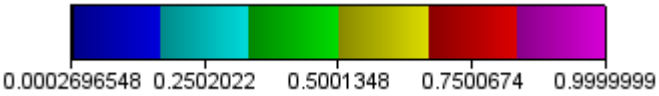
*Customize the Legend properties in the **Property Manager**.*

Show Legend

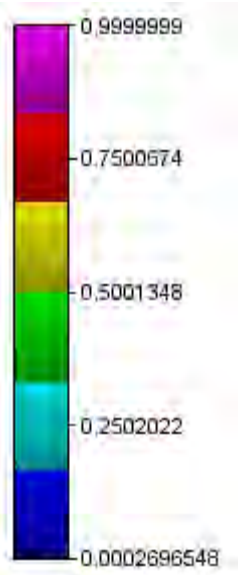
Check the box next to the *Show legend* option to display the legend for the selected module.

Orientation

Orientation specifies the direction of the "length" dimension. Available options are *Horizontal* and *Vertical* length dimension. To change the *Orientation*, click on the existing option and select the desired option from the list.

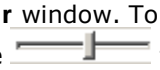


This legend has a Horizontal Orientation.

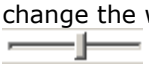


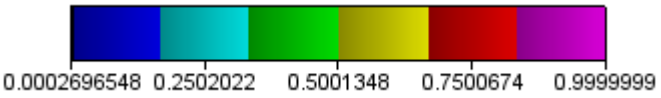
This legend has a Vertical Orientation.

X and Y Position

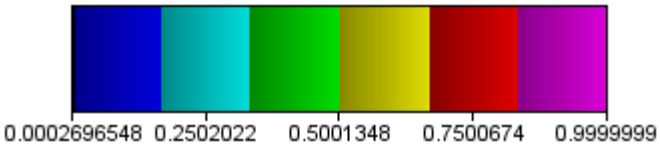
The *X position* and *Y position* control the location of the legend in the **Viewer** window. Adjust the X and/or Y position value to move the legend location. Values range from 0.0 to 1.0. 0.0 places the left bottom edge of the legend at the left edge for *X position* or the bottom edge for *Y position* in the **Viewer** window. 1.0 places the left bottom edge of the legend at the right edge for *X position* or top edge for *Y position* in the **Viewer** window. To change either value, highlight the existing value and type a new value or drag the  to increase or decrease the value.

Width

The *Width (points)* displays the width of the legend. For a vertical legend, the *Width (points)* is the distance from the left to the right of the legend. For a horizontal legend, the *Width (points)* is the distance from the top to the bottom of the legend. Values range between zero and 200 points. To change the width, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.



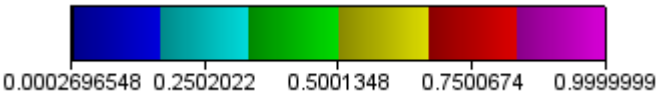
This Horizontal legend has a Width (points) set to a value of 20.



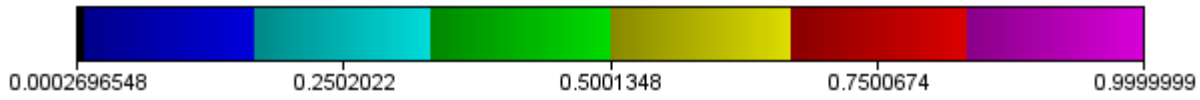
This Horizontal legend has a Width (points) set to a value of 40.

Length

The *Length (points)* displays the length of the legend. For a vertical legend, the *Length (points)* is the distance from the top to the bottom of the legend. For a horizontal legend, the *Length (points)* is the distance from the left to the right of the legend. Values range between zero and 1024 points. To change the length, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.



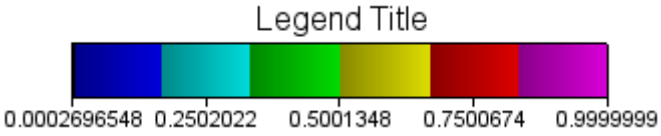
This Horizontal legend has a Length (points) set to a value of 200.



This Horizontal legend has a Length (points) set to a value of 400.

Title

Click the ⊞ next to the *Title* section to set the legend title properties. Enter an optional title to display above the legend. To enter the *Title*, click in the empty spot next to *Title*. Type the desired text. If text already appears next to *Title*, highlight this text and type new text.



This legend has a title.

Title Size

The *Size (points)* property is the height of the title text, in points. The size can range from 4 to 72 points. To change the height, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.

Labels

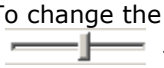
Click the ⊞ next to *Labels* to open the labels section for the legend. Set the number of labels, label height, label format, or set custom labels in this section.

Number of Labels

The *Number of labels* displays the number of text labels displayed next to the legend in the **Viewer** window. A legend can have zero labels up to 100 labels. To change the number of labels, highlight

the existing value and type a new value or click and drag the  to increase or decrease the value.

Label Size

The *Size (points)* of the label text is displayed in points. To change the label size, highlight the existing value and type a new value or click and drag the  to increase or decrease the value. The height can range between 4 and 72 points.

Use Custom Labels

Check the box next to *Use custom labels* to set user defined text next to the label. When this box is unchecked, automatic labels are created.

Custom Labels

When the box next to *Use custom labels* is checked, custom labels can be defined. Only labels in the proper format will be displayed. The format of the label string is "value:string." Use quotes if the label has spaces. Separate multiple labels with a comma or space. For example, the label string *0:Low* displays the text *Low* at the value 0. The label string *0:Low, 1.5: Medium, 3:0: High* would display the labels *Low*, *Medium*, and *High* at data values of 0.0, 1.5, and 3.0, respectively. To enter new custom labels, click in the empty box next to *Custom labels* and type the desired *value:label* text.

Label Format

Click the ⊞ next to *Label Format* to open the label format section and set the label format.

Font

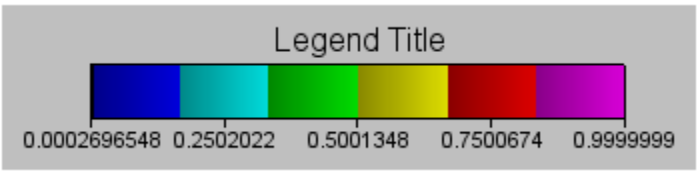
Click the ⊞ next to *Font* to open the legend font section. The *Font* displays the font to use to display the labels. To change the font, click on the existing font name and select the desired font from the list. All of the fonts installed on your computer are displayed in the list.

Antialias Text

Check the box next to *Antialias text* to make the font look smoother by slightly adjusting the colors along curves. For a crisper text appearance, uncheck the box.

Background Rectangle

Check the box next to *Background rectangle* to draw a rectangle around the legend. The rectangle is filled with the *Background color*. Uncheck the box to remove the background rectangle.



This legend has a gray background rectangle.

Foreground Color

The *Foreground color* displays the color to use as an outline for the legend and as the label color. To change the color, click on the current color and select the desired color in the color palette. Click *Other* to open the **Colors** dialog and specify a custom color.

Background Color

The *Background color* displays the color to use to fill the background rectangle. To change the color, click on the current color and select the desired color in the color palette. Click *Other* to open the **Colors** dialog and specify a custom color.

HeightField Examples

The below are a few ways that *HeightFields* can be used to display data.

Display Contours on HeightField

1. Load a .DEM or .GRD file with elevations or bathymetry by clicking the **File | Import** command.
2. In the **Lattice Import Options** dialog, select *Import as curvilinear lattice* and click OK. **Voxler** loads the .DEM file elevation values in the curvilinear lattice as both the *Component 1* and Z values.
3. Attach a *HeightField* module to the output connection pad of the module by clicking the **Network | Graphics Output | HeightField** command.
4. Attach a *Contours* module to the output connection pad of the grid module by clicking the **Network | Graphics Output | Contours** command.

The contours are automatically placed at the correct position on the *HeightField*.

Display Overlay on HeightField

1. Load a .DEM or .GRD file with elevations or bathymetry by clicking the **File | Import** command.
2. In the **Lattice Import Options** dialog, select *Import as curvilinear lattice* and click *OK*. **Voxler** loads the .DEM file elevation values in the curvilinear lattice as both the *Component 1* and Z values.
3. Attach a *HeightField* module to the output connection pad of the module by clicking the **Network | Graphics Output | HeightField** command.
4. Load an image to be displayed on the *HeightField* by clicking the **File | Import** command.
5. Connect the output port from the image to the *Input Overlay* port on the HeightField. Select *Connect Input Overlay*.
6. Click on the *HeightField* module to select it.
7. Click on the **Overlay** tab in the **Property Manager**.
8. Check the box next to the *Show overlay* option.
9. Set the *Fit* and *Color modulation*, as needed.

The overlay and *HeightField* colormap are combined and displayed as a composite color.

Load a Uniform Lattice at the Correct Z Height

Follow the steps below to create a *HeightField* from a .DEM file with the *HeightField* scale equal to the Z axis scale.

1. Load a .DEM or .GRD file with elevations or bathymetry by clicking the **File | Import** command.
2. In the **Lattice Import Options** dialog, select *Import as uniform lattice (default)* and click OK. **Voxler** loads the .DEM file elevation values in the uniform lattice as *Component 1* and assigns zeros to all the Z values in the lattice.
3. With the input data module (.DEM file) selected, write down the *Component 1 Limits* and the coordinate limits for *X min*, *X max*, *Y min*, and *Y max* from the **Property Manager**.
4. Attach a *Transform* module to the input data module by clicking the **Network | Computational | Transform** command.
5. Click on the *Transform* module to select it.
6. Click on the **Transform** tab.

7. Open the *Translation* section and change the *Z* value to the lower *Component 1 Limits* value recorded in step 3 above.
8. Attach a *HeightField* module to the output connection pad of the *Transform* module by clicking the **Network | Graphics Output | HeightField** command.
9. The *HeightField* base location is moved from 0 to the proper minimum Z value. The *Scale* calculation is as follows:

scale = (zMax - zMin) / (0.5 * sqrt ((xMax - xMin)^2 + (yMax - yMin)^2))

The resulting *HeightField* matches the Z axis scale.

How do I create a surface from my XYZ data file?

To create a surface from an XYZ data file, follow these steps:

1. Create a column of 0's in the data file.
2. Open **Voxler** and choose the **File | Import** command.
3. Select the data file and click the *Open* button.
4. In the **Data Import Options** dialog, set the appropriate settings for your data and click the *OK* button.
5. In the **Property Manager**, specify the *X Coordinate* and *Y Coordinate* columns, select the column of 0's as the *Z Coordinate*, and select the Z data column as the *Single Component*. Click the *OK* button.
6. Right-click the data module in the **Network** window and select **Computational | Gridder** to create a *Gridder* module.
7. Select the *Gridder* module in the **Network Manager**. In the **Properties** window, set any settings you wish and click the *Begin Gridding* button.
8. Once the gridding is complete, right-click the *Gridder* module and select **Graphics Output | HeightField**.

Alternatively, if you own **Surfer** you can grid the XYZ data in **Surfer** (using the **Grid | Data** command), load the Surfer grid file into **Voxler** (using the **File | Import** command), and attach a *HeightField* module directly to the grid file.

I attach a HeightField to a 2D curvilinear lattice, and when I change the scale from 0, the surface is distorted. Why?

This is a result of using a 2D curvilinear lattice with a *HeightField module*. Keep the *Scale* at 0 for this type of lattice.

A *HeightField* offsets the surface in the direction perpendicular to the surface by an amount proportional to the scale factor times component value. This means higher elevations will be distorted more. The higher elevations at the peaks extend outwards more than the lower regions and ultimately start to intersect with each other. Also, the distortion direction is perpendicular to the curvilinear lattice, so the offset will be in a more horizontal direction than vertical along the sides of the hills.

See *Tutorial: Heightfields Advanced Suggestions*.

When I attach a HeightField to a DEM or GRD file, how do I set the Scale so that the HeightField matches the Z axis?

Use the following equation to calculate the HeightField *Scale*:

Scale = (DataMax - DataMin) / (0.5 * sqrt((xMax-xMin)^2 + (yMax-yMin)^2))

To adjust for the offset of the DataMin value, insert a Transform module between the Gridder output and the HeightField input and offset the Z axis by the DataMin value.

Terminology note: DataMax and DataMin are referring to Zmax and Zmin

Chapter 12 - StreamLines and VectorPlots

StreamLines Module

The **Network | Graphics Output | StreamLines** command adds a *StreamLines* module.

The *StreamLines* module computes streamlines through a velocity field, a distribution of velocity in a given region. Streamlines are lines within a volume of space that indicate flow direction and magnitude. The technique injects massless particles at specified seed points and traces their paths through the field. The particle stops when either the new velocity is zero, the maximum stream length is exceeded, or when the stream intersects the bounds of the field. Stream points are sampled at a constant time interval. The greater the velocity, the farther apart are the points.

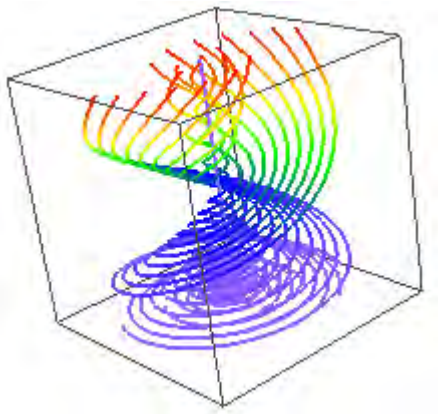
This module uses a 4th order Runge-Kutta algorithm to perform the integration of the vector field. This always uses the first three components of the lattice. Any other components are ignored. If there are less than three components, a value of 0 is assumed.

Inputs

Lattice is the input type for the *StreamLines* module.

Outputs

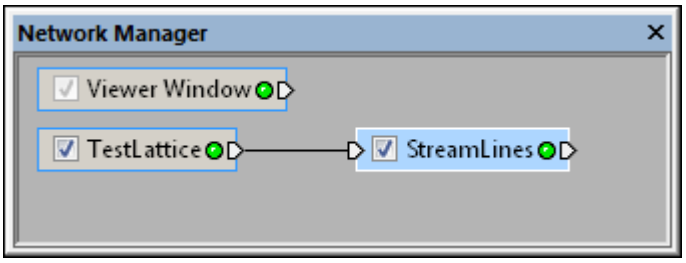
The *StreamLines* module creates an output geometry. It may be connected to the *Graphics Output Modules*. An *Info Module* may also be connected to the output node.



The StreamLines module uses lines, points, or spheres to represent the data. This example uses lines.

Properties

The *StreamLines* module properties are described below.



*Select the StreamLines module in the **Network Manager** to display its properties in the **Property Manager**.*

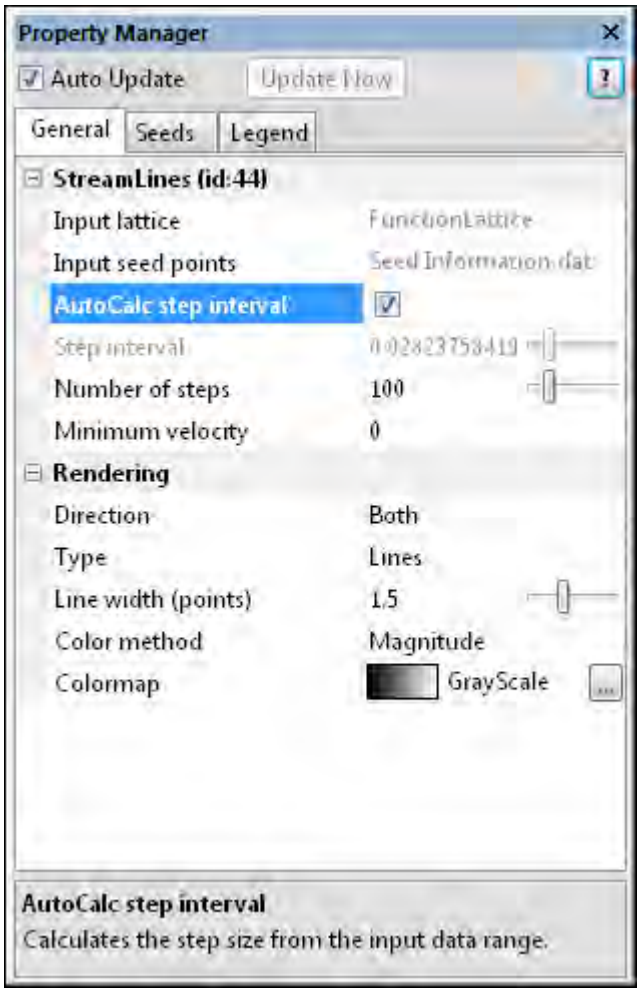
The *StreamLines* module contains the following tabs in the **Property Manager:**

General

Seeds

Legend

General Options



Customize the StreamLines properties.

Input

The *Input lattice* property shows the source lattice to which the module is connected. This option cannot be changed in the **Property Manager**, but can be changed in the **Network Manager** by changing the module input.

The *Input seed points* property shows the source data file to which the module is connected. This option cannot be changed in the **Property Manager**, but can be changed in the **Network Manager** by changing the module input.

AutoCalc Step Interval

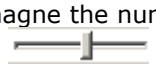
Check the box next to the *AutoCalc step interval* option to calculate the step size from the input data range. **Voxler** automatically chooses a reasonable value based on the input data.

Step Interval

The *Step interval* is the distance between steps as measured along the streamline. The interval is in map units. To change the *Step interval*, highlight the existing value and type a new value or click and drag the  to the desired value. A new velocity vector is calculated for each step. The particle trajectory is constructed using straight line segments. Using a smaller step size results in more segments and a smoother trajectory. It will take additional time and computer memory to generate the streamlines with very small values.

To return to the default *Step interval*, check the box next to the *AutoCalc step interval* command.

Number of Steps

The *Number of steps* determines how long each streamline will appear. To chagne the number of steps, highlight the existing value and type a new value or click and drag the  to the desired value. Values range between 1 and 500. The smaller the value, the smaller the streamlines that appear.

Minimum Velocity

The current streamline stops if the velocity field is less than the *Minimum velocity* value. To change the value, highlight the existing value and type a new value.

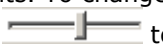
Direction

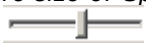
Choose the *Direction* to trace the streamline from the seed point. Options include *Forward*, *Backward*, and *Both*. To change the option, click on the existing option and select the desired option from the list.

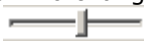
Type

The *Type* is the geometry used to represent the streamlines. Available options are *Points*, *Spheres*, and *Lines*. To change the type, click on the existing option and select the desired option from the list.

Points displays the streamlines as individual points along the streamline path. When the *Type* is set to *Points*, the *Point size* option is available. The *Point size* controls the size of the points in points.

Values range from zero to 48 points. To change the *Point size*, highlight the existing value and type a new value or click and drag the  to the desired value.

Spheres displays the streamlines as individual three dimensional spheres along the streamline path. Setting the *Type* to *Spheres* results in longer time needed to render the *StreamLines* than the other types. When the *Type* is set to *Spheres*, the *Sphere size* and *Sphere quality* options are available. The *Sphere size* controls the size of the spheres. Values range between zero and one. When set to zero, no spheres are displayed. The *Sphere quality* option changes the display of the spheres. Values range between zero and one. When set to zero, spheres are of lesser quality, but redraw faster. When set to one, spheres redraw slowly, but have high quality appearance. To change either the *Sphere size* or *Sphere quality*, highlight the existing value and type a new value or click and drag the  to the desired value.

Lines displays the streamlines as lines. When the *Type* is set to *Lines*, the *Line width (points)* option is available. The *Line width (points)* controls the thickness of the line, in points. Values range between zero and four. To change the line width, highlight the existing value and type a new value or click and drag the  to the desired value.

Color Method

The *Color method* property controls the color of the streamline. Available options are *Fixed* and *Magnitude*. *Fixed* uses a single fixed *Color* for all streamlines. *Magnitude* calculates the magnitude of the velocity field and converts it to a color using the associated Colormap. To change the method, click on the existing option and select the desired option from the list.

Color

The *Color* section specifies the fixed color. To change the color, click the colored box and choose a new color from the color palette. If the basic colors in the palette do not meet your needs, click *Other* to create a custom color.

Colormap

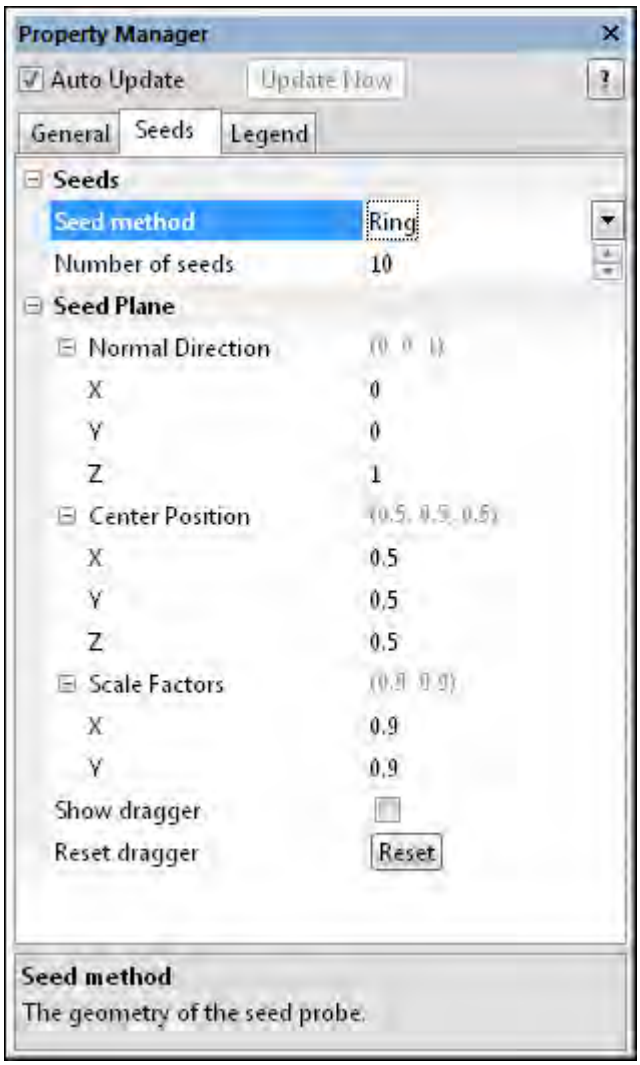
The *Colormap* property maps the vector magnitude to color. Click the bar next to *Colormap* to display the default color combinations. Click the  button to launch the **Colormap Editor** dialog.

Example

The *Streamlines* module requires an input lattice with one, two, or three components. For a three-component data file, attach three *Gridder* modules to the data module output and grid each component separately. Combine them into a single three-component lattice with a *Math* module and attach the output from the math module to the input of the streamlines module. Or, import a three component lattice instead of a data file.

StreamLines - Seeds Page

The *StreamLines* module **Seeds** page displays the starting location for all the points where streamlines should appear. To open the **Seeds** page, click on the *StreamLines* module in the **Network Window**. In the **Property Manager**, click on the **Seeds** tab.



This is an example of the information displayed in the Property Manager on the Seeds tab for a StreamLines module.

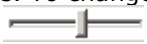
Seed Method

The *Seed method* specifies the geometry of the seed probe. Available options are *Line*, *Ring*, *Square*, and *Point set*. To change the method, click on the existing option and select the desired option from the list.

The *Line* method starts all seeds on a single line. The *Ring* method starts all seeds in a circle. The *Square* method to use the specified number of seed points as the number of seeds per side of the square. This results in the *Number of seeds* being squared.

The *Point set* option requires a data file connected to the *StreamLines* module. The data file contains the points where seeds should be started. This allows more control over seed point placement. In this case, the first N points are loaded from the connected point set, where N is the number of seed points specified in the *Number of seeds* option.

Number of Seeds

The *Number of seeds* contains the number of starting locations for streamlines. To change this value, highlight the existing value and type a new value or click and drag the  to increase or decrease the number of seeds. Values range from 1 to 100. The larger the value, the longer it will take to generate the streamlines and the more streamlines will appear.

Seed Plane

Click the ⊞ to open the *Seed Plane* section. This section controls the initial placement of the seeds.

Normal Direction

The *Normal Direction* is the direction of the normal (perpendicular vector) to the plane containing the seed points. Click the ⊞ to open the *Normal Direction* section. The normal direction has an *X*, *Y*, and *Z* component. To change the normal direction, highlight the existing value and type a new value.

Center Position

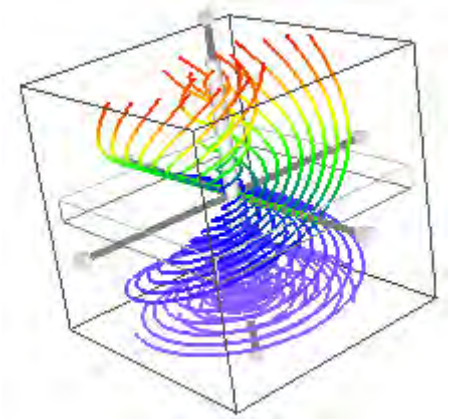
The *Center Position* is the position of the plane containing the seed points (in world coordinates). Click the ⊞ to open the *Center Position* section. The center position has an *X*, *Y*, and *Z* component. To change the center position, highlight the existing value and type a new value.

Scale Factor

The *Scale Factors* are the scale factors for the seed plane. A scale factor of 1.0 makes the seed distribution as wide as the *X* or *Y* extents of the volume. Click the ⊞ to open the *Scale Factors* section. The scale factors have an *X* and *Y* component. To change the scale factor, highlight the existing value and type a new value.

Dragger

Check the box next to the *Show dragger* option to display a jack dragger that allows the seed points to be moved or rotated graphically. When the dragger is visible, the seed point locations are lighted green.



The StreamLine dragger is used to interactively change the position of the seed plane.

Invoke rotations by clicking and dragging the line parts of the 3 principal "axes" of the dragger. Uniform scale operations can be completed by dragging any of the 6 cubes.

Translation is performed with the dragger by picking the flat transparent box or the solid middle part of the axis. Pressing the SHIFT key while translating the planar portion constrains to one of the principal axes.

Reset Dragger

Click the *Reset dragger* button to reposition the dragger and seed point locations back to the XY plane in the center of the lattice.

VectorPlot Module

The **Network | Graphics Output | VectorPlot** command adds a *VectorPlot* module.

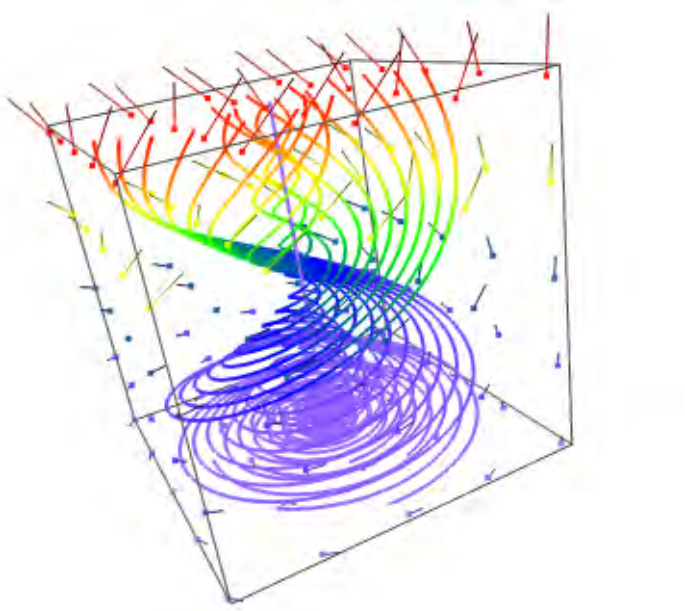
The *VectorPlot* module displays vectors on a three-dimensional lattice or point set. Care should be used with this visualization technique, as it is easy to add too many vectors and clutter up the display, making it difficult to discern patterns. This module uses the first three components of an input lattice. If the lattice has fewer components, those components are treated as zero values. Arrows are drawn with the base at the node location. It is often useful to combine this module with another module such as *StreamLines*.

Inputs

Lattice and point data are input types for the *VectorPlot* module.

Outputs

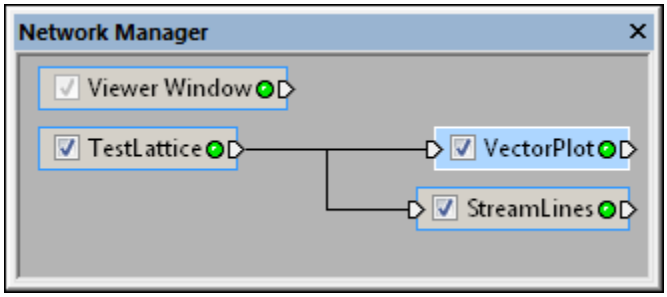
The *VectorPlot* module creates an output geometry. It may be connected to the *Graphics Output Modules*. An *Info Module* may also be connected to the output node.



The VectorPlot module output can be enhanced when used in conjunction with the StreamLines module.

Properties

The *VectorPlot* module properties are described below.



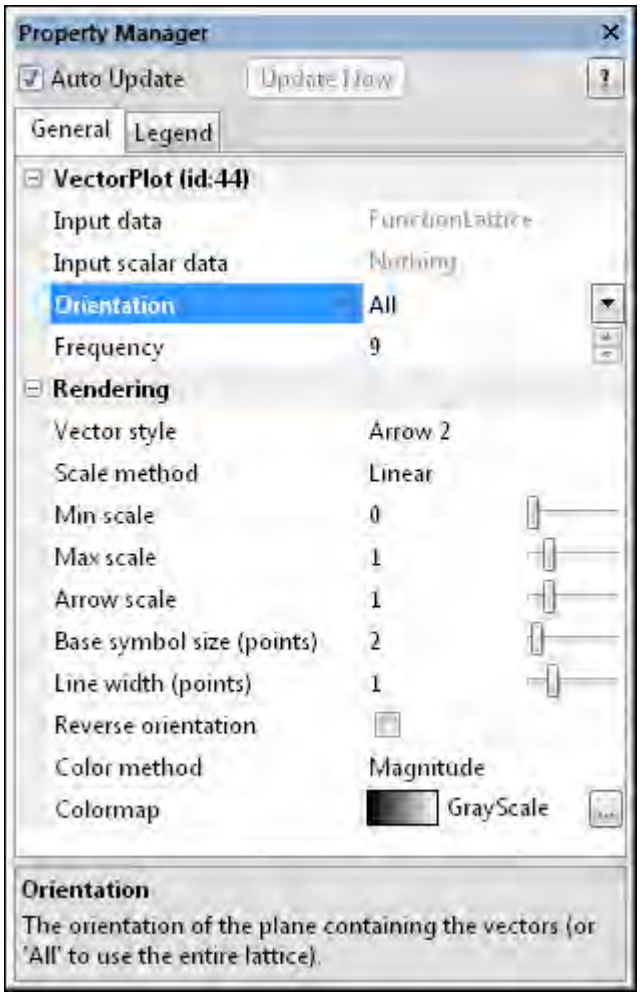
*Select the VectorPlot module in the **Network Manager** to display its properties in the **Property Manager**.*

The *VectorPlot* module contains the following tabs in the **Property Manager:**

General

Legend

General Options



Customize the VectorPlot properties.

Input

The *Input data* property shows the source data to which the module is connected. This option cannot be changed in the **Property Manager**, but can be changed in the **Network Manager** by changing the module input.

The *Input scalar data* property shows the source scalar data to which the module is connected. This option cannot be changed in the **Property Manager**, but can be changed in the **Network Manager** by changing the module input.

Orientation

The *Orientation* property allows the vectors to be constrained to a plane in one of the three cardinal directions. Available options are *XY plane*, *XZ plane*, *YZ plane*, or *All*. *All* displays vectors from the entire three-dimensional lattice. To change the orientation, click on the existing option and select the desired direction from the list.

Frequency

The *Frequency* option indicates the sampling frequency along each dimension for lattices, or for the entire data sequence for point sets. For example, if the frequency is 4, then every fourth vector is displayed. Value should be between 1 and the number of nodes in each direction in the lattice. The larger the value, the less vectors are displayed in the **Viewer** window. To change the value, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.

Vector Style

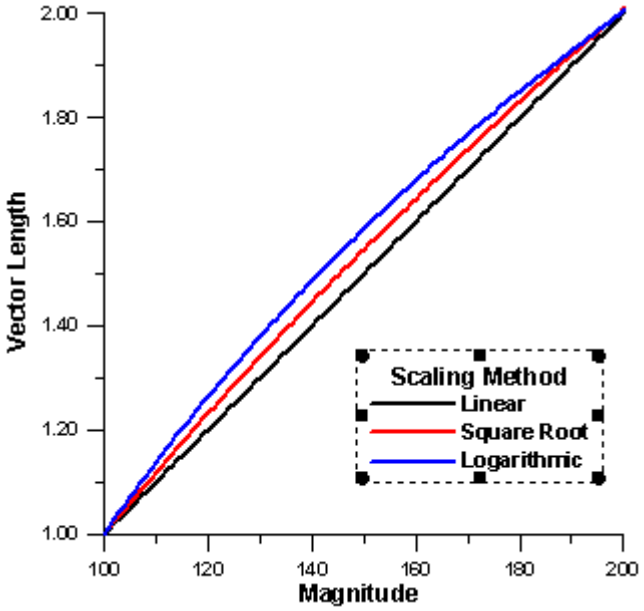
Choose a *Vector style* to determine the symbol style used to represent each vector. Choices include *Line*, *Arrow 2*, *Arrow 4*, *Arrow cross*, *Arrow solid*, and *Fade*. To change the vector style, click on the existing vector style name and select the desired option from the list.

The *Line* style creates a line with no arrowhead. The line color for each line is the same and is based on the lattice component value at that location. The *Arrow 2*, *Arrow 4*, *Arrow cross*, and *Arrow solid* creates a line with an arrowhead. The color of the line is the same and is based on the lattice component value at that location. The *Arrow 2* has 2 lines creating the arrowhead. The *Arrow 4* has four lines creating the arrowhead. The *Arrow cross* has four 3D lines creating the arrowhead. The *Arrow solid* creates a square pyramid for the arrowhead.

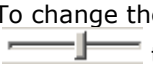
The *Fade* style creates a line with no arrowhead. The color of the line fades from the specified color or colormap to black.

Scale Method

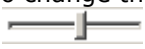
Choose the method used to scale the vector symbols. There are three ways to scale vectors between the minimum and maximum data values: linearly, logarithmically, or by square root. *Linear* scaling provides a better visual representation of symbols that, for the most part, are scaled in one dimension (such as arrows with varying shaft length). When scaling the arrows in two dimensions (symbol width and shaft length), *Square root* or *Logarithmic* scaling displays the arrows more effectively. This is because the symbol size is perceived by the human eye as having an enclosed area, and area is a distance-squared function. To change the scaling method, click on the existing option and select the desired option from the list.



Min Scale

The *Min scale* is the smallest size allowed for vectors. The smallest component value is mapped to the *Min scale* value. To change the *Min scale*, highlight the existing value and type the desired value or click and drag the  to the desired value. Values can be between zero (no visible vector) and 5. The larger the number, the larger the vectors that are displayed.

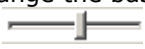
Max Scale

The *Max scale* is the largest size allowed for vector. The largest component value is mapped to the *Max scale* value. To change the *Max scale*, highlight the existing value and type the desired value or click and drag the  to the desired value. Values can be between zero (no visible vector) and 5. The larger the number, the larger the vectors that are displayed.

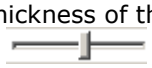
Arrow Scale

The *Arrow scale* option sets the scale factor for the arrowhead portion of the vector symbol. To change the size of the arrowheads, highlight the existing value and type the desired value or click and drag the  to increase or decrease the arrowhead size. Values can be between zero (no arrowhead) and 5. The larger the value, the larger the arrowhead. The *Arrow scale* option is not available when the *Vector style* is set to *Line* or *Fade*.

Base Symbol Size

The *Base symbol size (points)* sets the radius of the symbol located at the starting point of the vector. This value is in points and ranges from zero (no symbol) to 48 points. To change the base symbol size, highlight the existing value and type a new value or click and drag the  to increase or decrease the size. Symbol color matches the vector color.

Line Width

The *Line width (points)* sets the width of the line to use for the vectors. This value is in points and ranges from zero to four points. To change the thickness of the vector lines, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.

Reverse Orientation

Check the box next to the *Reverse orientation* command to reverse the orientation of the vectors. The symbol location does not change, but the vector goes in the opposite direction.

Color Method

Choose the method used to calculate a color for the vectors. Available options are *Fixed*, *Magnitude*, and *Scalar*. To change the color method, click on the existing option and select the desired option from the list.

The *Fixed* color method sets all of the vectors to the same color. When the *Color method* is *Fixed*, the *Color* option becomes available.

The *Magnitude* color method sets the color of each vector based on the lattice component value at the vector starting location. When the *Color method* is set to *Magnitude*, the *Colormap* option becomes available.

The *Scalar* color method requires a second lattice to be connected to the *VectorPlot* module. When the *Color method* is set to *Scalar*, the *Colormap* and *Color component* options becomes available. For each vector, the corresponding scalar value is extracted from the connected lattice and is used to look up the color from the Colormap. The same lattice can be specified for both the primary input and the scalar data. To connect a second lattice to the *VectorPlot*, import the lattice. Once the lattice appears, drag a connection from the lattice to the *VectorPlot*. Select the *Connect Scalar Lattice* option in the window that appears.

Color

The *Color* is the color of the vector. To change the color, click on the existing color and select the desired color from the list. Alternatively, click the *Other* box to open the **Colors** dialog, where you can set custom colors.

Colormap

The *Colormap* property maps the vector magnitude to color. Click the color bar next to *Colormap* to display the default color combinations. Click the  button to launch the **Colormap Editor** dialog.

Color Component

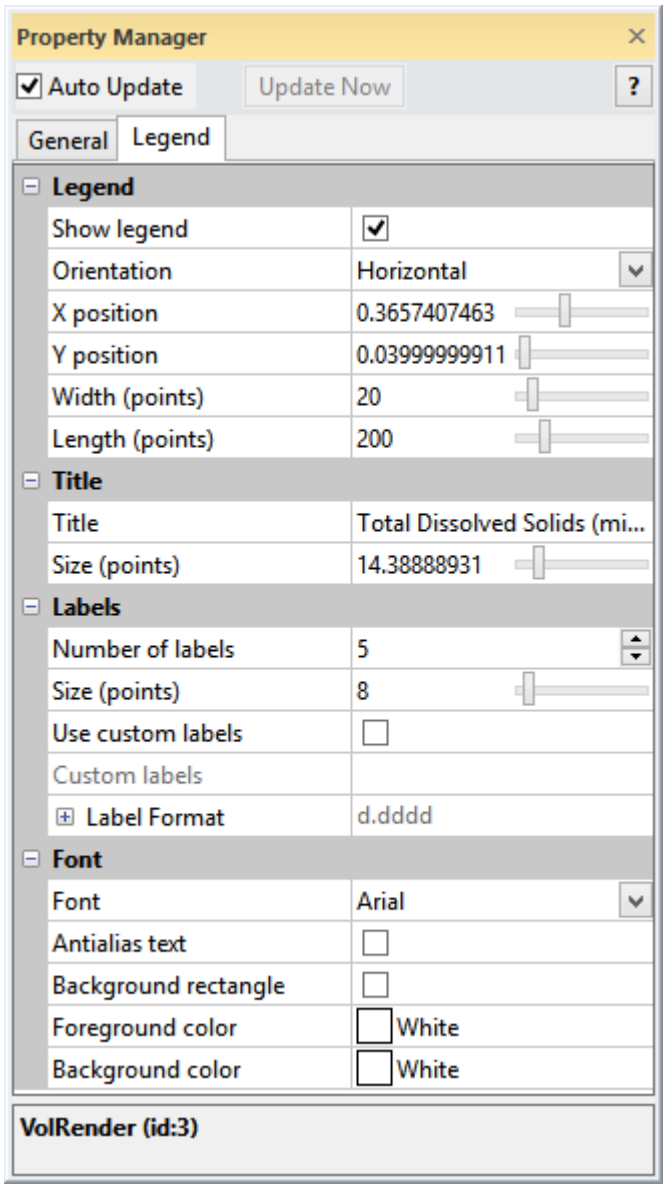
The *Color component* is the component from the *Scalar Lattice* to use for interpolating colors. The specified component is then used with the *Colormap* to determine the vector colors. To change the component, highlight the existing value and type a new value or click the  to increase or decrease the component number.

Legend

A legend is a key to a plot that labels data by color, pattern, or symbol. Legends can be added to many modules, including *FaceRender*, *HeightField*, *Isosurface*, *ScatterPlot*, *StreamLines*, *VectorPlot*, *VolRender*, and *WellRender*.

Properties

Select a module that supports legends. In the **Property Manager**, click the *Show* button next to *Legend* to display the legend in the **Viewer** window. The legend properties are described below.



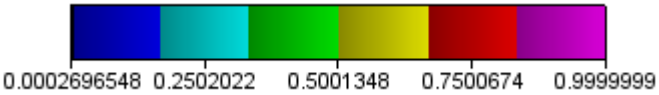
*Customize the Legend properties in the **Property Manager**.*

Show Legend

Check the box next to the *Show legend* option to display the legend for the selected module.

Orientation

Orientation specifies the direction of the "length" dimension. Available options are *Horizontal* and *Vertical* length dimension. To change the *Orientation*, click on the existing option and select the desired option from the list.



This legend has a Horizontal Orientation.

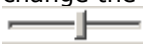


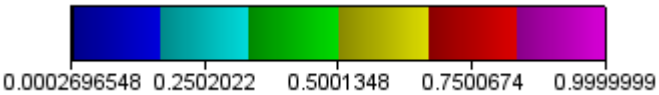
This legend has a Vertical Orientation.

X and Y Position

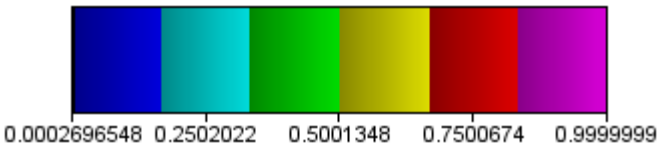
The *X position* and *Y position* control the location of the legend in the **Viewer** window. Adjust the X and/or Y position value to move the legend location. Values range from 0.0 to 1.0. 0.0 places the left bottom edge of the legend at the left edge for *X position* or the bottom edge for *Y position* in the **Viewer** window. 1.0 places the left bottom edge of the legend at the right edge for *X position* or top edge for *Y position* in the **Viewer** window. To change either value, highlight the existing value and type a new value or drag the  to increase or decrease the value.

Width

The *Width (points)* displays the width of the legend. For a vertical legend, the *Width (points)* is the distance from the left to the right of the legend. For a horizontal legend, the *Width (points)* is the distance from the top to the bottom of the legend. Values range between zero and 200 points. To change the width, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.

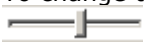


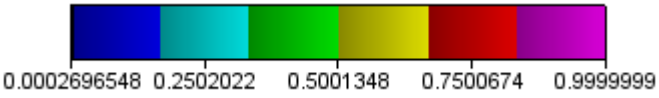
This Horizontal legend has a Width (points) set to a value of 20.



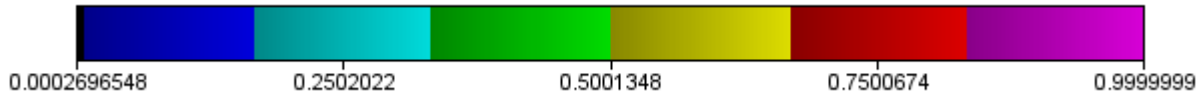
This Horizontal legend has a Width (points) set to a value of 40.

Length

The *Length (points)* displays the length of the legend. For a vertical legend, the *Length (points)* is the distance from the top to the bottom of the legend. For a horizontal legend, the *Length (points)* is the distance from the left to the right of the legend. Values range between zero and 1024 points. To change the length, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.



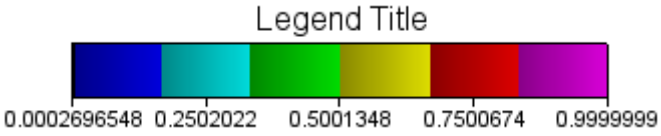
This Horizontal legend has a Length (points) set to a value of 200.



This Horizontal legend has a Length (points) set to a value of 400.

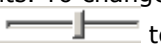
Title

Click the ⊞ next to the *Title* section to set the legend title properties. Enter an optional title to display above the legend. To enter the *Title*, click in the empty spot next to *Title*. Type the desired text. If text already appears next to *Title*, highlight this text and type new text.



This legend has a title.

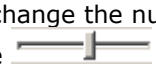
Title Size

The *Size (points)* property is the height of the title text, in points. The size can range from 4 to 72 points. To change the height, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.

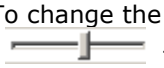
Labels

Click the ⊞ next to *Labels* to open the labels section for the legend. Set the number of labels, label height, label format, or set custom labels in this section.

Number of Labels

The *Number of labels* displays the number of text labels displayed next to the legend in the **Viewer** window. A legend can have zero labels up to 100 labels. To change the number of labels, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.

Label Size

The *Size (points)* of the label text is displayed in points. To change the label size, highlight the existing value and type a new value or click and drag the  to increase or decrease the value. The height can range between 4 and 72 points.

Use Custom Labels

Check the box next to *Use custom labels* to set user defined text next to the label. When this box is unchecked, automatic labels are created.

Custom Labels

When the box next to *Use custom labels* is checked, custom labels can be defined. Only labels in the proper format will be displayed. The format of the label string is "value:string." Use quotes if the label has spaces. Separate multiple labels with a comma or space. For example, the label string *0:Low* displays the text *Low* at the value 0. The label string *0:Low, 1.5: Medium, 3:0: High* would display the labels *Low*, *Medium*, and *High* at data values of 0.0, 1.5, and 3.0, respectively. To enter new custom labels, click in the empty box next to *Custom labels* and type the desired *value:label* text.

Label Format

Click the ⊞ next to *Label Format* to open the label format section and set the label format.

Font

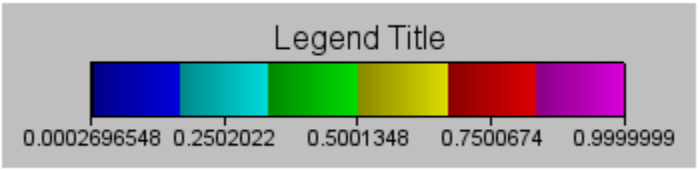
Click the ⊞ next to *Font* to open the legend font section. The *Font* displays the font to use to display the labels. To change the font, click on the existing font name and select the desired font from the list. All of the fonts installed on your computer are displayed in the list.

Antialias Text

Check the box next to *Antialias text* to make the font look smoother by slightly adjusting the colors along curves. For a crisper text appearance, uncheck the box.

Background Rectangle

Check the box next to *Background rectangle* to draw a rectangle around the legend. The rectangle is filled with the *Background color*. Uncheck the box to remove the background rectangle.



This legend has a gray background rectangle.

Foreground Color

The *Foreground color* displays the color to use as an outline for the legend and as the label color. To change the color, click on the current color and select the desired color in the color palette. Click *Other* to open the **Colors** dialog and specify a custom color.

Background Color

The *Background color* displays the color to use to fill the background rectangle. To change the color, click on the current color and select the desired color in the color palette. Click *Other* to open the **Colors** dialog and specify a custom color.

Chapter 13 - OrthoImages and ObliqueImages

OrthoImage Module

The **Network | Graphics Output | OrthoImage** command adds a *OrthoImage* module.

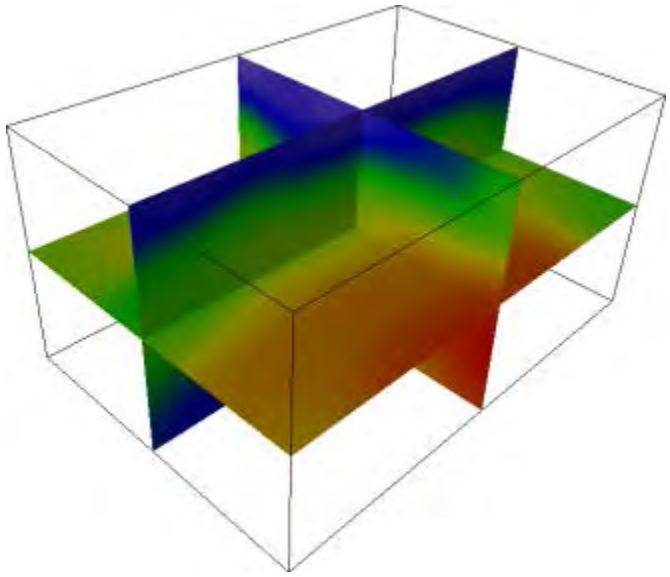
The *OrthoImage* module displays an orthogonal slice through a lattice parallel to one of the three axis planes (XY, XZ, or YZ). Orthogonal indicates elements are perpendicular or at right angles. The slice is represented by mapping data to a colormap for scalar data, or as direct RGBA colors if the lattice already contains color data. The scalar to color mapping may be specified with a linear gray mapping function with contrast enhancement or with a Colormap.

Inputs

Uniform lattice is the input type for the *OrthoImage* module.

Outputs

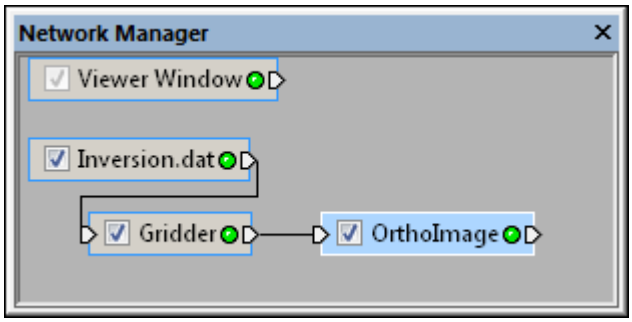
The *OrthoImage* module creates an output geometry. It may be connected to the *Graphics Output Modules*. An *Info Module* may also be connected to the output node.



OrthoImage modules in the XY, XZ, and YZ axial planes can be applied simultaneously to a lattice.

Properties

The *OrthoImage* module properties are described below.



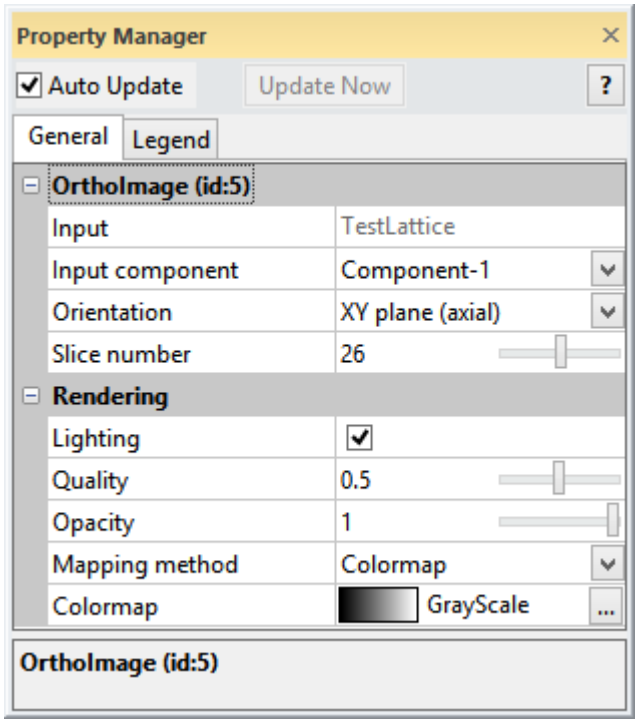
*Select the OrthoImage module in the **Network Manager** to display its properties in the **Property Manager**.*

The *OrthoImage* module contains the following tabs in the **Property Manager:**

General

Legend

General Options



Customize the OrthoImage properties.

Input

The *Input* property shows the source to which the module is connected. This option cannot be changed in the **Property Manager**, but can be changed in the **Network Manager** by changing the module input.

Input Component

The *Input component* contains the number of the component that is being displayed as the *OrthoImage*. To change the component, click the current selection and select the desired component from the list.

Orientation

Choose the direction of the *OrthoImage* slice through the volume. Choices include *XY plane (axial)*, *XZ plane (coronal)*, *or YZ plane (sagittal).* To change the *Orientation*, click on the existing option and select the desired option. Choose one of the preset local axis plane normals. *Axial* indicates the XY plane that travels horizontally. *Coronal* indicates the XZ plane that travels vertically. *Sagittal* indicates the YZ plane that travels vertically.

Slice Number

The *Slice number* is the index of the slice to display. The *Slice number* ranges from 1 to the maximum number of nodes in the specified direction. The slice number resets to the middle of the volume when the *Orientation* is changed. It is possible to specify an orientation so that the slice is exactly on edge, in which case nothing is drawn. Rotate the camera slightly to see the image. This option is not available if a single slice exists. To change the slice displayed, highlight the existing value and type a new value or click and drag the  to increase or decrease the slice value.

Lighting

Check the *Lighting* box to modulate the image colors according to the current lighting.

Quality

The *Quality* determines how resolution of the *OrthoImage*. This value ranges from 0.0 (lowest quality) to 1.0 (highest quality). The lowest quality image uses near neighbor interpolation; the higher quality images use bilinear interpolation. Note that higher quality images take longer to render, but the appearance is smoother in the **Viewer** window. To change the *Quality*, highlight the existing value and type a new value or click and drag the  to the desired value.

Opacity

Choose the level of opacity. This value ranges from 0.0 for completely transparent images to 1.0 for completely opaque images. To change the opacity, highlight the existing value and type a new value or click and drag the  to the desired value.

Mapping Method

The *Mapping method* appears only when the lattice contains scalar data. Lattices containing RGBA data are mapped directly and do not need this option. Available options are *Linear* and *Colormap*. To change the method, click on the existing method and select the desired method from the list.

The *Linear* mapping method allows specification of a linear scaling window similar to that used in medical image viewing programs. The window refers to a contiguous subset of the data specified by the *Center* and *Width* of the range. The window is linearly scaled from black at the bottom to white at the top. When the *Mapping method* is set to *Linear*, the *Width* and *Center* options become available.

The *Width* is the amount of variation allowed over the entire image. This value ranges from zero to two. A value of zero results in a sharp change in color from black to white. A value of two results in a gradual change in color. The *Center* is the location of the data window. The *Center* ranges from -1 to +1. A value of zero centers the data window. Change the *Width* and *Center* values by highlighting the existing value and type a new value or by clicking and dragging the  to increase or decrease the value. Changing these values enhances the contrast of various features within the data, e.g., bone, skin.

The *Colormap* mapping method applies a *Colormap* transfer function to the scalar data to convert data to colors. To change the colors, click on the existing colormap and select the desired colormap from the list. Or, click on the  to set custom colormap options.

ObliqueImage Module

The **Network | Graphics Output | ObliqueImage** command adds a *ObliqueImage* module.

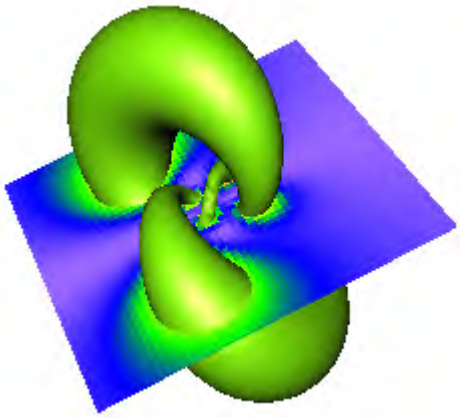
The *ObliqueImage* module displays a color image on a two-dimensional cutting plane through a lattice. In medical terminology, this is known as a multi-planar reconstruction (MPR). The slice is represented by colors mapped through a Colormap for single component data, or as direct RGBA colors for lattices containing color data.

Inputs

Lattice is the input type for the *ObliqueImage* module.

Outputs

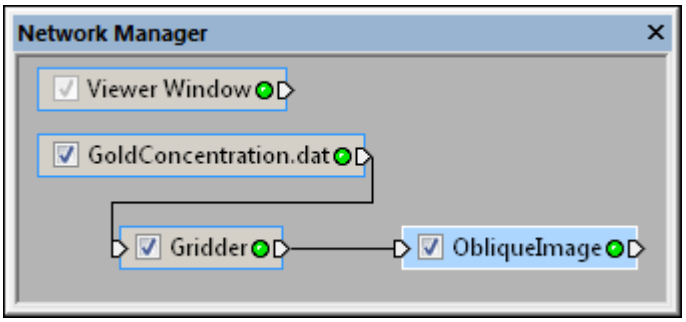
The *ObliqueImage* module creates an output geometry. It may be connected to the *Graphics Output Modules*. An *Info Module* may also be connected to the output node.



The ObliqueImage module displayed in blue above has been added to an isosurface.

Properties

The *ObliqueImage* module properties are described below.



*Select the ObliqueImage module in the **Network Manager** to display its properties in the **Property Manager**.*

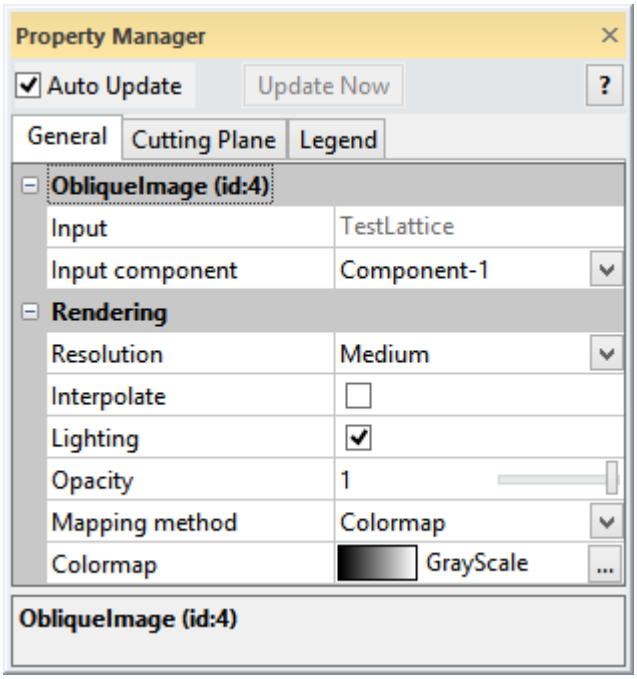
The *ObliqueImage* module contains the following tabs in the **Property Manager:**

General

Cutting Plane

Legend

General Options



Customize the ObliqueImage properties.

Input

The *Input* property shows the source to which the module is connected. This option cannot be changed in the **Property Manager**, but can be changed in the **Network Manager** by changing the module input.

Input Component

The *Input component* contains the number of the component that is being displayed as the *ObliqueImage*. To change the component, click the current selection and select the desired component from the list.

Resolution

The *Resolution* determines the sampling frequency to use along the cutting plane. Available options are *Coarse*, *Medium*, *Fine*, and *Very fine*. The finer the *Resolution*, the smoother the *ObliqueImage* appears in the **Viewer** window. To change the *Resolution*, click on the existing option and select the desired option from the list.

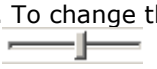
Interpolate

Check the box next to the *Interpolate* command to use trilinear interpolation to sample the lattice onto the cutting plane. If the box is not checked, then nearest neighbor interpolation is used. The trilinear interpolation often provides a more smooth image.

Lighting

Check the box next to the *Lighting* command to modulate the image colors according to the current lighting.

Opacity

Choose the level of opacity. This value ranges from 0.0 for completely transparent images to 1.0 for completely opaque images. To change the opacity, highlight the existing value and type a new value or click and drag the  to the desired value.

Mapping Method

The *Mapping method* appears only when the lattice contains scalar data. Lattices containing RGBA data are mapped directly and do not need this option. Available options are *Linear* and *Colormap*. To change the method, click on the existing method and select the desired method from the list.

The *Linear* mapping method allows specification of a linear scaling window similar to that used in medical image viewing programs. The window refers to a contiguous subset of the data specified by the *Center* and *Width* of the range. The window is linearly scaled from black at the bottom to white at the top. When the *Mapping method* is set to *Linear*, the *Width* and *Center* options become available.

The *Width* is the amount of variation allowed over the entire image. This value ranges from zero to two. A value of zero results in a sharp change in color from black to white. A value of two results in a gradual change in color. The *Center* is the location of the data window. The *Center* ranges from -1 to +1. A value of zero centers the data window. Change the *Width* and *Center* values by highlighting the existing value and type a new value or by clicking and dragging the  to increase or decrease the value. Changing these values enhances the contrast of various features within the data, e.g., bone, skin.

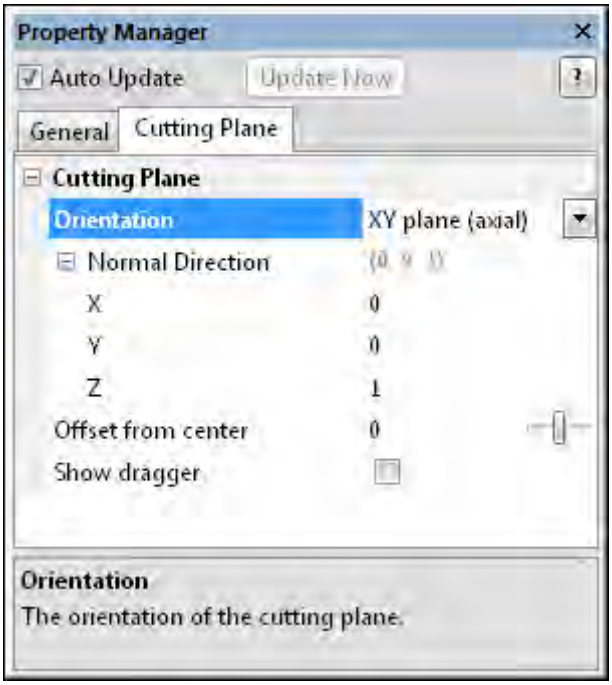
The *Colormap* mapping method applies a *Colormap* transfer function to the scalar data to convert data to colors. To change the colors, click on the existing colormap and select the desired colormap from the list. Or, click on the to set custom colormap options.

Cutting Plane

The *Cutting Plane,* a property of some modules, determines the orientation ("normal") of the cutting plane. You can choose one of the preset local axis plane normal, or you can choose an arbitrary direction by entering values for X, Y, and Z. The plane normal may be specified numerically with the *Normal Direction* property or graphically when the dragger is shown.

Properties

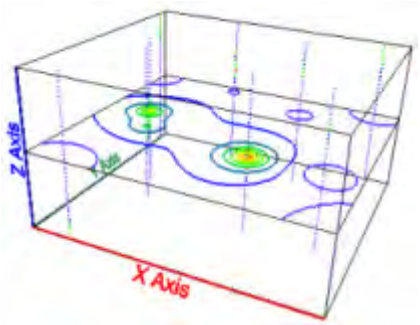
The *Cutting Plane* properties are described below. This feature may be accessed via the *Contours*, *ObliqueImage*, and *Slice* modules.



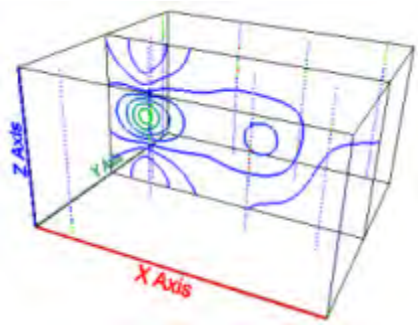
The Cutting Plane properties may be changed to affect the orientation ("normal") of the cutting plane.

Orientation

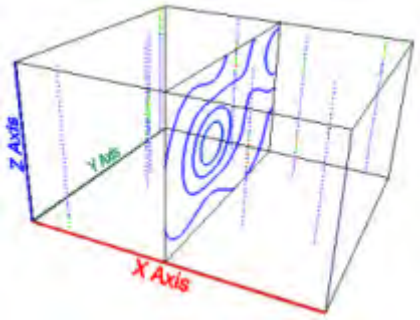
To change the *Orientation*, click on the existing option and select the desired option. Choose one of the preset local axis plane normals— *XY plane (axial)*, *XZ plane (coronal)*, or *YZ plane (sagittal)*— or choose *Custom* to enter custom *Normal Direction* values. *Axial* indicates the XY plane that travels horizontally. *Coronal* indicates the XZ plane that travels vertically. *Sagittal* indicates the YZ plane that travels vertically. The *Custom* setting lets you type a value under *Normal Direction* for each axis.



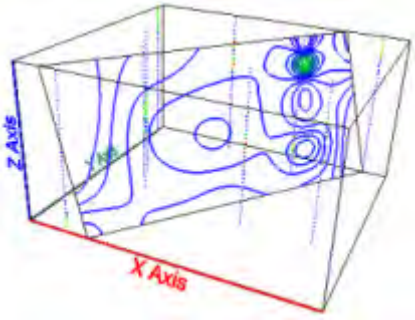
The contours Orientation is set to XY plane (axial).



The contours Orientation is set to XZ plane (coronal).



The contours Orientation is set to YZ plane (sagittal).



The contours Orientation is set to Custom.

Normal Direction

Enter a value under *Normal Direction* for the *X*, *Y*, and *Z* components of the plane normal. Changing these values changes the cutting plane orientation with respect to the normal for that axis. Changing the *X* value when the *Y* and *Z* values are zero creates a YZ cutting plane. Setting each of the *X, Y,* and *Z* values to 1 produces a plane with an oblique orientation. This vector is the normal to the cutting plane.

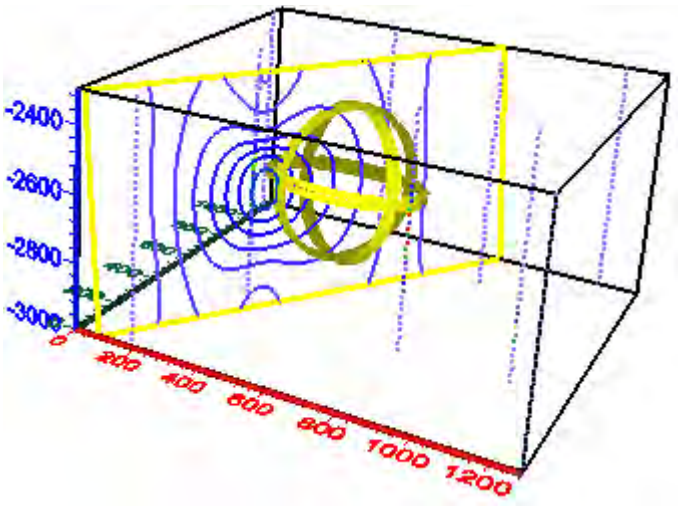
Offset from Center

The *Offset from center* property measures the distance of the cutting plane from the center of the lattice. To change the offset, highlight the existing value and type a new value or click and drag the  to increase or decrease the distance from the center. Units are in axis units. If the X axis goes from 0 to 60, a value of 0 for the *Offset from center* will place the plane at X= 30. A value of just less than 30 will place the plane at the maximum X value. A value of just greater than -30 will place the plane at the minimum X value.

Show Dragger

The *Dragger* allows interactive positioning and rotation of the plane.

Check the *Show dragger* box to show the dragger— a virtual, rotatable trackball— and allow interactive positioning and rotation of the plane. The dragger allows the orientation and offset of the cutting plane to be specified.



The Dragger is shown above in yellow and can be used to interactively rotate the cutting plane.

Drag one of the three bands to rotate around a principal axis in the direction of the ring. Drag anywhere on the ball (between the rings) to perform an unconstrained rotation in any direction.

To specify a user-defined rotation axis, press the SHIFT key while clicking the left mouse button and dragging. A new distinctively-colored axis is added.

To scale the size of the trackball, press the CTRL key and drag the trackball.

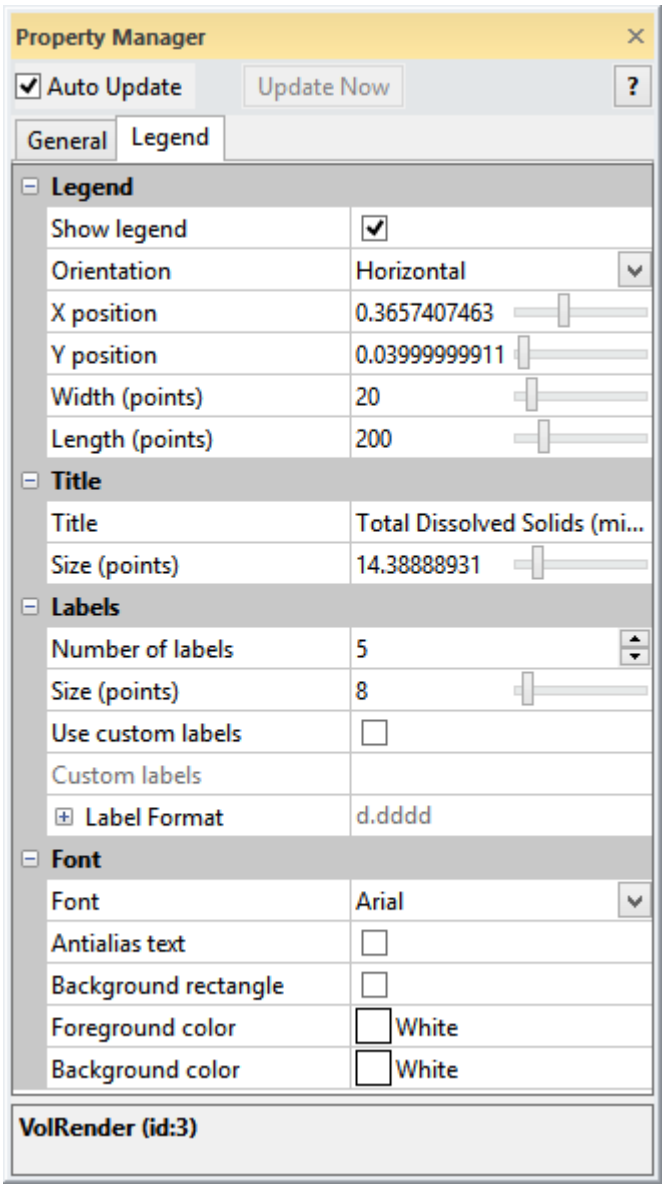
To offset the plane in the perpendicular direction, drag the cutting plane itself.

Legend

A legend is a key to a plot that labels data by color, pattern, or symbol. Legends can be added to many modules, including *FaceRender*, *HeightField*, *Isosurface*, *ScatterPlot*, *StreamLines*, *VectorPlot*, *VolRender*, and *WellRender*.

Properties

Select a module that supports legends. In the **Property Manager**, click the *Show* button next to *Legend* to display the legend in the **Viewer** window. The legend properties are described below.



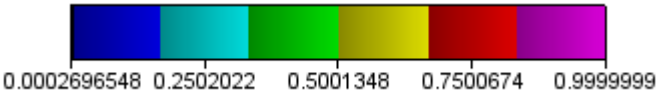
*Customize the Legend properties in the **Property Manager**.*

Show Legend

Check the box next to the *Show legend* option to display the legend for the selected module.

Orientation

Orientation specifies the direction of the "length" dimension. Available options are *Horizontal* and *Vertical* length dimension. To change the *Orientation*, click on the existing option and select the desired option from the list.



This legend has a Horizontal Orientation.

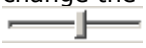


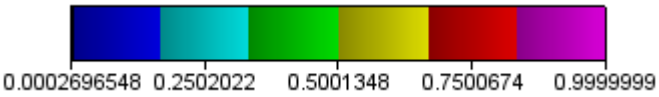
This legend has a Vertical Orientation.

X and Y Position

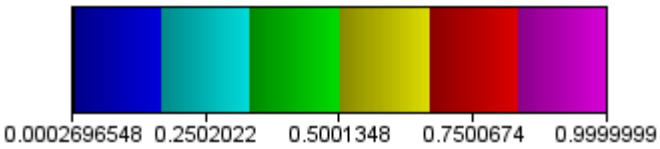
The *X position* and *Y position* control the location of the legend in the **Viewer** window. Adjust the X and/or Y position value to move the legend location. Values range from 0.0 to 1.0. 0.0 places the left bottom edge of the legend at the left edge for *X position* or the bottom edge for *Y position* in the **Viewer** window. 1.0 places the left bottom edge of the legend at the right edge for *X position* or top edge for *Y position* in the **Viewer** window. To change either value, highlight the existing value and type a new value or drag the  to increase or decrease the value.

Width

The *Width (points)* displays the width of the legend. For a vertical legend, the *Width (points)* is the distance from the left to the right of the legend. For a horizontal legend, the *Width (points)* is the distance from the top to the bottom of the legend. Values range between zero and 200 points. To change the width, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.

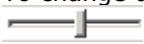


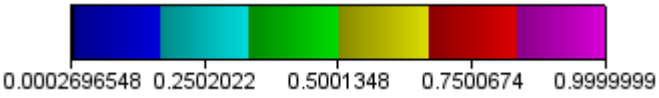
This Horizontal legend has a Width (points) set to a value of 20.



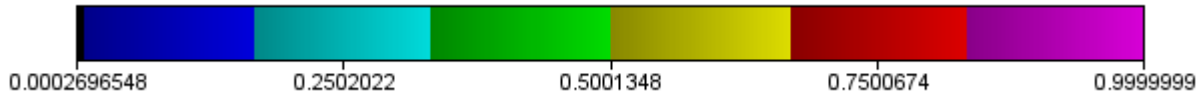
This Horizontal legend has a Width (points) set to a value of 40.

Length

The *Length (points)* displays the length of the legend. For a vertical legend, the *Length (points)* is the distance from the top to the bottom of the legend. For a horizontal legend, the *Length (points)* is the distance from the left to the right of the legend. Values range between zero and 1024 points. To change the length, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.



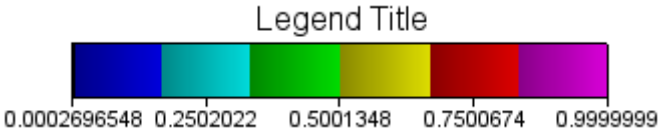
This Horizontal legend has a Length (points) set to a value of 200.



This Horizontal legend has a Length (points) set to a value of 400.

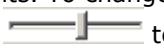
Title

Click the ⊞ next to the *Title* section to set the legend title properties. Enter an optional title to display above the legend. To enter the *Title*, click in the empty spot next to *Title*. Type the desired text. If text already appears next to *Title*, highlight this text and type new text.



This legend has a title.

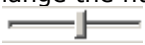
Title Size

The *Size (points)* property is the height of the title text, in points. The size can range from 4 to 72 points. To change the height, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.

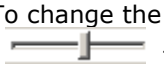
Labels

Click the ⊞ next to *Labels* to open the labels section for the legend. Set the number of labels, label height, label format, or set custom labels in this section.

Number of Labels

The *Number of labels* displays the number of text labels displayed next to the legend in the **Viewer** window. A legend can have zero labels up to 100 labels. To change the number of labels, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.

Label Size

The *Size (points)* of the label text is displayed in points. To change the label size, highlight the existing value and type a new value or click and drag the  to increase or decrease the value. The height can range between 4 and 72 points.

Use Custom Labels

Check the box next to *Use custom labels* to set user defined text next to the label. When this box is unchecked, automatic labels are created.

Custom Labels

When the box next to *Use custom labels* is checked, custom labels can be defined. Only labels in the proper format will be displayed. The format of the label string is "value:string." Use quotes if the label has spaces. Separate multiple labels with a comma or space. For example, the label string *0:Low* displays the text *Low* at the value 0. The label string *0:Low, 1.5: Medium, 3:0: High* would display the labels *Low*, *Medium*, and *High* at data values of 0.0, 1.5, and 3.0, respectively. To enter new custom labels, click in the empty box next to *Custom labels* and type the desired *value:label* text.

Label Format

Click the ⊞ next to *Label Format* to open the label format section and set the label format.

Font

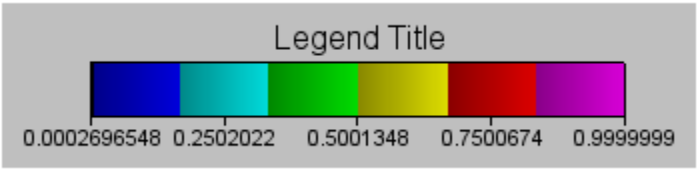
Click the ⊞ next to *Font* to open the legend font section. The *Font* displays the font to use to display the labels. To change the font, click on the existing font name and select the desired font from the list. All of the fonts installed on your computer are displayed in the list.

Antialias Text

Check the box next to *Antialias text* to make the font look smoother by slightly adjusting the colors along curves. For a crisper text appearance, uncheck the box.

Background Rectangle

Check the box next to *Background rectangle* to draw a rectangle around the legend. The rectangle is filled with the *Background color*. Uncheck the box to remove the background rectangle.



This legend has a gray background rectangle.

Foreground Color

The *Foreground color* displays the color to use as an outline for the legend and as the label color. To change the color, click on the current color and select the desired color in the color palette. Click *Other* to open the **Colors** dialog and specify a custom color.

Background Color

The *Background color* displays the color to use to fill the background rectangle. To change the color, click on the current color and select the desired color in the color palette. Click *Other* to open the **Colors** dialog and specify a custom color.

Chapter 14 - VolRenders

VolRender Module

The **Network | Graphics Output | VolRender** command adds a *VolRender* module.

Most visualization techniques convert volume data to surfaces. The *VolRender* module uses an alternative technique called direct volume rendering to render voxels directly. A voxel is short for volume pixel, the smallest distinguishable box-shaped part of a three-dimensional image.

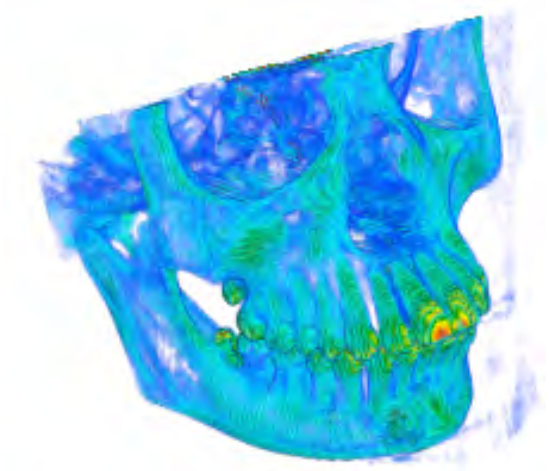
Volume rendering is a three-dimensional display of data that simulates the transmission and absorption of light through the points in the volume. Light rays are cast through the volume, where particles within the volume simultaneously emit and absorb light. The color of an individual pixel on the screen is computed by compositing the contributions from each particle intersecting the ray. This allows visualization of inhomogeneity inside objects with appropriate adjustment of the opacity.

Inputs

Uniform lattice is the input type for the *VolRender* module.

Outputs

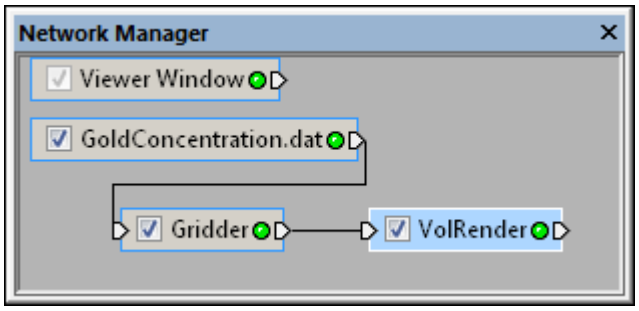
The *VolRender* module creates an output geometry. It may be connected to the *Graphics Output Modules*. An *Info Module* may also be connected to the output node.



The VolRender module uses direct volume rendering, which factors light parameters into each point.

Properties

The *VolRender* module properties are described below.



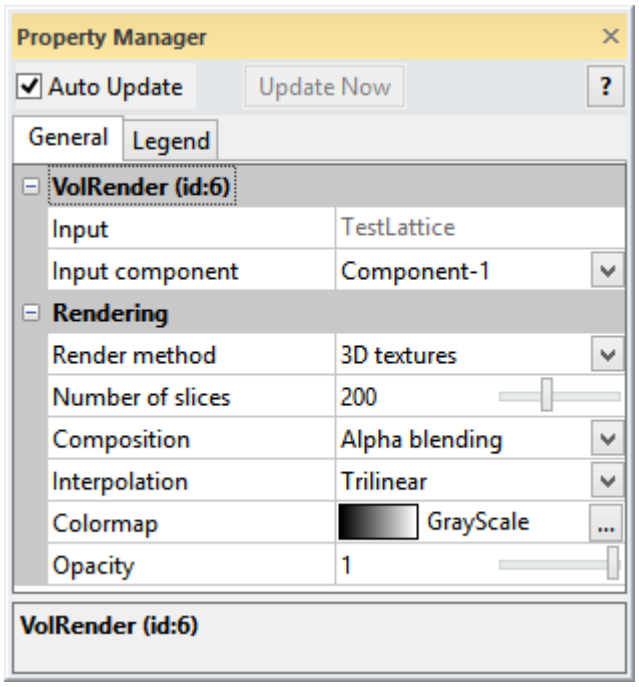
*Select the VolRender module in the **Network Manager** to display its properties in the **Property Manager**.*

The *VolRender* module contains the following tabs in the **Property Manager:**

General

Legend

General Options



Customize the VolRender properties.

Input

The *Input* property shows the source lattice to which the module is connected. This option cannot be changed in the **Property Manager**, but can be changed in the **Network Manager** by changing the module input.

Input Component

The *Input component* specifies which of the input components contains the data used to compute the volume when there is more than one component. To change the *Input component*, click the current selection and select the desired component from the list.

Render Method

Choose the *Render method* used to render the volume. Choices include *2D texture*, *3D texture*, and *Paletted textures*. The availability of the *2D textures*, *3D textures*, and *Paletted textures* is a function of your video graphics card and driver. Different cards and drivers have different features available. Some features are emulated in software and are much slower than features enabled by hardware.

For the *2D texture* method, slices parallel to the object's X, Y, or Z planes— whichever is most orthogonal to the viewing direction— are rendered with a blending function to composite the pixels between slices. The color data is applied via two-dimensional texture mapping hardware on the video card. This method should work on most video hardware, but it requires three times more memory and has a "popping" effect as the graphics are rotated. This is normal and occurs when a different set of texture planes are selected to keep them as orthogonal to the viewing direction as possible. This method does not perform texture tiling, meaning that the volume dimensions after rounding up to an even power of two must be less than or equal to the value returned by the video driver.

The *3D texture* method renders slices parallel to the image plane with a blending function to composite the pixels between slices. The color data is applied via three-dimensional texture mapping hardware or software emulation on the video card. This method does not exhibit the popping effect, plus it uses less memory than the *2D texture* method; however, it requires three-dimensional texture support from the video hardware. This method is also more prone to aliasing effects with some data sets. This option is not shown if the current video hardware does not support it.

The *Paletted textures* option is available only as an extension to OpenGL. It is available with some video cards, including older NVIDIA and Intel cards. In some cases it provides faster updates (redraws) and uses one-quarter of the texture memory when the *Colormap* is modified. Each texel (pixel) in the texture requires only a single byte (8 bits). This is then used as an index into a table of 256 RGBA colors. There can only be 256 unique colors, but the texture maps (which can be quite large for a large three-dimensional volume) are only 1 byte per texel instead of 4 (for RGBA). For example, a 512 x 512 x 512 texture would require only 128MB rather than 512MB.

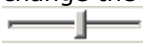
The *Paletted textures* option may be implemented in software on some video cards, and it is actually slower than the alternative RGBA (non-paletted) mode. This option is turned on, if available. You may wish to turn it off to compare the redraw performance.

The *Paletted textures* option is usually not displayed if the video card does not support it, though some video cards erroneously report that they do support the option when in fact it is not supported. The option is also not displayed when the *VolRender* module is attached to an image or RGBA lattice, since colors are taken directly from the RGB values in the lattice.

The *3D textures* and *Paletted textures* may be slow if these features are emulated by software on your graphics card. Newer graphics cards enable the features through hardware and are much faster.

Number of Slices

When the *Render method* is set to *3D textures*, the *Number of slices* option becomes available. This is the number of 2D sample slices that are created through the 3D volume. Increasing the number smooths the 3D texture, but uses more memory. Decreasing the value speeds redraw, but results in a less accurate depiction of the volume. Values range between 1 and 512. The default is 200. To

change the number, highlight the existing value and type the desired value or click and drag the  to increase or decrease the value.

Composition

The *Composition* options control how voxels are composited along the viewing rays. Available options are *Maximum intensity*, *Sum intensity*, and *Alpha blending*. To change the composition, click on the existing option and select the desired option from the list.

Maximum intensity uses the brightest pixel along each viewing ray, resulting in a display similar to an X-ray image. Note that there is no shading effect in this method. *Sum intensity* adds the values of the pixels along each ray. *Alpha blending* combines the pixels using a blending function.

Interpolation

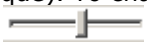
The *Interpolation* is the method used within the lattice to interpolate values between nodes. Available options are *Nearest neighbor* and *Trilinear*. To change the interpolation, click on the existing option and select the desired option from the list.

Nearest neighbor uses the lattice node closest to the location for the value. *Trilinear* interpolates several lattice node values around the location to create the value. As a result, *Trilinear* produces a smoother gradation.

Colormap

The *Colormap* property maps the component value to a color. To change the colormap, click the bar next to *Colormap* to display the default color combinations. Click the desired colormap from the list. Or, click the  button to launch the **Colormap Editor** dialog. The *Colormap* option is not displayed when the *VolRender* module is attached to an image or RGBA lattice since colors are taken directly from the RGB values in the lattice. Blanked lattice nodes less than the *Colormap* minimum are displayed with the color specified for the *Colormap* maximum. Lattice nodes that are greater than the *Colormap* maximum are displayed with the color specified for the *Colormap* maximum.

Opacity

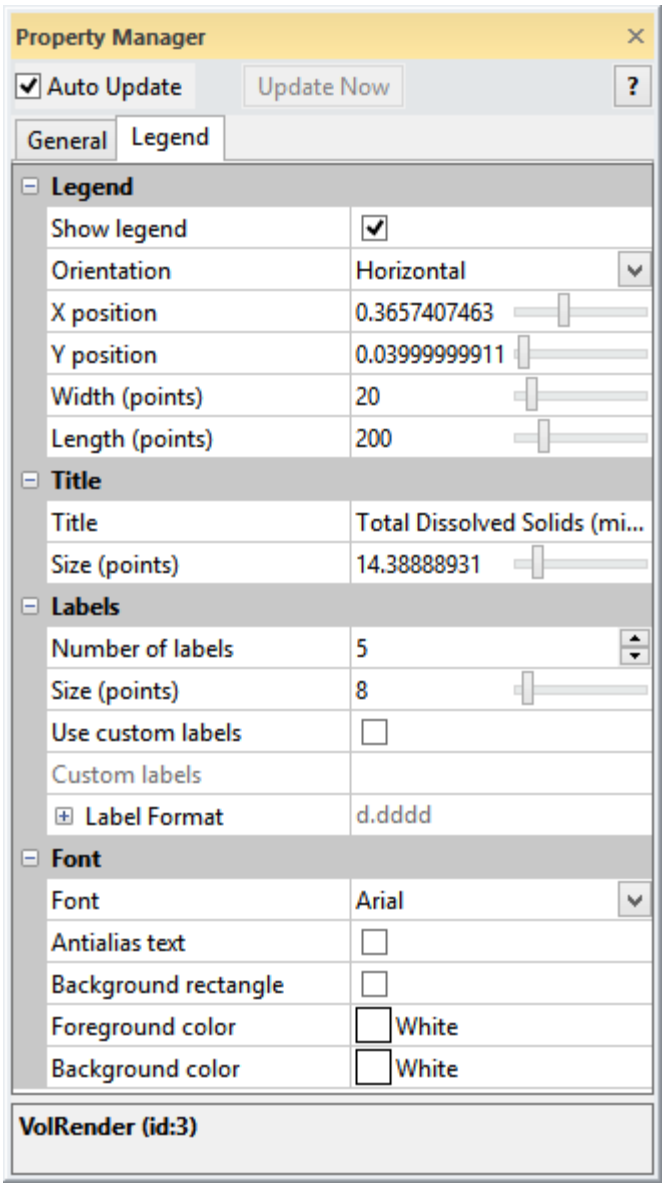
Choose the level of opacity. This value ranges from 0.0 (completely transparent) to 1.0 (completely opaque). To change the level, highlight the existing value and type a new value or click and drag the  to the desired value. The opacity is combined with the opacity specified in the **Colormap Editor** dialog.

VolRender - Legend Page

A legend is a key to a plot that labels data by color, pattern, or symbol. Legends can be added to many modules, including *FaceRender*, *HeightField*, *Isosurface*, *ScatterPlot*, *StreamLines*, *VectorPlot*, *VolRender*, and *WellRender*.

Properties

Select a module that supports legends. In the **Property Manager**, click the *Show* button next to *Legend* to display the legend in the **Viewer** window. The legend properties are described below.



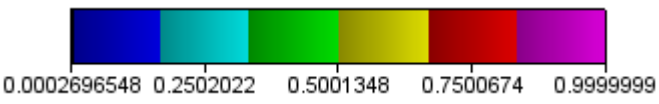
Customize the Legend properties in the ***Property Manager****.*

Show Legend

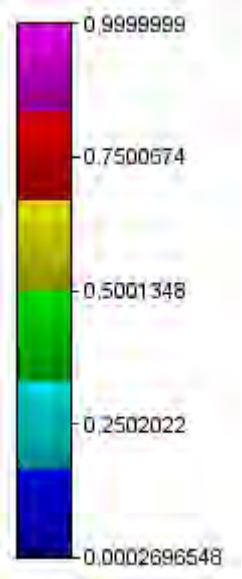
Check the box next to the *Show legend* option to display the legend for the selected module.

Orientation

Orientation specifies the direction of the "length" dimension. Available options are *Horizontal* and *Vertical* length dimension. To change the *Orientation*, click on the existing option and select the desired option from the list.



This legend has a Horizontal Orientation.



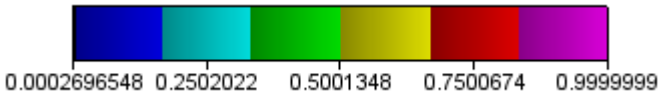
This legend has a Vertical Orientation.

X and Y Position

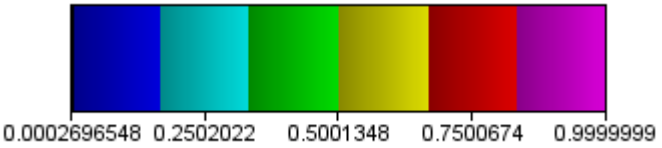
The *X position* and *Y position* control the location of the legend in the **Viewer** window. Adjust the X and/or Y position value to move the legend location. Values range from 0.0 to 1.0. 0.0 places the left bottom edge of the legend at the left edge for *X position* or the bottom edge for *Y position* in the **Viewer** window. 1.0 places the left bottom edge of the legend at the right edge for *X position* or top edge for *Y position* in the **Viewer** window. To change either value, highlight the existing value and type a new value or drag the  to increase or decrease the value.

Width

The *Width (points)* displays the width of the legend. For a vertical legend, the *Width (points)* is the distance from the left to the right of the legend. For a horizontal legend, the *Width (points)* is the distance from the top to the bottom of the legend. Values range between zero and 200 points. To change the width, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.



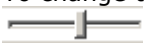
This Horizontal legend has a Width (points) set to a value of 20.

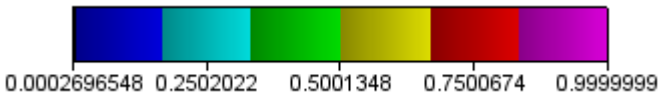


This Horizontal legend has a Width (points) set to a value of 40.

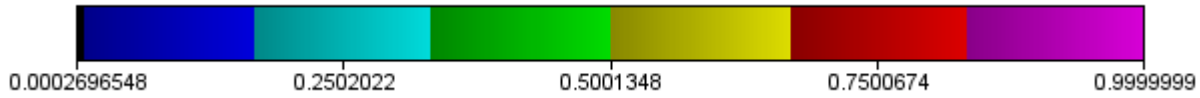
Length

The *Length (points)* displays the length of the legend. For a vertical legend, the *Length (points)* is the distance from the top to the bottom of the legend. For a horizontal legend, the *Length (points)* is the distance from the left to the right of the legend. Values range between zero and 1024 points.

To change the length, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.



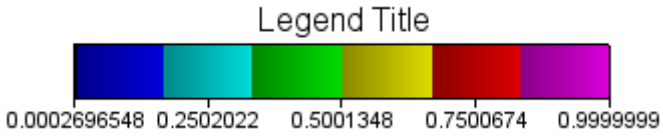
This Horizontal legend has a Length (points) set to a value of 200.



This Horizontal legend has a Length (points) set to a value of 400.

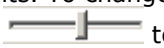
Title

Click the ⊞ next to the *Title* section to set the legend title properties. Enter an optional title to display above the legend. To enter the *Title*, click in the empty spot next to *Title*. Type the desired text. If text already appears next to *Title*, highlight this text and type new text.



This legend has a title.

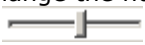
Title Size

The *Size (points)* property is the height of the title text, in points. The size can range from 4 to 72 points. To change the height, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.

Labels

Click the ⊞ next to *Labels* to open the labels section for the legend. Set the number of labels, label height, label format, or set custom labels in this section.

Number of Labels

The *Number of labels* displays the number of text labels displayed next to the legend in the **Viewer** window. A legend can have zero labels up to 100 labels. To change the number of labels, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.

Label Size

The *Size (points)* of the label text is displayed in points. To change the label size, highlight the existing value and type a new value or click and drag the  to increase or decrease the value. The height can range between 4 and 72 points.

Use Custom Labels

Check the box next to *Use custom labels* to set user defined text next to the label. When this box is unchecked, automatic labels are created.

Custom Labels

When the box next to *Use custom labels* is checked, custom labels can be defined. Only labels in the proper format will be displayed. The format of the label string is "value:string." Use quotes if the label has spaces. Separate multiple labels with a comma or space. For example, the label string *0:Low* displays the text *Low* at the value 0. The label string *0:Low, 1.5: Medium, 3:0: High* would display the labels *Low*, *Medium*, and *High* at data values of 0.0, 1.5, and 3.0, respectively. To enter new custom labels, click in the empty box next to *Custom labels* and type the desired *value:label* text.

Label Format

Click the ⊞ next to *Label Format* to open the label format section and set the label format.

Font

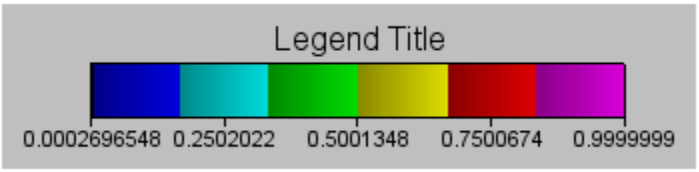
Click the ⊞ next to *Font* to open the legend font section. The *Font* displays the font to use to display the labels. To change the font, click on the existing font name and select the desired font from the list. All of the fonts installed on your computer are displayed in the list.

Antialias Text

Check the box next to *Antialias text* to make the font look smoother by slightly adjusting the colors along curves. For a crisper text appearance, uncheck the box.

Background Rectangle

Check the box next to *Background rectangle* to draw a rectangle around the legend. The rectangle is filled with the *Background color*. Uncheck the box to remove the background rectangle.



This legend has a gray background rectangle.

Foreground Color

The *Foreground color* displays the color to use as an outline for the legend and as the label color. To change the color, click on the current color and select the desired color in the color palette. Click *Other* to open the **Colors** dialog and specify a custom color.

Background Color

The *Background color* displays the color to use to fill the background rectangle. To change the color, click on the current color and select the desired color in the color palette. Click *Other* to open the **Colors** dialog and specify a custom color.

How do I blank a lattice above an irregular topography surface?

Use a *Math* module with a Boolean IF function to blank the area above a surface.

Attach the 3D lattice as input "A" and a grid file with the topography as input "B" to the input of the *Math* module.

Use an IF function with the following syntax:

IF Z > B, 0, A

Or

Z > B? 0:A

With a GRD file, the elevation data is imported as component 1.

Both statements say to set output value to 0 if the Z value is greater than the elevation data in the GRD file (or "B"). If the Z value is less than the elevation value in B, use the data value in A. In this example, a value of 0 is used, but any value less than the Z minimum of the data can be used.

Connect a *VolRender* module to the output of the *Math* module, and set the 0 values to be transparent in the **Colormap editor** accessed via the ellipsis button ... to the right of the Colormap example in the VolRender Property Manager.

The same principle can be applied to blanking to the right of an X value in a BLN file. Remove the BLN header, save to a DAT file in **Surfer**, and grid the DAT file using the X column for the Z value. In **Voxler**, attach the *Gridder* module and .GRD file to a *Math* module as above, and change the reference in the *Math* equation from Z to X:

X > B? 0 : A

Chapter 15 - FaceRenders

FaceRender Module

The **Network | Graphics Output | FaceRender** command adds a *FaceRender* module.

The *FaceRender* module displays uninterpolated cubes of an input lattice. A *FaceRender* cube represents one unit in each of the X, Y, and Z directions. Component values are represented by different colors in the *FaceRender.* To determine the component value and color for each cube, **Voxler** calculates the average component value by summing the values at each of the eight corner points and dividing by eight. If one or more of the corner points has a null (blank) value, that cube is not displayed. Additionally, the cube is not displayed if the color map value for the average data value for that cube is partially or fully transparent.

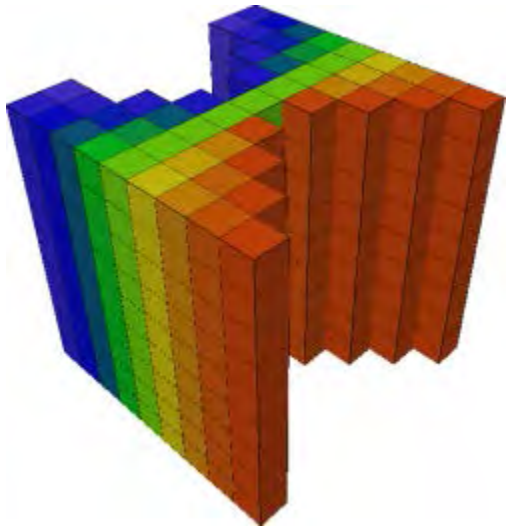
Inputs

Lattice is the input type for the *FaceRender* module.

Outputs

The *FaceRender* module creates an output geometry. It may be connected to the *Axes* and *BoundingBox Graphics Output Modules*. An *Info Module* may also be connected to the output node.

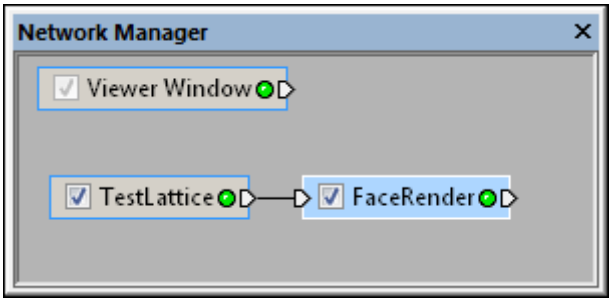
FaceRender modules cannot be clipped with a *ClipPlane* module. To limit the *FaceRender* to a portion of the total volume, change the **Geometry** or adjust the *Colormap*.



This sample vector vortex test lattice displayed as a FaceRender with some sections transparent.

Properties

The *FaceRender* module properties are described below.



*Select the FaceRender module in the **Network Manager** to display its properties in the **Property Manager**.*

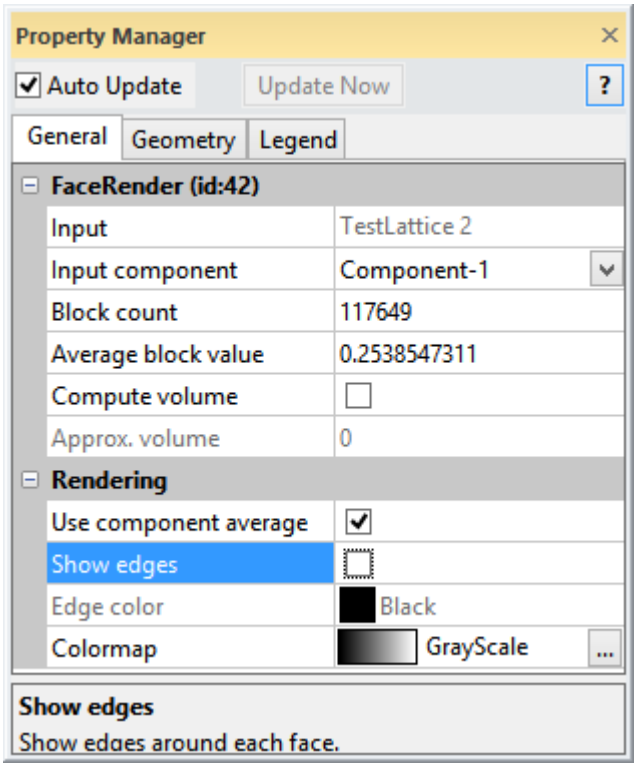
The *FaceRender* module contains the following tabs in the **Property Manager:**

General

Geometry

Legend

General Options



Customize the FaceRender properties.

Input

The *Input* property shows the source to which the module is connected. This option cannot be changed in the **Property Manager**, but can be changed in the **Network Manager** by changing the module input.

Input Component

The *Input component* specifies which of the input components contains the data used to compute the *FaceRender*. To change the *Input component*, click the current selection and select the desired component from the list.

Block Count

The *Block count* lists the number of cubes currently displayed by the *FaceRender*. This value will change as the cubes are made transparent by the *Colormap* or when the **Geometry** page is changed.

Average Block Value

The *Average block value* is the average component value of all visible blocks. The average is calculated by summing the individual block values for all displayed blocks and dividing by the *Block count* number. This value changes as the cubes are made transparent by the *Colormap* or when the **Geometry** page is changed.

Volume

Check the box next to *Compute volume* to calculate the volume displayed by all colored cubes in the *FaceRender* module. the volume is displayed next to the *Approx. volume* option. Non-displayed cubes are not included in the volume calculation. Volume calculations are generated for each block. The total *Approx. volume* is the sum of the individual volumes for all blocks. Results are provided in cubic units based on the units of the input grid file. To make volumetric sense, the X, Y, and Z units should be the same.

To increase accuracy in the volume calculations, increase the number of cells in the X, Y, and Z directions for the lattice or *Gridder*.

Use Component Average

Check the *Use component average* option to color every block by the average component value of the corners. If the *Use component average option* is unchecked, each block is colored by the component value from the lower left corner component value.

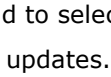
Show Edges

Check the box next to the *Show edges* command to display lines around the face for each cube. When this option is unchecked, lines are not displayed.

Edge Color

The *Edge color* displays the color of the lines that are around each cube face. To change the color, click on the existing color and select a new color from the list. Or, click the *Other* button to open the **Colors** dialog, where custom colors can be defined.

Colormap

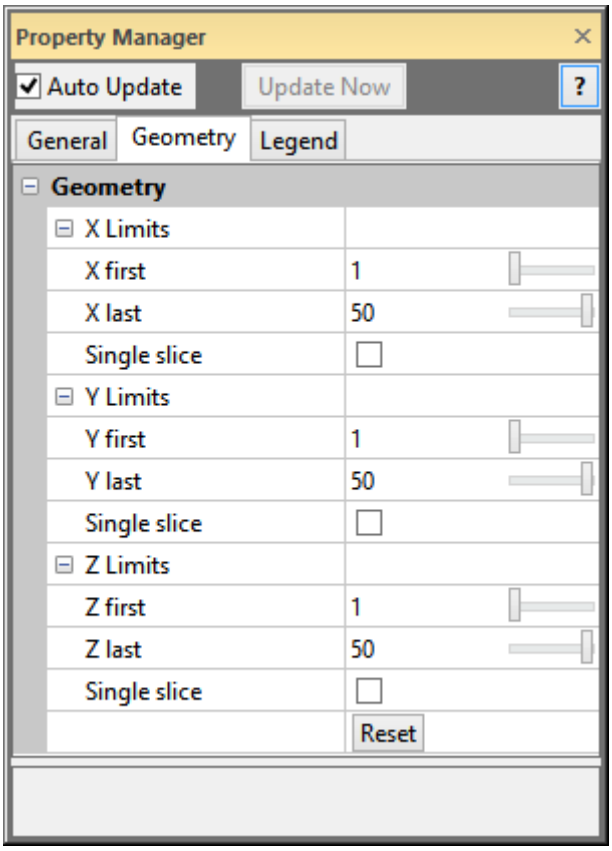
The *Colormap* maps component values to colors. To change the colormap used by the *FaceRender*, click the existing color bar to the right of the *Colormap* command to select a different colormap from the list. Click on the desired colormap and the *FaceRender* updates. Alternatively, click the  button to open the **Colormap Editor** dialog. The **Colormap Editor** dialog allows you to create a custom colormap and to change the mapping of color to data values. By default, the minimum and

maximum values are mapped to the data minimum and maximum, and a smaller range may not be present in the cutting plane.

Note that *FaceRender* cannot use partially transparent colors. Cubes are fully transparent when the *Opacity* value in the **Colormap Editor** dialog is zero. When the *Opacity* value in the **Colormap Editor** is not zero, cubes are fully opaque.

FaceRender Module - Geometry Page

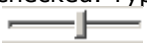
The *FaceRender* module **Geometry** page displays the extents of the *FaceRender*. To open the **Geometry** page, click on the *FaceRender* module in the **Network Window**. In the **Property Manager**, click on the **Geometry** tab.



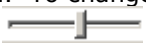
This is an example of the information displayed in the Property Manager on the Geometry tab for a FaceRender module.

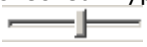
X Limits

The *X Limits* item displays the limits in the X direction for the *FaceRender*. Enter the *X first* and *X last* values. The *X first* value is the first grid node to be displayed. The *X last* value is the last grid node to be displayed. Values for *X first* and *X last* are index values for the X direction. The first node in the direction in the input grid is 1. The next node is 2, and so on. To change the value, highlight the existing value and type the desired value or click and drag the  to increase or decrease the index grid node value.

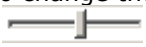
Check the *Single slice* check box to view a single *voxel* thick slice of the *FaceRender* in the X direction. When the *Single slice* check box is checked, the *X first* value determines which grid node is visible. The *X last* property is disabled since the slice is only one *voxel* thick. The *X first* property is stays limited to less than the *X last* property when the *Single slice* box is checked. Type the desired value into the *X first* field to view a specific slice. Click and drag the  to pan through the volume.

Y Limits

The *Y Limits* item displays the limits in the Y direction for the *FaceRender*. Enter the *Y first* and *Y last* values. The *Y first* value is the first grid node to be displayed. The *Y last* value is the last grid node to be displayed. Values for *Y first* and *Y last* are index values for the Y direction. The first node in the direction in the input grid is 1. The next node is 2, and so on. To change the value, highlight the existing value and type the desired value or click and drag the  to increase or decrease the index grid node value.

Check the *Single slice* check box to view a single *voxel* thick slice of the *FaceRender* in the Y direction. When the *Single slice* check box is checked, the *Y first* value determines which grid node is visible. The *Y last* property is disabled since the slice is only one *voxel* thick. The *Y first* property is stays limited to less than the *Y last* property when the *Single slice* box is checked. Type the desired value into the *Y first* field to view a specific slice. Click and drag the  to pan through the volume.

Z Limits

The *Z Limits* item displays the limits in the Z direction for the *FaceRender*. Enter the *Z first* and *Z last* values. The *Z first* value is the first grid node to be displayed. The *Z last* value is the last grid node to be displayed. Values for *Z first* and *Z last* are index values for the Z direction. The first node in the direction in the input grid is 1. The next node is 2, and so on. To change the value, highlight the existing value and type the desired value or click and drag the  to increase or decrease the index grid node value.

Check the *Single slice* check box to view a single *voxel* thick slice of the *FaceRender* in the Z direction. When the *Single slice* check box is checked, the *Z first* value determines which grid node is visible. The *Z last* property is disabled since the slice is only one voxel thick. The *Z first* property is stays limited to less than the *Z last* property when the *Single slice* box is checked. Type the desired value into the *Z first* field to view a specific slice. Click and drag the  to pan through the volume.

Reset

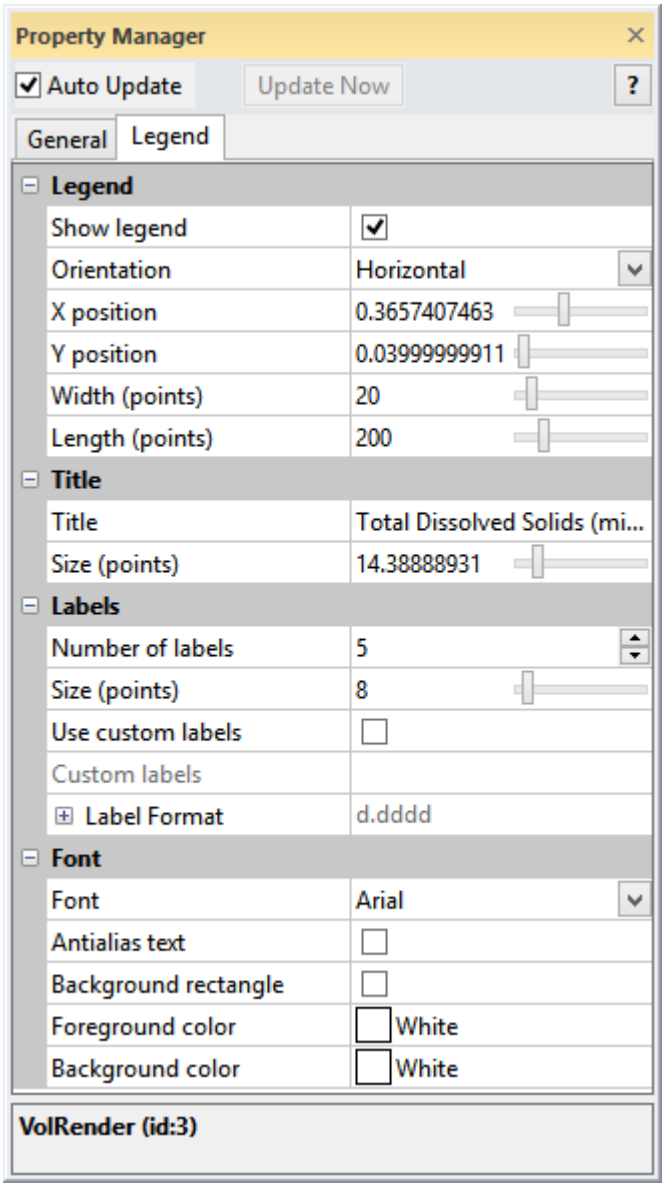
Click the *Reset* button to return all of the values to the smallest and largest possible values. Clicking the *Reset* button does not uncheck any *Single slice* checkboxes.

FaceRender - Legend Page

A legend is a key to a plot that labels data by color, pattern, or symbol. Legends can be added to many modules, including *FaceRender*, *HeightField*, *Isosurface*, *ScatterPlot*, *StreamLines*, *VectorPlot*, *VolRender*, and *WellRender*.

Properties

Select a module that supports legends. In the **Property Manager**, click the *Show* button next to *Legend* to display the legend in the **Viewer** window. The legend properties are described below.



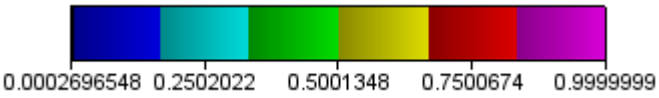
*Customize the Legend properties in the **Property Manager**.*

Show Legend

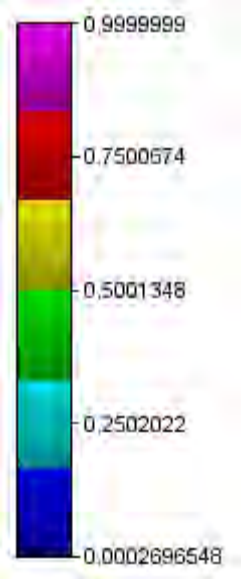
Check the box next to the *Show legend* option to display the legend for the selected module.

Orientation

Orientation specifies the direction of the "length" dimension. Available options are *Horizontal* and *Vertical* length dimension. To change the *Orientation*, click on the existing option and select the desired option from the list.



This legend has a Horizontal Orientation.

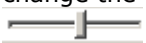


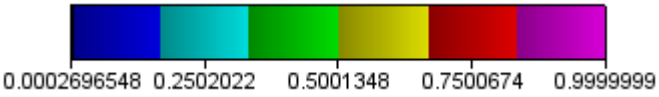
This legend has a Vertical Orientation.

X and Y Position

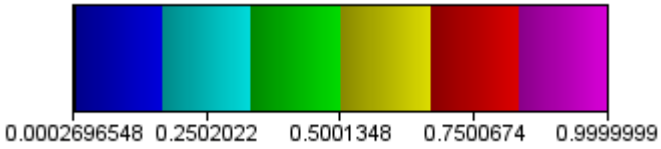
The *X position* and *Y position* control the location of the legend in the **Viewer** window. Adjust the X and/or Y position value to move the legend location. Values range from 0.0 to 1.0. 0.0 places the left bottom edge of the legend at the left edge for *X position* or the bottom edge for *Y position* in the **Viewer** window. 1.0 places the left bottom edge of the legend at the right edge for *X position* or top edge for *Y position* in the **Viewer** window. To change either value, highlight the existing value and type a new value or drag the  to increase or decrease the value.

Width

The *Width (points)* displays the width of the legend. For a vertical legend, the *Width (points)* is the distance from the left to the right of the legend. For a horizontal legend, the *Width (points)* is the distance from the top to the bottom of the legend. Values range between zero and 200 points. To change the width, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.

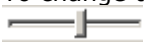


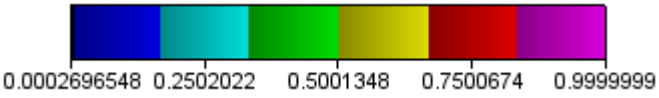
This Horizontal legend has a Width (points) set to a value of 20.



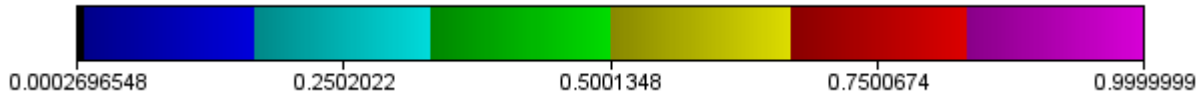
This Horizontal legend has a Width (points) set to a value of 40.

Length

The *Length (points)* displays the length of the legend. For a vertical legend, the *Length (points)* is the distance from the top to the bottom of the legend. For a horizontal legend, the *Length (points)* is the distance from the left to the right of the legend. Values range between zero and 1024 points. To change the length, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.



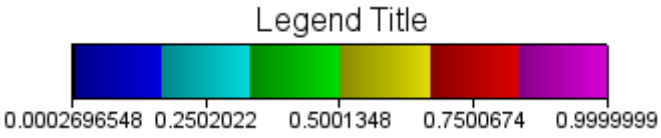
This Horizontal legend has a Length (points) set to a value of 200.



This Horizontal legend has a Length (points) set to a value of 400.

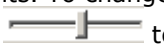
Title

Click the ⊞ next to the *Title* section to set the legend title properties. Enter an optional title to display above the legend. To enter the *Title*, click in the empty spot next to *Title*. Type the desired text. If text already appears next to *Title*, highlight this text and type new text.



This legend has a title.

Title Size

The *Size (points)* property is the height of the title text, in points. The size can range from 4 to 72 points. To change the height, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.

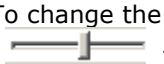
Labels

Click the ⊞ next to *Labels* to open the labels section for the legend. Set the number of labels, label height, label format, or set custom labels in this section.

Number of Labels

The *Number of labels* displays the number of text labels displayed next to the legend in the **Viewer** window. A legend can have zero labels up to 100 labels. To change the number of labels, highlight the existing value and type a new value or click and drag the  to increase or decrease the value.

Label Size

The *Size (points)* of the label text is displayed in points. To change the label size, highlight the existing value and type a new value or click and drag the  to increase or decrease the value. The height can range between 4 and 72 points.

Use Custom Labels

Check the box next to *Use custom labels* to set user defined text next to the label. When this box is unchecked, automatic labels are created.

Custom Labels

When the box next to *Use custom labels* is checked, custom labels can be defined. Only labels in the proper format will be displayed. The format of the label string is "value:string." Use quotes if the label has spaces. Separate multiple labels with a comma or space. For example, the label string *0:Low* displays the text *Low* at the value 0. The label string *0:Low, 1.5: Medium, 3:0: High* would display the labels *Low*, *Medium*, and *High* at data values of 0.0, 1.5, and 3.0, respectively. To enter new custom labels, click in the empty box next to *Custom labels* and type the desired *value:label* text.

Label Format

Click the ⊞ next to *Label Format* to open the label format section and set the label format.

Font

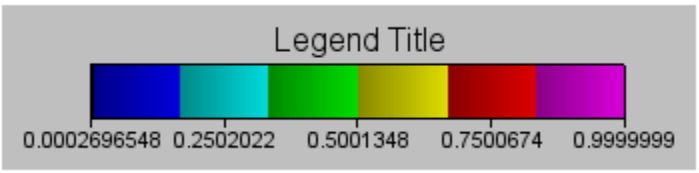
Click the ⊞ next to *Font* to open the legend font section. The *Font* displays the font to use to display the labels. To change the font, click on the existing font name and select the desired font from the list. All of the fonts installed on your computer are displayed in the list.

Antialias Text

Check the box next to *Antialias text* to make the font look smoother by slightly adjusting the colors along curves. For a crisper text appearance, uncheck the box.

Background Rectangle

Check the box next to *Background rectangle* to draw a rectangle around the legend. The rectangle is filled with the *Background color*. Uncheck the box to remove the background rectangle.



This legend has a gray background rectangle.

Foreground Color

The *Foreground color* displays the color to use as an outline for the legend and as the label color. To change the color, click on the current color and select the desired color in the color palette. Click *Other* to open the **Colors** dialog and specify a custom color.

Background Color

The *Background color* displays the color to use to fill the background rectangle. To change the color, click on the current color and select the desired color in the color palette. Click *Other* to open the **Colors** dialog and specify a custom color.

How can I calculate volume?

Voxler can calculate the volume of an *Isosurface* or *FaceRender* module. To display volumes, data should be in the same units in the X, Y, and Z directions. For instance, all directions should contain meters or feet. Using latitude and longitude or mixing units is not recommended as the volume will not have meaning.

To display the volume:

1. Attach an *Isosurface* or *FaceRender* module to a *Gridder* or lattice.
2. In the **Property Manager**, click on the **General** tab.
3. Check the box next to the *Compute volume* option.
4. The *Approx. volume* or the *Volume >= isovalue* and *Volume <=isovalue* is displayed in the **Property Manager**. This is in X, Y, Z units.

Alternatively, export lattice slices to the **Surfer** .GRD file format.

In **Voxler**, save data as .GRD:

1. Select a uniform lattice module in the **Network Manager** and choose the **File | Save Data** command to export a uniform lattice to the *GRD Surfer Grid format*. The exported lattice has a "Z" dimension of one. This means the lattice is two-dimensional. If an input lattice with a Z dimension of greater than one is exported, then **Voxler** prompts you to save multiple slices into separate .GRD files.
2. If a multi-slice uniform lattice is output to a .GRD file format, it is output as a series of slices in the Z direction. The **Select Slices** dialog appears and the user is prompted for the range of slices to output and a file name template that adds the slice number to each generated file name.

In **Surfer**, calculate volume:

1. Choose the **Grid | Volume** command to calculate the area above a certain level, and multiply it by the thickness of the slice to get the volume.
2. Repeat for each slice.

Chapter 16 - WellRenders

WellRender Module

The **Network | Graphics Output | WellRender** command adds a *WellRender* module. The *WellRender* module displays well traces along the well path generated from well data. Wells are displayed as tubes. Thickness and color can vary down the well. In addition, direction data and interval data can be displayed on the well.

In order for well traces to be displayed, the *WellData* module needs to contain valid well paths. For a well to have a valid path, one of the following needs to be true:

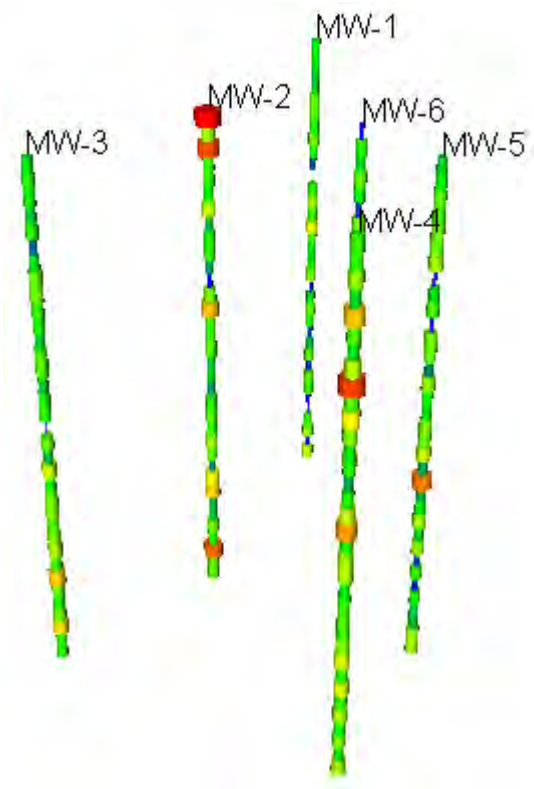
1. The data is imported with azimuth, dip, and measured depth.
2. The data is imported with directional survey information. A path is then computed on the **Well Paths** tab.
3. The data is imported with pre-computed X, Y, and Z values positioned along the well path.

Inputs

WellData is the input type for the *WellRender* module.

Outputs

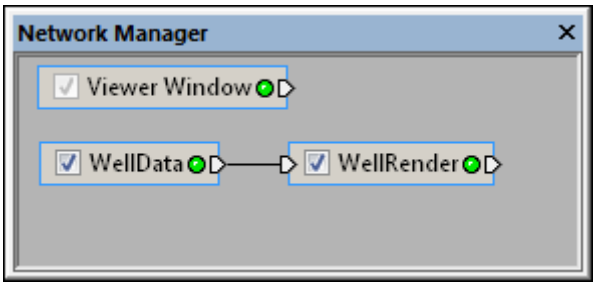
The *WellRender* module creates an output geometry. It may be connected to the *Graphics Output Modules*. An *Info Module* may also be connected to the output node.



Display wells with variable width and color to highlight areas of interest.

Properties

The *WellRender* module properties are described below.



*Select the WellRender module in the **Network Manager** to display its properties in the **Property Manager**.*

The *WellRender* module contains the following tabs in the **Property Manager:**

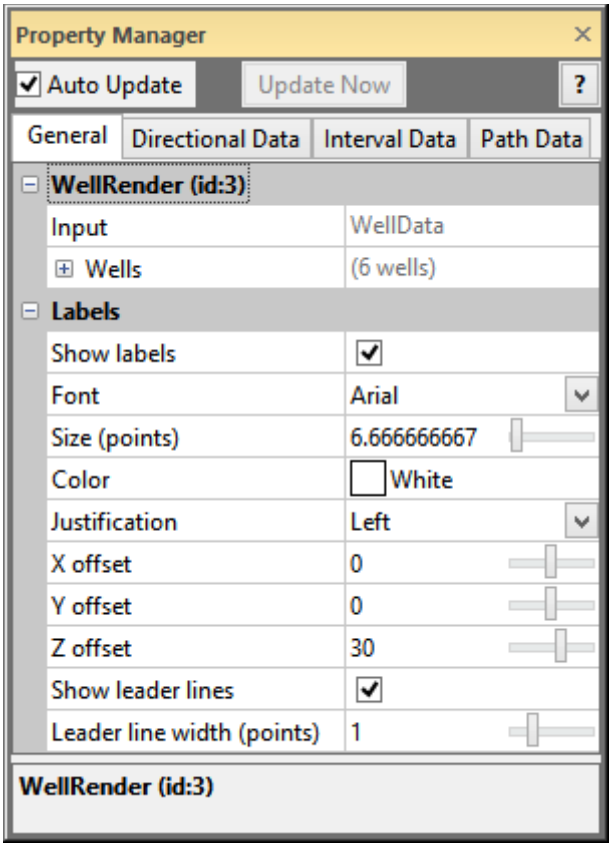
General

Directional Data

Interval Data

Path Data

General Options



Customize the WellRender properties.

Input

The *Input* property shows the source well data to which the module is connected. This option cannot be changed in the **Property Manager**, but can be changed in the **Network Manager** by changing the module input.

Wells

The *Wells* section contains a list of all wells that are displayed by the *WellRender*. Click on the ⊞ to expand the *Wells* section. To add a well to the list, click the **File | Import** command and add new well data to the existing *WellData* module. To turn off the display of a well in the list, uncheck the box next to the well name. To display a well, check the box next to the well name.

To delete a well permanently from the project, click on the *WellData* module. Right-click on the input file in the **Property Manager** and select **Delete [file name]** from the context menu. To hide a well, uncheck the well name on the **General** tab for the *WellRender* module.

Labels

The *Labels* section contains options for displaying labels for the well names. Click on the ⊞ to expand the section and set label properties.

Show Labels

Check the box next to *Show labels* to display well names. The well names are imported with the well data. Well names are always shown parallel to the view on the screen.

Font

The *Font* option sets the font used for writing the labels. All available TrueType fonts are listed as options. To change the font, click the current font name and select the desired font name from the list.

Size

The *Size (points)* option sets the size of the labels in points. Values range from 4 to 288 points. The larger the value, the larger the text appears on the screen. To change the value, highlight the existing value and type a new value or click and drag the  to the desired size.

Color

The *Color* option specifies the text color. To change the color, click the colored box and choose a new color from the color palette. If the basic colors in the palette do not meet your needs, click *Other* to create a custom color. The label line is drawn to match the color of the label.

Justification

Choose *Left*, *Right*, or *Center* justification for the label. The *Justification* is relative to the three-dimensional anchor point. *Left* anchors the left edge of the text to the anchor point, *Right* anchors the right edge of the text to the anchor point, and *Center* anchors the middle of the text to the anchor point.

X Offset

The *X offset* moves the screen position of the label away from the symbol along the X axis. Values are in pixels and range from -100 pixels to +100 pixels. From the top view for example, a negative value moves the label to the left, and a positive value moves the label to the right. The label stays aligned with the screen regardless of how it is moved. To change the offset, highlight the existing value and type a new value or click and drag the  to the desired value.

Y Offset

The *Y offset* moves the screen position of the label away from the symbol along the Y axis. Values are in pixels and range from -100 pixels to +100 pixels. From the top view for example, a negative value moves the label below the symbol, and a positive value moves the label above the symbol. The label stays aligned with the screen regardless of how it is moved. To change the offset, highlight the existing value and type a new value or click and drag the  to the desired value.

Z Offset

The *Z offset* moves the screen position of the label away from the symbol along the Z axis. Values are in pixels and range from -100 pixels to +100 pixels. From the top view for example, a negative value moves the label behind the symbol, and a positive value moves the label in front of the symbol. The label stays aligned with the screen regardless of how it is moved. To change the offset, highlight the existing value and type a new value or click and drag the  to the desired value.

Leader Lines

Check the box next to the *Show leader lines* option to display lines connecting the label and the top of the well.

Leader Line Width

Use the *Leader line width (points)* option to scale the thickness of the line that connects the well to the label. Values are in points and range from zero to four. To change the line thickness, highlight the existing value and type a new one, or drag the  to the desired value.

WellRender Module - Directional Data Page

The *WellRender* module **Directional Data** page contains the options to display fractures or oriented features along the well paths. All wells in the *WellRender* have the same properties applied. To open the **Directional Data** page, click on the *WellRender* module in the **Network Manager**. In the **Property Manager**, click on the **Directional Data** tab.

Directional data are oriented features, such as fractures. These features include azimuth and dip values down the well and are imported into **Voxler** as two logs. When the *WellRender* is displayed with directional data, a disk oriented in the direction of the dip and azimuth will be displayed at the appropriate depth. The disk size and color can be fixed or specified by another log. This can allow fracture attributes that are not the direction to also be shown on the *WellRender*.



This example shows 3 well traces, with directional disks approximately evenly spaced down the well.

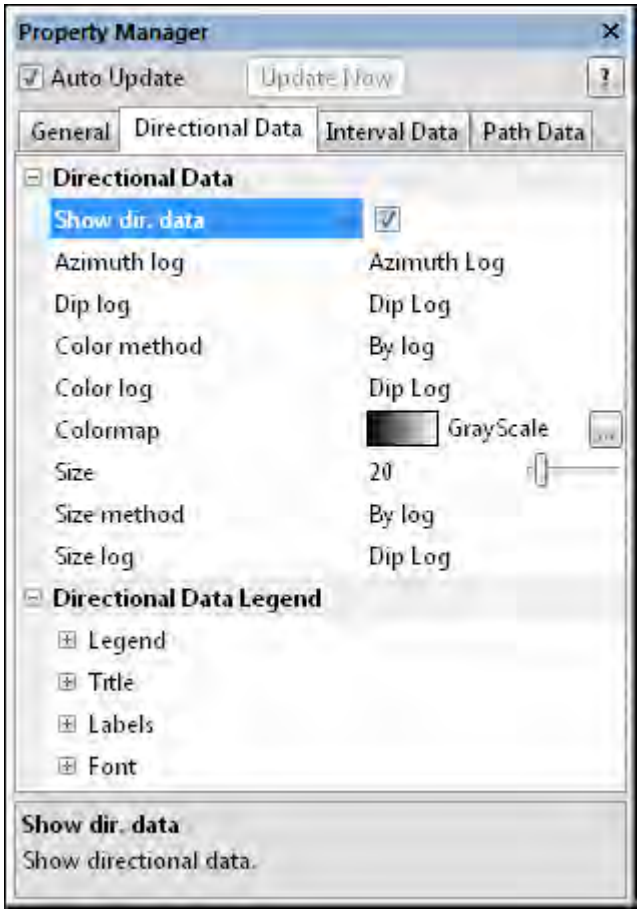
The *WellRender* module contains the following tabs in the **Property Manager:**

General

Directional Data

Interval Data

Path Data



*The **Directional Data** page controls the display of directional disks along the well path.*

Show Directional Data

Check the box next to the *Show dir. data* option to turn on the display of directional disks.

Azimuth Log

The *Azimuth log* is the log that is used as the source for the compass azimuth angle. This data is imported as a Log and contains various directional values in degrees, down-the-hole. To change the log, click on the existing option and select the desired log from the list.

Dip Log

The *Dip log* is the log that is used as the source for the directional dip data. This data is imported as a log and contains the offset from horizontal of the dipping feature. To change the log, click on the existing option and select the desired log from the list.

Color Method

The method used to generate the directional disk colors. The color may be set to a *Fixed* color or to *By log,* which matches a colormap with the *Color log*. Mapping the *Color log* through a Colormap can be useful when wanting to highlight some feature of the directional data with changing colors. To change the *Color method*, click on the existing option and select the desired option from the list.

When set to *Fixed*, the *Color* option becomes available. The *Color* is the color of the directional disk. To change the color, click on the existing color and select the desired color from the list. Alternatively, click the *Other* box to open the **Colors** dialog, where you can set custom colors.

When the *Color method* is set to *By log*, the *Colormap* option becomes available. To change the colormap used by the directional disk, click the existing color bar to the right of the *Colormap* command to select a different colormap from the list. Click on the desired colormap and the disks update. Alternatively, click the  button to open the **Colormap Editor** dialog. The **Colormap Editor** dialog allows you to create a custom colormap and to change the mapping of color to data values.

Color Log

The *Color log* is the log used to color the directional disks when the *Color method* is set to *By log*. To change the *Color log*, click on the existing option and select the desired log from the list.

Size

The *Size* option controls the relative size of the directional disks. When the *Size method* is set to *Fixed*, all disks are the specified *Size.* To change the *Size*, highlight the existing value and type a new value or click and drag the  to the desired value. Values range between zero and 200. Values are in points. The *Size* indicates the radius of the directional disks being drawn.

Size Method

The *Size method* is the method used to determine the size of the disks. Available options are *Fixed* and *By log.* When the *Size method* is set to *Fixed*, all directional disks appear with a radius specified by the *Size* value. When set to *By log*, the *Size log* option becomes available. The *Size log* is the log data that is used to size the directional disks. To change the *Size method*, click on the existing option and select the desired method from the list.

Size Log

The *Size log* is the log data that is used to size the directional disks. The smallest value in the log is sized to a value of 1/2 the *Size* value. The largest value in the log is sized to a value of 1.5 times the *Size* value. Disk sizes vary linearly between these values. To change the *Size log*, click on the existing option and select the desired log from the list.

Directional Data Legend

The options in the *Directional Data Legend* section are discussed on the **Legend** page.

WellRender Module - Interval Data Page

The *WellRender* module **Interval Data** page contains the options to display interval log values or from-to data along the well paths. All wells in the *WellRender* have the same properties applied. To open the **Interval Data** page, click on the *WellRender* module in the **Network Manager**. In the **Property Manager**, click on the **Interval Data** tab.



This example shows 3 well traces, with the size of the well tube determined by interval log data.

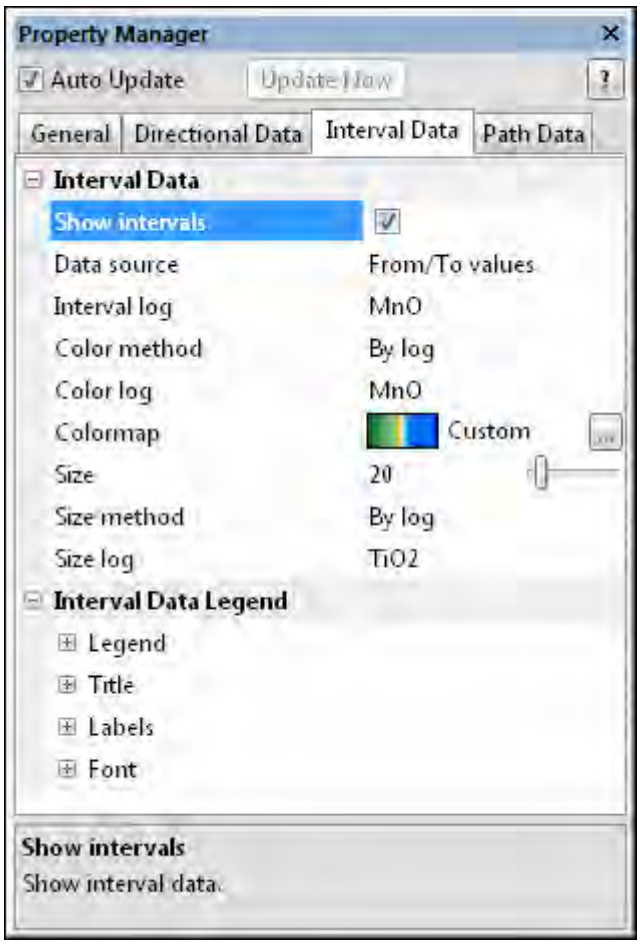
The *WellRender* module contains the following tabs in the **Property Manager:**

General

Directional Data

Interval Data

Path Data



*The **Interval Data** page controls the width of the well trace, when variable width is desired.*

Show Interval Data

Check the box next to the *Show intervals* option to turn on the display of variable width well traces.

Data Source

The *Data source* is the source for the interval values. Values can be determined from *Log values* or *From/To values*. To change the source, click on the existing value and select the desired value from the list.

Interval Log

The *Interval log* is the source log for determining the intervals. If the *Data source* is set to *Log values*, the values in the *Interval log* are used as depths. So, if the first value is 300, the first interval will be 300 units long. If the *Data source* is set to *From/To values*, an interval log with a *From Depth* and *To Depth* column needs to be imported. To change the *Interval log*, click on the existing option and select the desired log from the list.

Color Method

The *Color method* is the method used to generate the interval colors. The method may be set to a *Fixed* color or to *By log*, which matches a colormap with the *Color log*. Mapping the *Color log* through a Colormap can be useful when wanting to highlight some feature of the interval data with

changing colors. To change the *Color method*, click on the existing option and select the desired option from the list.

When set to *Fixed*, the *Color* option becomes available. The *Color* is the color of all of the intervals displayed. To change the color, click on the existing color and select the desired color from the list. Alternatively, click the *Other* box to open the **Colors** dialog, where you can set custom colors.

When the *Color method* is set to *By log*, the *Colormap* option becomes available. To change the colormap used by the intervals, click the existing color bar to the right of the *Colormap* command to select a different colormap from the list. Click on the desired colormap and the intervals update. Alternatively, click the  button to open the **Colormap Editor** dialog. The **Colormap Editor** dialog allows you to create a custom colormap and to change the mapping of color to data values.

Color Log

The *Color log* is the log used to color the intervals when the *Color method* is set to *By log*. To change the *Color log*, click on the existing option and select the desired log from the list.

Size

The *Size* option controls the relative width of the well trace for each interval. When the *Size method* is set to *Fixed*, all disks are the specified *Size.* To change the *Size*, highlight the existing value and type a new value or click and drag the  to the desired value. Values range between zero and 200. Values are in points. The *Size* indicates the radius of the well trace intervals being drawn.

Size Method

The *Size method* is the method used to determine the width of the well trace for the intervals. Available options are *Fixed* and *By log.* When the *Size method* is set to *Fixed*, all intervals appear with a radius specified by the *Size* value. When set to *By log*, the *Size log* option becomes available. The *Size log* is the log data that is used to size the width of the well trace for each interval. To change the *Size method*, click on the existing option and select the desired method from the list.

Size Log

The *Size log* is the log data that is used to size the width of the well trace for each interval. The smallest value in the log is scaled to a value of 1/2 the *Size* value. The largest value in the log is scaled to a value of 1.5 times the *Size* value. The width of the well trace varies linearly between these values. To change the *Size log*, click on the existing option and select the desired log from the list.

Interval Data Legend

The options in the *Interval Data Legend* section are discussed on the **Legend** page.

WellRender Module - Path Data Page

The *WellRender* module **Path Data** page contains the options to display continuous curve or log data along the well paths. All wells in the *WellRender* have the same properties applied. To open the **Path Data** page, click on the *WellRender* module in the **Network Manager**. In the **Property Manager**, click on the **Path Data** tab.

The *WellRender* module contains the following tabs in the **Property Manager:**

General

Directional Data

Interval Data

Path Data



*Set the well path properties on the **Path Data** tab in the Property Manager.*

Show Paths

Check the box next to the *Show paths* option to display the well trace along the well path.

Color Method

The *Color method* is the method used to generate the well log path. The color of the log may be set to a *Fixed* color or to *By log*, which matches a colormap with the log value. To change the *Color method*, click on the existing option and select the desired option from the list.

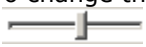
When set to *Fixed*, the *Color* option becomes available. The *Color* is the color of the well tube. To change the color, click on the existing color and select the desired color from the list. Alternatively, click the *Other* box to open the **Colors** dialog, where you can set custom colors.

When the *Color method* is set to *By log*, the *Colormap* option becomes available. To change the colormap used by the *WellRender*, click the existing color bar to the right of the *Colormap* command to select a different colormap from the list. Click on the desired colormap and the *WellRender* updates. Alternatively, click the  button to open the **Colormap Editor** dialog. The **Colormap Editor** dialog allows you to create a custom colormap and to change the mapping of color to data values.

Color Log

The *Color log* is the input log used to generate the color values. To change the log, click on the existing log name and select the desired log from the list.

Size

The *Size* is the relative size of the well trace tubes. The larger the value, the wider the well path is. Values range from zero, which displays no path, to 200. To change the path size, highlight the existing value and type a new value or click and drag the  to increase or decrease the well path.

Size Method

The *Size method* is the method used to determine the relative size of the path. Available options are *Fixed* and *By log.* When set to *Fixed,* the size is set by the *Size* option. When set to *By log,* the size is set with values from a log column. The *Size log* option becomes available, which contains the values that should be used to proportionally size the log. The largest size is determined by the *Size* value.

Legend

Show or hide the legend and change its properties. Click the ⊞ to expand the *Path Legend* section, where you can change the properties of the path legend. This option is only available when the *Color method* is set to *By log*.

How to Import and Display Wells

Voxler can import well data from numerous sources, including from LAS files. Typically the well data (logs) will be imported from one file and the physical location of the well (collars) will be imported from another file. Often, trajectory data indicating the direction of the well trace will also be imported, when the well is not assumed to be vertical.

Primary Data Requirements

In order for well traces to be displayed, the *WellData* module needs to contain valid well paths. For a well to have a valid path, one of the following needs to be true:

1. The data is imported with azimuth, dip, and measured depth.
2. The data is imported with directions survey information. A path is then computed automatically.
3. The data is imported with pre-computed X, Y, and Z values down-the-hole.

Log ASCII Standard LAS files and non-tabular format data files will be imported directly into the *WellData* module. When well data is in a tabular file format, a *Data Source* module will be created upon import.

Importing Well Collar Data

For this example, well collars, trajectory data, and log data are on three tabs of an Excel spreadsheet. Each tab is imported separately, with the following steps.

1. Click the **File | Import** command.
2. In the **Import** dialog, select the well collar file. For this example, select the *SampleWellData.xlsx* file from the Samples directory. By default, the Samples directory is located at C:\Program Files\Golden Software\Voxler 4\Samples. Click *Open*.
3. In the **XLSX Import Options** dialog, select the *Collars* table and click *OK*.
4. A Data Source module titled SampleWellData.xlsx - Collars is created.
5. Specify the following options in the **Property Manager**:
    a. Change the Output type to Wells
    b. Change the *Sheet type* to *Collars*. This step is not required. It simplifies the **Property Manager** display, showing only the properties applicable to a worksheet containing collar data.
    c. Set Well Name (ID) to Column A: ID
    d. Set Top X (Easting) to Column B: Easting
    e. Set Top Y (Northing) to Column C: Northing
    f. Set Top Z (Elevation) to Column D: Elevation
    g. Set Azimuth to Column E: Azimuth
    h. Set Vertical Direction to Dip
    i. Set Dip to Column F: Dip
    j. Set Total Depth to Column G: Depth
6. Click the **Network | Well | WellData** command to create a *WellData* module.
7. Verify that the *SampleWellData.xlsx - Collars* module output is connected to one of the *WellData* module inputs.

Importing Trajectory Data

At this point, all you have imported is the top location of the six wells. To import the well trajectories (traces), follow the steps in this section.

1. Click the **File | Import** command or click the *Add Data* button in the **Property Manager Inputs** page for the *WellData* module.
2. In the **Import** dialog, select the well file. Select the *SampleWellData.xlsx* file again and click *Open*.
3. In the **XLSX Import Options** dialog, select the *Trajectories* table and click *OK*.
4. A Data Source module titled SampleWellData.xlsx - Trajectories is created.
5. Specify the following options in the **Property Manager**:
    a. Change the Output type to Wells
    b. Change the *Sheet type* to *Directional Survey*. This step is not required, but it is recommended.
    c. Set Well Name (ID) to Column A: ID
    d. Set Azimuth to Column C: Azimuth
    e. Set Vertical direction to Inclination
    f. Set Inclination to Column D: Inclination
    g. Set Measured Depth to Column B: MD
6. Connect the *SampleWellData.xlsx - Trajectories* module output to one of the *WellData* module inputs.

The information is imported, and the actual X, Y, and Z values for the path of the well are calculated automatically by the program.

Displaying the Wells

At this point, we can display the wells by clicking the **Network | Graphic Output | WellRender** command. The well traces are then displayed.



*The initial well traces are displayed in the **Viewer** window.*

Importing Log Data

The well appearance can be altered by importing additional data.

1. Click on the *WellData* module in the **Network Manager**.
2. Click the **File | Import** command or click the *Add Data* button on the **Inputs** page of the **Property Manager**.
3. In the **Import** dialog, select the well file. Select the SampleWellData.xlsx again and click *Open*.
4. In the **XLSX Import Options** dialog, select the *Samples* table and click *OK*.
5. A Data Source module titled SampleWellData.xlsx - Samples is created.
6. Specify the following options in the **Property Manager**:
    a. Change the Output type to Wells
    b. Change the *Sheet type* to *From / To Logs*. This step is not required, but it is recommended.
    c. Set Well Name (ID) to Column A: ID
    d. Set From to Column B: From
    e. Set To to Column C: To
    f. Set Log columns to 2
    g. Set Log-1 to Column D: TiO2
    h. Set Log-2 to Column E: MnO
7. Connect the *SampleWellData.xlsx - Samples* module output to one of the *WellData* module inputs.

Displaying Log Data on the Wells

Once the log data is imported, the well appearance can be modified to display the logs using these steps:

1. Click on the *WellRender* module in the **Network Manager**.
2. Click on the Interval Data tab in the Property Manager.
3. Check the box next to *Show intervals* to add variable width log information.
4. Change the *Interval log* to MnO to use the MnO log *To Depth* and *From Depth* as the interval definition.
5. Change the *Color method* to *By log* so the colors of the intervals vary with log data.
6. Set the *Color log* to MnO.
7. Change the *Colormap* to *Rainbow* to display different colors along the length. The colors are determined by mapping the values in the MnO log to the colors in the *Rainbow* colormap.

8. Set the Size method to By log.
9. Set the *Size log* to TiO2. The width of the log is now determined by the data in the TiO2 log.



The wells are now displayed as tubes of variable width, based on the data in the TiO2 log, and color, based on the data in the MnO log.

Gridding and Mapping WellData

Once *WellData* is imported, it can be displayed as a *WellRender* or it can be extracted into X, Y, and Z values and gridded for display as a *VolRender*, *Isosurface*, *HeightField*, or any other type of lattice-based map.

This section starts with an already existing *WellData* module. To import the data used for these sections, follow the steps on the *Importing WellData and Displaying Wells* page.

Extracting WellData

Before anything can be done with the log information, the wells need to be extracted to a data file.

1. Click on the *WellData* module in the **Network Manager**.
2. Click the **Network | Computational | ExtractPoints** command to attach an *ExtractPoints* module to the data.
3. Click on the *ExtractPoints* module in the **Network Manager**.
4. In the **Property Manager**, change the *Output components* to 2, so that both the MnO and TiO2 can be extracted.
5. Change the *Component [2]* to the TiO2 log.

Gridding WellData

The extracted data can be gridded using a *Gridder* module.

1. Click the **Network | Computational | Gridder** command to attach a *Gridder* module to the extracted points.
2. In the **Property Manager**, set the *input component* to 2 to grid the TiO2 data.
3. Because the data is more likely to be relevant to other values in the same Z plane, an anisotropic search method or gridding method may need to be set. For now, we'll use the default isotropic settings.
4. Click the *Begin Gridding* button to grid the TiO2 data.

Mapping WellData

Once the data is gridded, any lattice based maps can be added. Click the **Network | Graphics Output | Isosurface** to add an *Isosurface* module. The *Isosurface* properties can be changed to better connect the wells.

Modifying Wells

Wells are imported into *WellData* modules. Once the wells exist in the *WellData* module, the wells can be displayed with a *WellRender* module or the data can be extracted with an *ExtractPoints* module. The below list a few of the features of the wells that can be changed.

Importing Wells

To import wells in a tabular format data file:

1. Click the **File | Import** command.
2. In the **Import** dialog, select the well data file and click *Open*.
3. In the **Property Manager**, specify the *Output type*, *Sheet type*, and data columns.
4. If necessary, create a *WellData* module by clicking *Well* | *WellData* in the **Module Manager**.
5. Connect the *Data Source* module to the *WellData* module.

To import wells in a non-tabular format:

1. Click the **File | Import** command.
2. In the **Import** dialog, select the well data file and click *Open*.
3. Specify any file import options in the file type specific **Import Options** dialog.
4. In the **Network Manager**, you can see that a *WellData* module was created. Click on the *WellData* module to select it. In the **Property Manager**, you can see the wells that were imported. If you click on the ⊞ next to any well, you can see the information that was imported for that well.

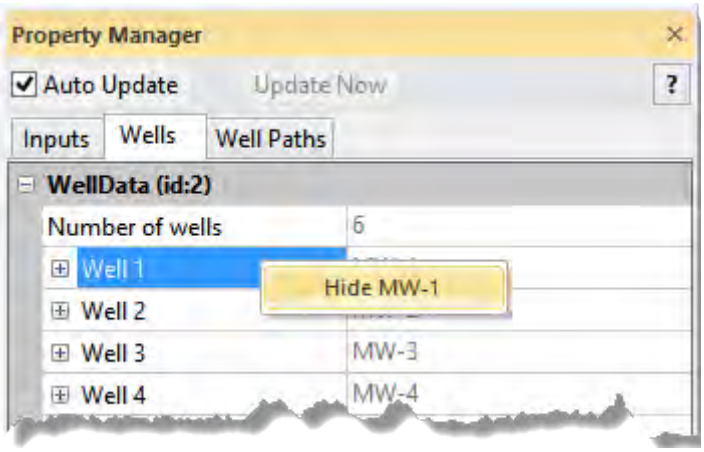
Importing Additional Wells

If an existing *WellData* module exists and your files are in a tabular format, you can use the steps in the *Importing Wells* section above to import additional wells.

If the data file is in a non-tabular format, such as LAS, check the *Add to existing WellData module* check box in the **Import Options** dialog. Alternatively, click on the *WellData* module. In the **Property Manager**, click the *Add Data* button.

Hiding Wells

To hide wells, click on the *WellData* module in the **Network Manager**. In the **Property Manager**, click on the well to be hidden. Right-click and select **Hide [well name]** from the menu. The well is disabled from the *WellData* module and hidden in the viewer window. To show a hidden well, right-click on the well and select **Show [well name]** from the menu.



Select the Hide MW-1 text to hide the MW-1 well in the viewer window.

Alternatively you can hide wells in the *WellRender* module properties. Click on the *WellRender* module. In the **Property Manager**, click on the **General** tab. In the *Wells* section, uncheck the box next to the well you don't want to display.

Changing Wells

To change well information, update the well information in the **Voxler** worksheet. If the well data is imported into a *Data Source* module, click the *Edit Worksheet* button in the **Property Manager** to update the well data in the worksheet. Changes made to the well information are automatically visible when you return to the **Viewer** window.

If the well data was imported directly into the *WellData* module, update the well information in the original data file. Next, in **Voxler**, use the steps in the *Importing Wells* section above to import the changed data. **Voxler** will automatically change the data in the *WellData* node to the new data.

Displaying Wells

To display wells, add a *WellRender* module to the *WellData* module. Click on the *WellData* module to select it. Click the **Network | Graphics Output | WellRender** command. The default display for the wells is shown.

Setting Well Properties

Well properties are set in the *WellRender* module. Click on the *WellRender* module in the **Network Manager** to select it. Properties are changed in the **Property Manager**.

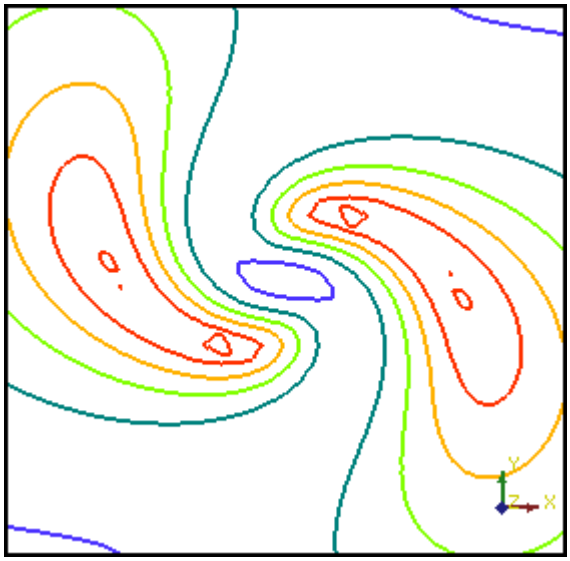
Chapter 17 - Views and Altering the View Style

This chapter discusses the various **View** menu commands in **Voxler**, and how the **View** menu commands are used to change the display of the **Viewer** window.

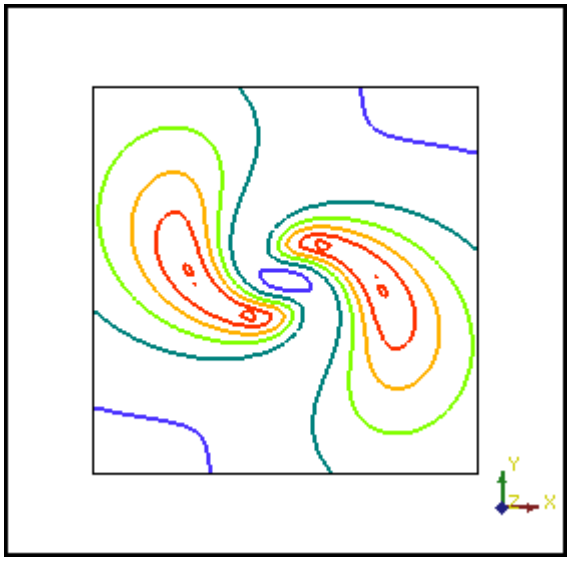
Fit to Window

Click the **View | Fit to Window** command, press the F4 key, or click the  button to zoom the camera until the scene approximately fills the window extents. A varying amount of unused space is typically left around the edge of the window. This is due to calculations using three-dimensional bounding boxes that do not always fit the generated geometry tightly.

This command is useful when the geometry is no longer visible in the **Viewer** window, perhaps because new objects have been added or existing objects have been modified. The **View | Fit to Window** command allows you to view all of the items in the **Viewer** window.



*The **Viewer** window before the **View | Fit to Window** command is executed.*



*The **Viewer** window after the **View | Fit to Window** command is executed.*

Trackball

Click the **View | Trackball** command or click the  button to rotate the view in the **Viewer** window. This command uses a virtual trackball interface, which can be thought of as an invisible sphere centered in the middle of the **Viewer** window. You can rotate the sphere and graphics in the window by dragging the mouse on this invisible sphere.

Move the mouse cursor to the **Viewer** window. Hold down the left mouse button and drag the mouse to rotate the view. To keep the view stationary, stop moving the mouse prior to releasing the left mouse button.

Spin

To spin the view, release the left mouse button prior to stopping the mouse. The faster the mouse is moved, the faster the graphics spin. Graphics are rotated in the direction the mouse is moving at the time the mouse button is released. Click the mouse in the **Viewer** window without moving it to stop the spinning.

Dragger

You can add a dragger to some modules for use in rotating the associated geometry.

View

The view can be changed in a number of ways while in trackball mode. Hold down the CTRL key and the left mouse button and then move the mouse up or down to zoom the view in or out. Hold down the SHIFT key and the left mouse button and then move the mouse around to pan the view. Finally, you can scroll the mouse wheel up or down to zoom the view in or out. The mouse controls can be customized on the **Mouse** page of the **Tools | Options** dialog.

Changing the view using the trackball cannot be undone.

Zoom Realtime

The **View | Zoom Realtime** command zooms in and out as the mouse is dragged up and down or when the mouse wheel is moved.

To use this feature, choose the **View | Zoom Realtime** command or click the  button to enter zoom mode. The cursor changes to  when in zoom realtime mode. Left-click in the **Viewer** window, hold down the left mouse button and then drag it up or down in the **Viewer** window to zoom in or out. As you drag the mouse up, the screen is zoomed in. As you drag the mouse down, the screen is zoomed out. You can also use the mouse wheel to zoom in or out. Move the wheel away from you to zoom in and toward you to zoom out.

Pan

Hold down the SHIFT key and the left mouse button and then drag the mouse around to pan the view. Alternatively, use the **View | Pan** command to pan the **Viewer** window.

Customize

These mouse controls can be customized on the **Mouse** page of the **Tools | Options** dialog.

Exit Zoom Realtime

To exit the zoom realtime mode, press the ESC key on the keyboard. The trackball mode becomes active.

Undo

Changing the zoom level cannot be undone.

Pan

Click the **View | Pan** command or click the  button to enter pan mode. The cursor changes to  when in pan mode. The **Pan** command allows you to move your location in the **Viewer** window while retaining the current level of magnification.

Hold down the left mouse button in the **Viewer** window and drag the mouse up, down, left, or right to pan the scene in the desired direction. Hold down the CTRL key and the left mouse button and then move the mouse up or down to zoom the view in or out. Alternatively, use the **View | Zoom Realtime** command.

Customize

These mouse controls can be customized on the **Mouse** page of the **Options** dialog.

Exit Pan

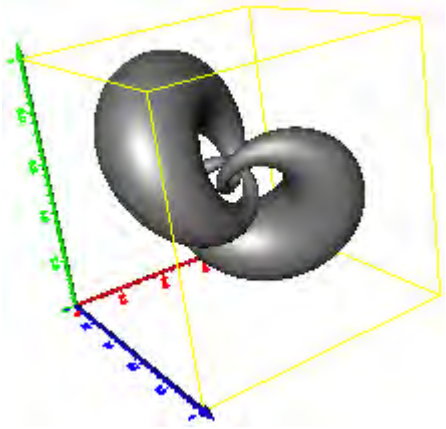
To exit the pan mode, press the ESC key on the keyboard. The trackball mode becomes active.

Undo

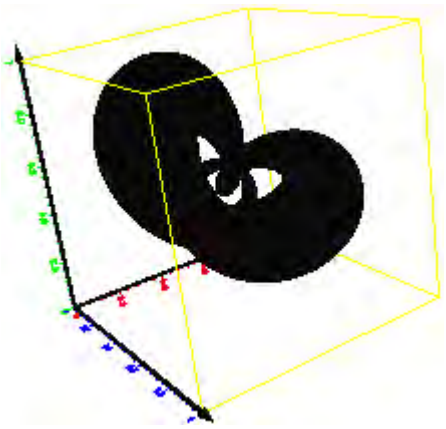
Changing the view with the pan command cannot be undone.

Headlight

Click the **View | Headlight** command or  button to turn the camera headlight on or off. The headlight is a directional light positioned at the camera which shines in the same direction that the camera is pointing. This light is usually left on unless additional light modules have been placed in the scene. Turning off all lights can result in a blank window.



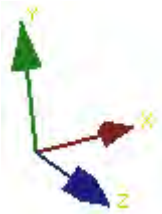
In this example, the headlight is turned on. This is the default setting.



In this example, the headlight is turned off.

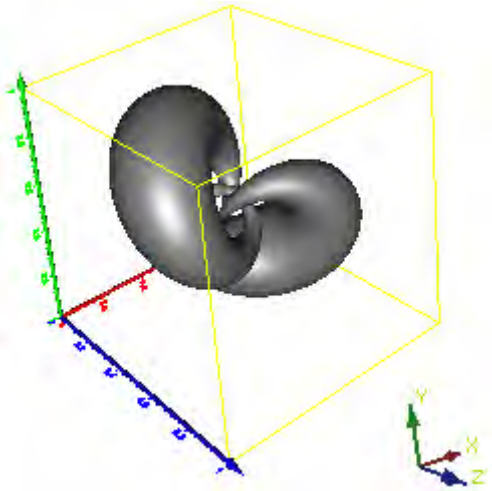
World Axis Triad

Click the **View | World Axis Triad** command or  button to turn the world axis triad on or off. This feature is located in the lower right corner of the **Viewer** window. The axis triad is informational only. Turning it off does not affect the network or geometry in any way.



This is the world axis triad.

The world axis triad is a depiction of the X, Y, and Z directions that shows the **Viewer** window camera orientation. The world axis triad draws the X axis in red, the Y axis in green, and the Z axis in blue.



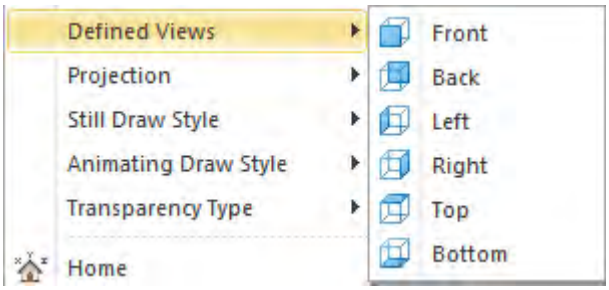
This is an example of the world axis triad and the geometry it is reflecting.

Customize Axis Colors

Customize the axis triad colors on the **Colors** page of the **Options** dialog. Choose the **Tools | Options** command to open the **Options** dialog.

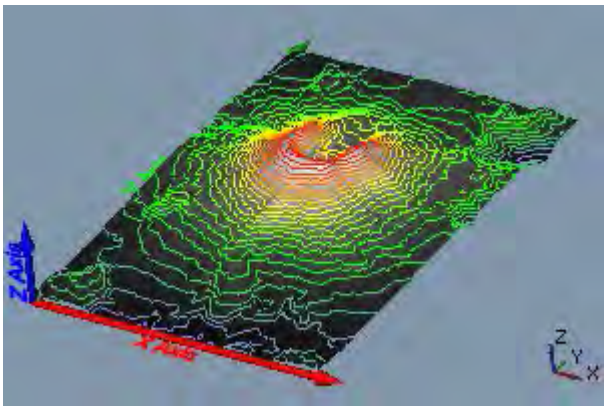
Defined Views

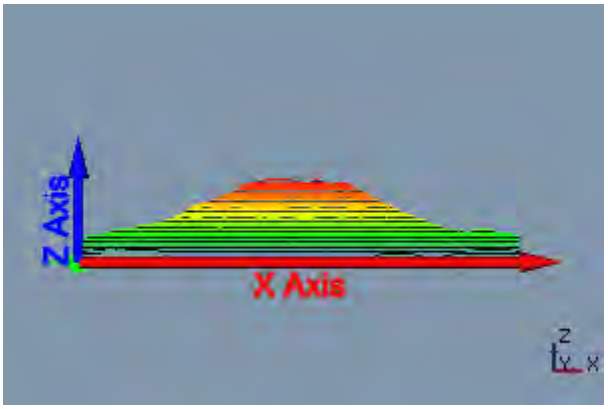
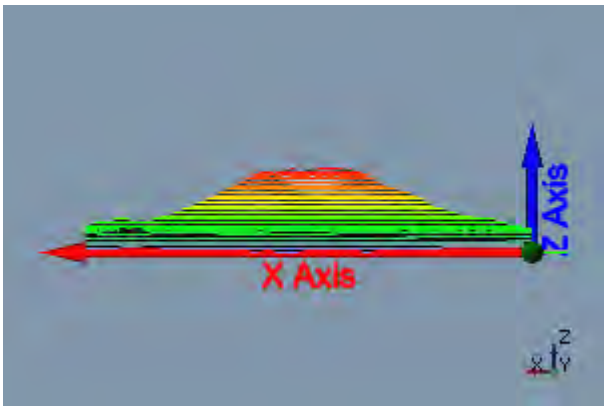
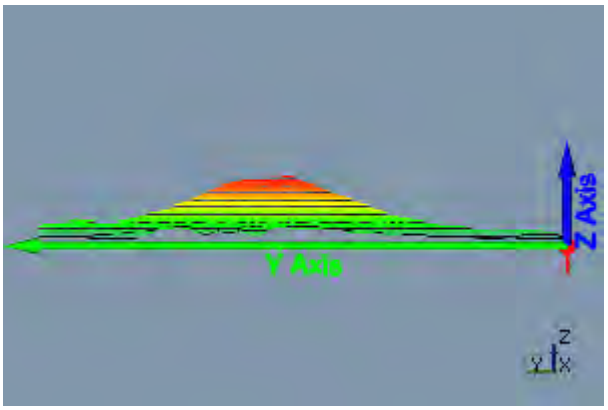
Click the **View | Defined Views** command or click the  button to select the direction from which the scene is viewed in the **Viewer** window. Choices include *Front*, *Back*, *Left*, *Right*, *Top*, and *Bottom*.

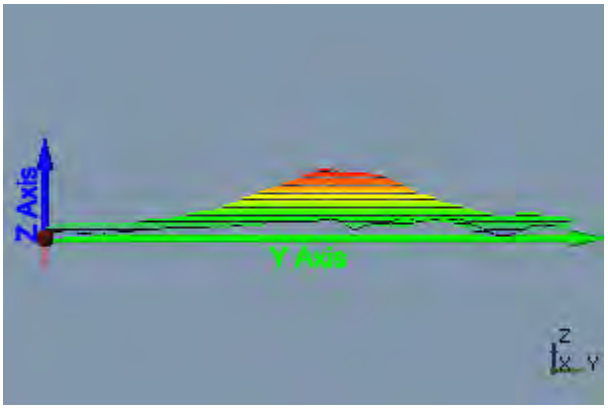
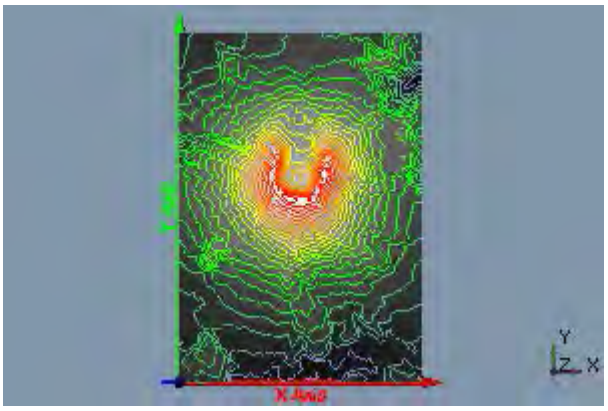
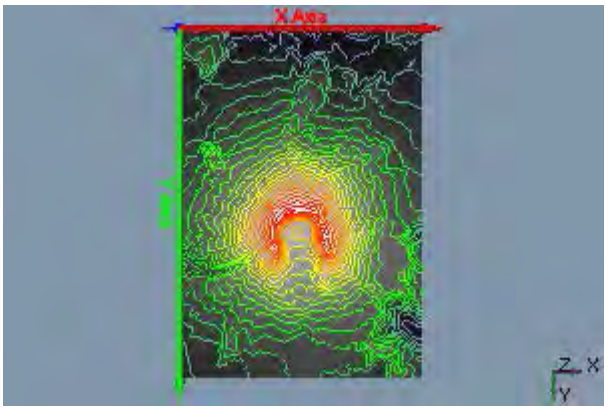


*Choose the **View | Defined Views** command to select one of the choices.*

The sample file *Helens (ContourMap).VOXB* will be used to display the different defined views. Notice the world axis triad differences in each example. The axes are also displayed. The red axis is the X Axis, the green axis is the Y Axis, and the blue axis is the Z Axis. Note the arrow direction for the axes.

Defined View	Example
None	 *The sample file Helens (ContourMap).voxb with no defined view.*

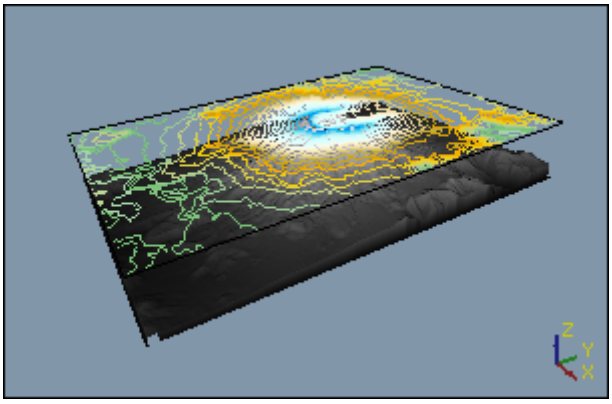
Front	 *The sample file Helens (ContourMap).voxb with a Front defined view.*
Back	 *The sample file Helens (ContourMap).voxb with a Back defined view.*
Left	 *The sample file Helens (ContourMap).voxb with a Left defined view.*

Right	 *The sample file Helens (ContourMap).voxb with a Right defined view.*
Top	 *The sample file Helens (ContourMap).voxb with a Top defined view.*
Bottom	 *The sample file Helens (ContourMap).voxb with a Bottom defined view.*

Perspective Projection

The projection affects how the three-dimensional scene is drawn in the **Viewer** window.

Click the **View | Projection | Perspective** command or click the  button to enable perspective projection, which is the **Voxler** default camera. The most distinguishing characteristic of perspective projection is foreshortening: the farther an object is from the camera, the smaller it appears in the final image. Perspective projection emulates the human eye so scenes appear more realistic or lifelike— larger when viewed closely, smaller when viewed from a distance.

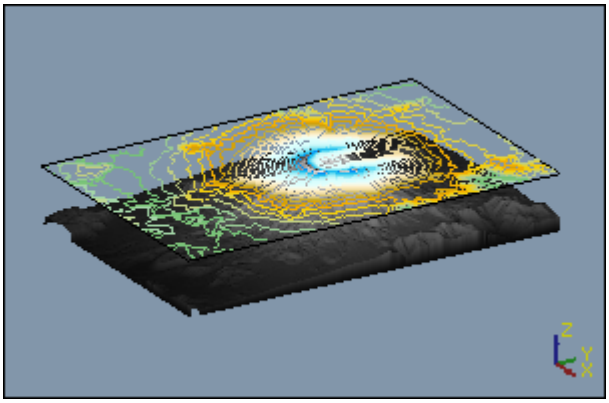


The sample file Helens (ContourMap).voxb with a Perspective projection.

Orthographic Projection

The projection affects how the three-dimensional scene is drawn in the **Viewer** window.

Choose the **View | Projection | Orthographic** command or click the  button to enable orthographic projection. The orthographic projection produces a parallel projection with no distortion for distance. As a result, it is sometimes difficult to determine how far an object is from you when viewing it in orthographic projection. This view is useful, however, when you need to measure distances or angles, or exactly align objects in three-dimensional space.



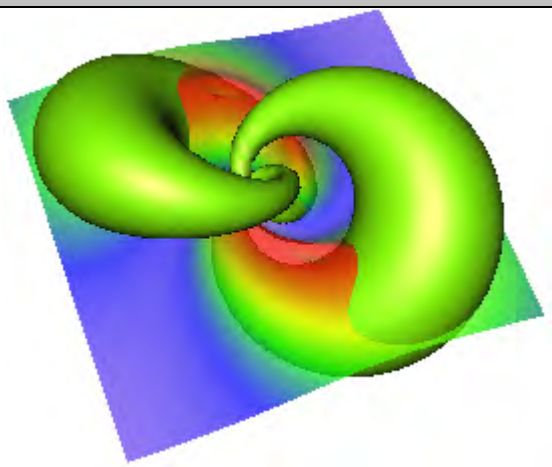
The sample file Helens (ContourMap).voxb with an Orthographic projection.

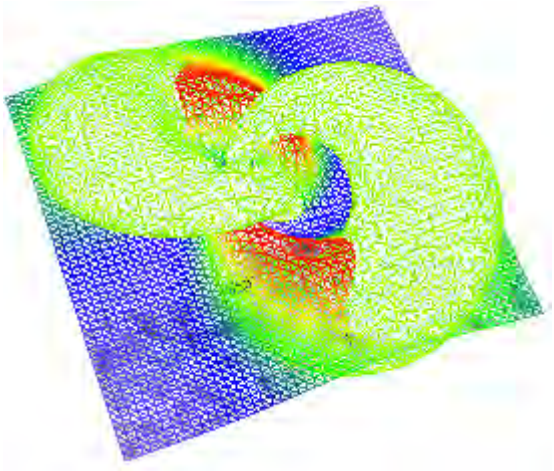
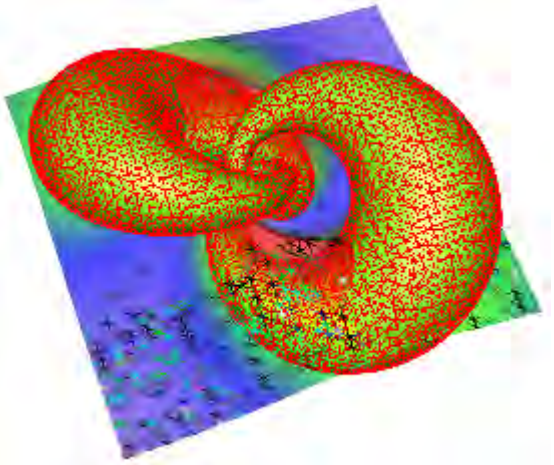
Still Draw Style

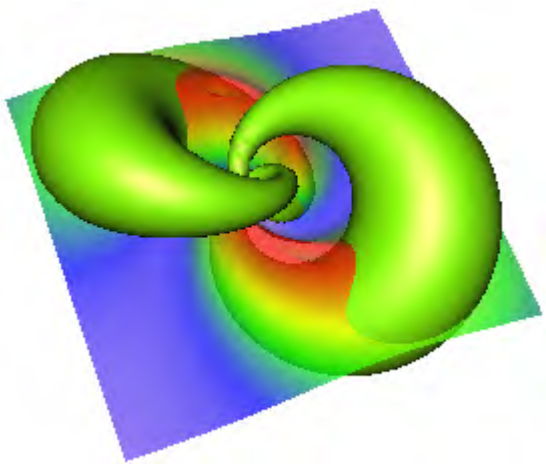
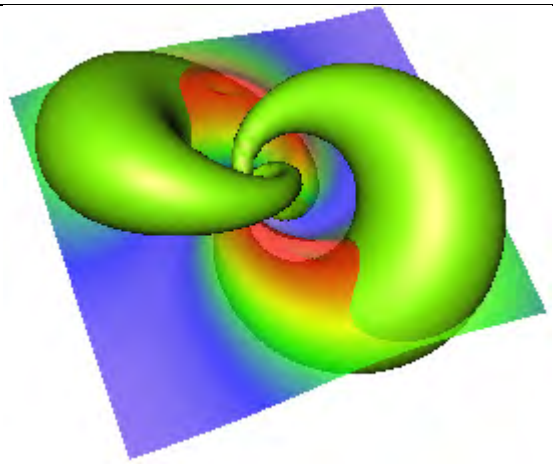
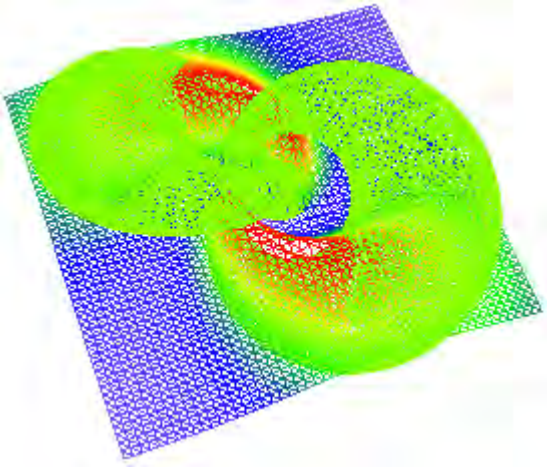
Click the **View | Still Draw Style** command to select a still draw style, the drawing style used to draw non-spinning graphics. Click the **View | Animating Draw Style** command to select the drawing style used to draw spinning graphics.

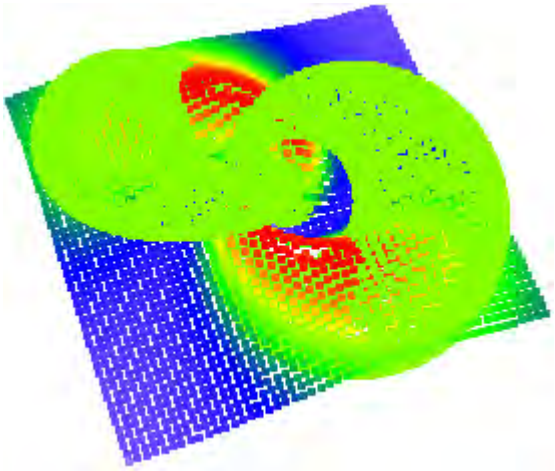
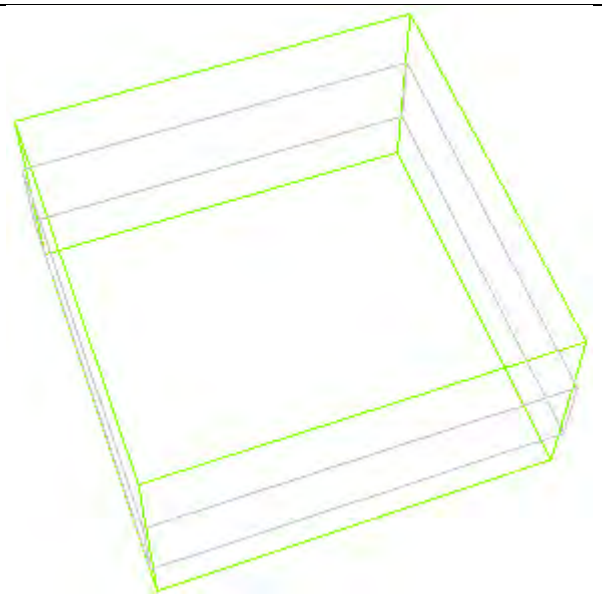
The still drawing style temporarily disables graphics and enables **Voxler** to render the geometry more quickly.

The different still draw styles are described below.

Still Draw Style	Description	Image Example
As Is	Render the geometry as specified with its original style. This is the default selection.	

Hidden Line	Render the scene in "hidden line" mode; that is, as wireframe with no see-through. Note: this is a more memory intensive way to render an image, as the scene must be rendered twice to achieve the effect of hiding lines behind the invisible geometry.	
Wireframe Overlay	Render the scene as normal, but overlay a set of lines showing the contours of all polygons.	

No Texture	Render the scene without textures.	
Low Resolution	Render all complex shape types with low complexity to improve rendering performance. The term "complex" in this context refers to spheres, cones, cylinders, NURBS surfaces, and others which are tessellated to polygons before being rendered.	
Wireframe	Render all polygon geometry in wireframe mode. All polygons are drawn with lines only (no fill).	

Points	Render only the vertex positions of the geometry.	
Bounding Box (no depth)	Render the scene's bounding boxes instead of rendering the full geometry. This mode is an efficient way to optimize rendering performance for scenes with high primitive counts while moving the camera about. Displaying only the bounding box while in motion may be useful to rotate to a new location.	

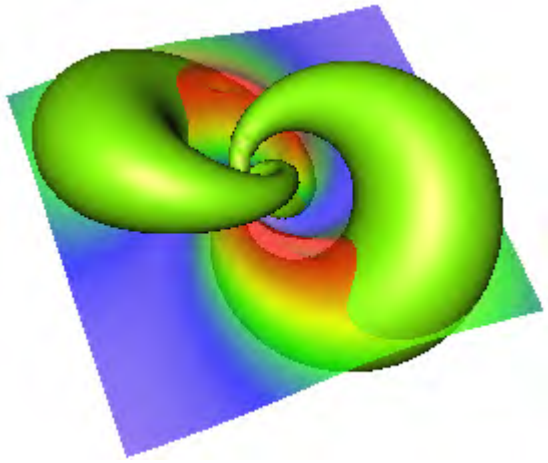
Animating Draw Style

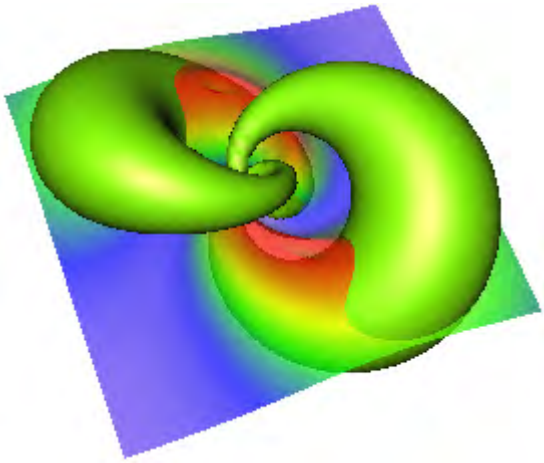
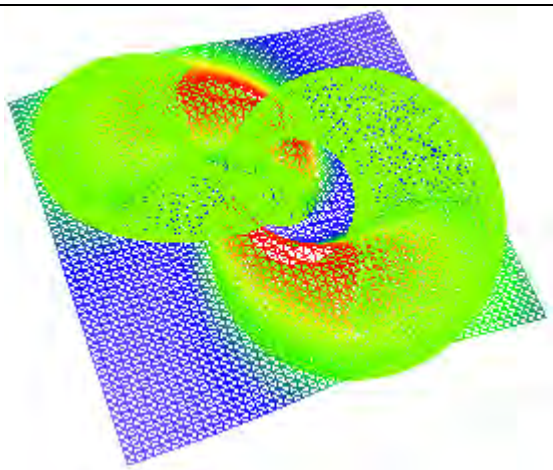
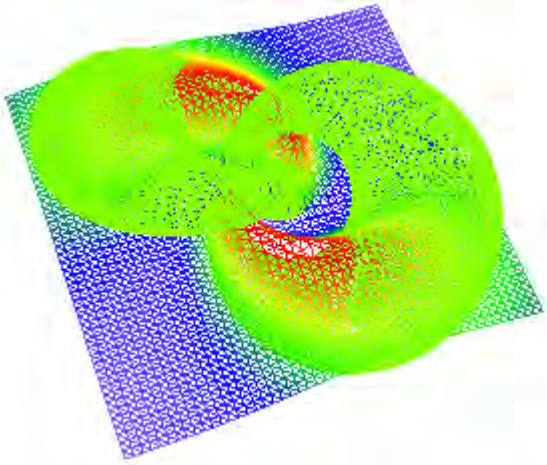
Click the **View | Animating Draw Style** command to select an animated drawing style, the drawing style used to draw spinning graphics. Choose the **View | Still Draw Style** command to select the drawing style used to draw non-spinning graphics.

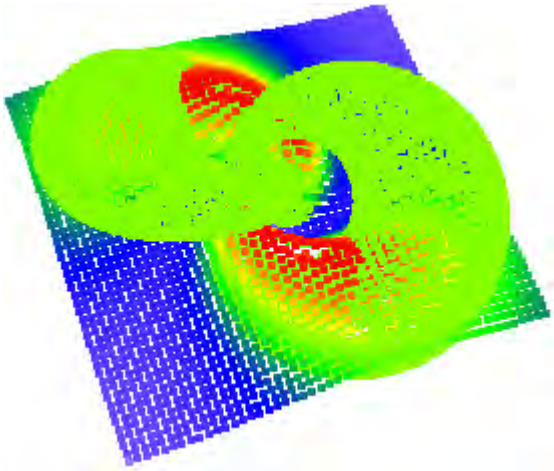
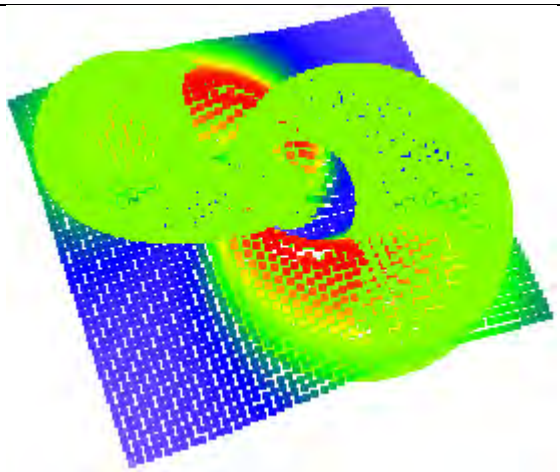
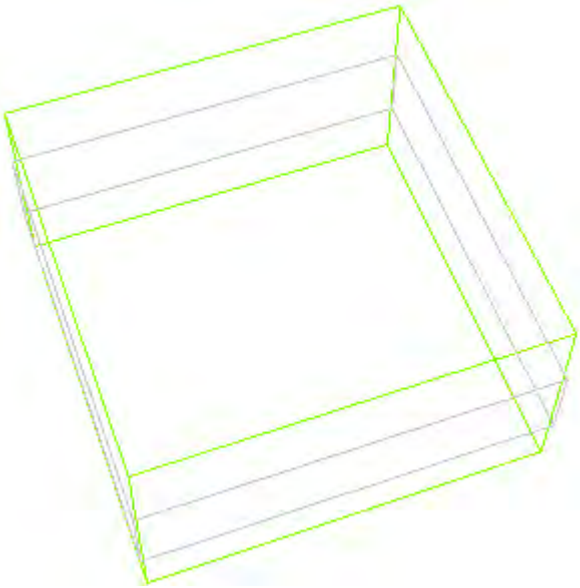
The type of animated drawing style selected may affect the quality and size of a captured video. Adjust the animated drawing style before choosing the **Actions | Capture Video** command.

Animating Drawing Styles

The different animating draw styles are described below.

Animating Draw Style	Description	Image Example
Same as Still	Render the scene in the same manner as the scene with a still camera. This is the default selection.	Depends on the **Still Draw Style** selection.
No Texture	Render the scene without textures.	

Low Resolution	Render all complex shape types with low complexity to improve rendering performance. The term "complex" in this context refers to spheres, cones, cylinders, NURBS surfaces, and others which are tessellated to polygons before being rendered.	
Wireframe	Render all polygon geometry in wireframe mode. All polygons are drawn with lines only (no fill).	
Low Res Wireframe (no depth)	Render the scene as a wireframe. All polygons are drawn with lines only (no fill). No correction for depth is made.	

Points	Render only the vertex positions of the geometry.	
Low Res Points (no depth)	Render the scene as vertex points without correcting for depth.	
Bounding Box (no depth)	Render the scene's bounding boxes instead of rendering the full geometry. Note: Setting this mode for the Animating Draw Style is a very efficient way to optimize rendering performance for scenes with high primitive counts while moving the camera position.	

Transparency Type

Click the **View | Transparency Type** command to specify how transparency is rendered. Some settings provide faster rendering while others give better quality.

Default (Fastest) Transparency Type

The faster *Blend* method is used as the default. If the default transparency method does not work with your plot, try the other methods by choosing the **View | Transparency Type** command or right-clicking in the **Viewer** window and selecting **Transparency Type** menu item.

Best Quality Transparency Type

The best quality method is *Sorted Object, Sorted Triangle Blend.* This method accumulates all the triangles (which requires a lot of memory), then sorts them according to distance from the viewer (which requires a lot of time). Although this method uses the most memory and time, it should be used when the quality of the image is the most important factor.

Color Blending

Color blending is the process of mixing two colors together to produce a third color.

The first color is the *source color*. This is the new color being added. The second color is the *destination color*. This is the color that already exists. Each color has a separate blend factor that determines how much of each color is combined into the final product. Once the source and destination colors have been multiplied by their blend factors, the results are combined according to the specified blend function.

*(source * source blend factor*) (*blend function*) (*destination * destination blend factor*)

Alpha blending uses the alpha channel of the source color to create a transparency effect so that the destination color appears through the source color.

Smooth Rotation

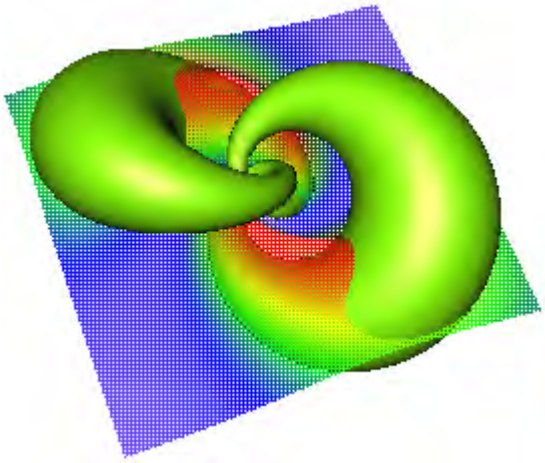
To ensure smooth rotation of the graphic, try multiple *Transparency Type* selections. Some selections may create a jerky rotation behavior due to a large number of triangles in relation to your machine's capabilities. To see the number of triangles in an object, connect an *Info* module.

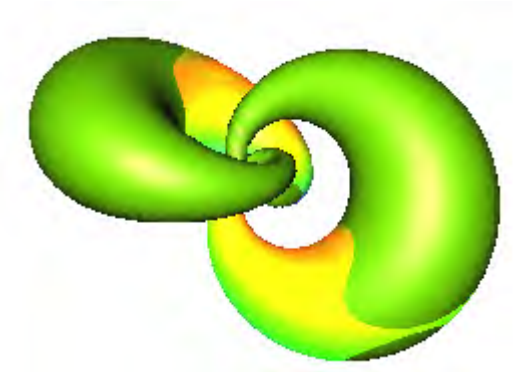
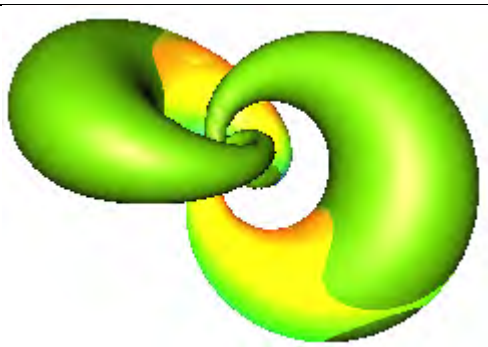
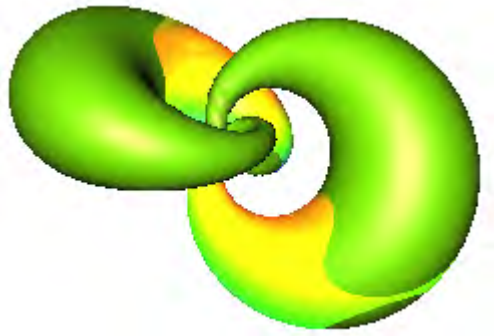
Notes

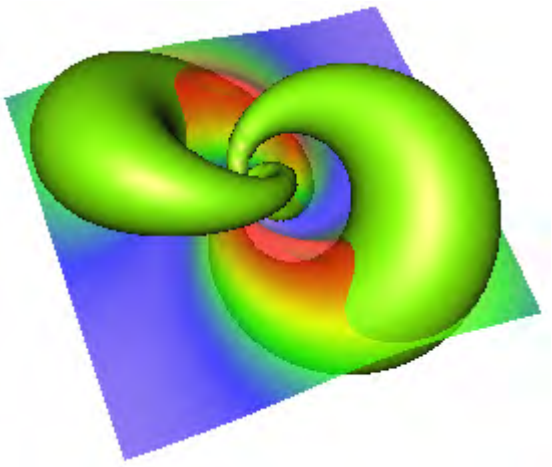
Correct transparency rendering of multiple polygons is not possible unless polygons are rendered in sorted order and polygons are non-intersecting. Polygon sorting can be very time consuming.

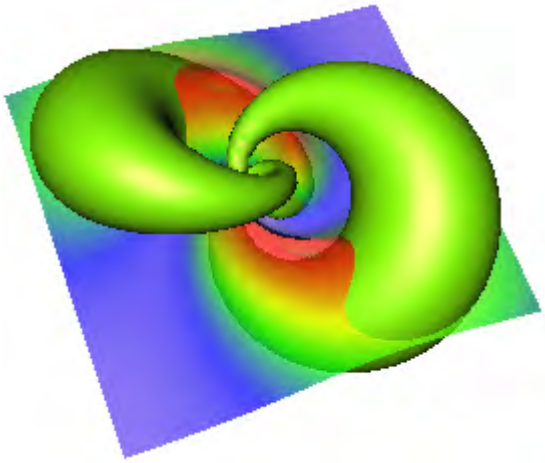
Transparency Types

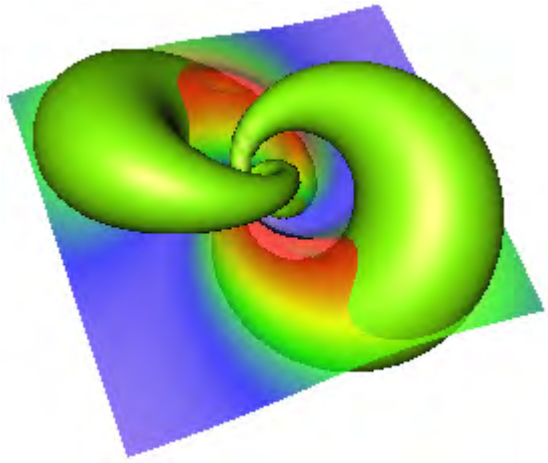
The different transparency types are described below. In the image examples below, the *HeightField* has 75% opacity and the *Isosurface* of the has 100% opacity.

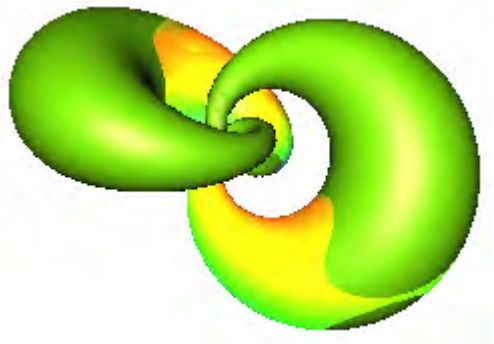
Transparency Type	Description	Image Example
Screen Door	Render transparent triangles with a dither pattern. With most video cards, this is a fast but lower quality transparency mode. This mode stands apart from the other transparency types in that it always renders the transparent parts of the scene with intact internal depth ordering of objects/polygons. Polygons rendered with only transparent textures are not shown as transparent when using this mode, as the *Screen Door* type works on polygons rather than pixels.	

Add	Render transparent objects with additive alpha blending. Additive blending is primarily used to create special transparency effects. The new pixel color is calculated as the current pixel color plus the source pixel color multiplied by the source pixel alpha value.	
Delayed Add	Render transparent objects with additive alpha blending in a second rendering pass with depth buffer updates disabled.	
Sorted Object Add	Render transparent objects with additional alpha blending. Opaque objects are rendered first; transparent objects are rendered back to front with z-buffer updates disabled.	

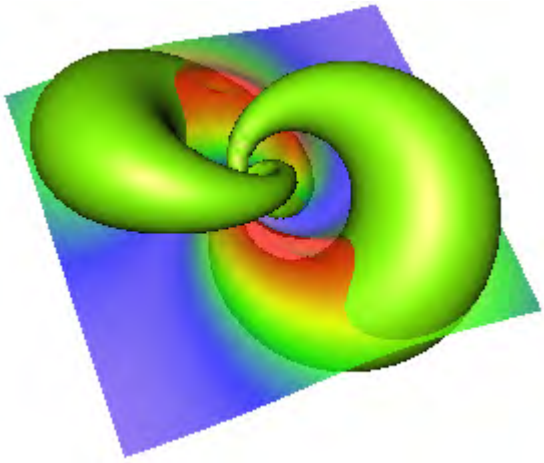
Blend	Render transparent objects using multiplicative alpha blending the most commonly used type for rendering transparent objects. The new pixel value is calculated as the old pixel color multiplied with the numeral one minus the source alpha value plus the source pixel color multiplied by the source alpha value. Use this transparency mode only if you have one transparent object in your scene and it is rendered after any opaque objects.	

Delayed Blend	Render transparent objects using multiplicative alpha blending in a second rendering pass with depth buffer updates disabled. Use this transparency type when you have one transparent object or several transparent objects that you are certain might never overlap (when projected on the screen). This method is not as fast as the *Blend* transparency type, since the scene graph is traversed twice. This method makes two passes through the object list.	

Sorted Object Blend	Render transparent objects using multiplicative alpha blending. Opaque objects are rendered first; transparent objects are rendered back to front with z-buffer updates disabled. Use this transparency mode when you have several transparent objects that might overlap when projected on the screen. This method requires 1 plus the number of transparent objects rendering passes.	

Sorted Object, Sorted Triangle Add	Render transparent objects back to front. The triangles in each object are sorted back to front before rendering. Use this transparency type when you have one or more transparent objects in which you know the triangles might overlap inside the object. This transparency type might be very slow if you have an object with many triangles, as all triangles have to be sorted before rendering. Lines and points are not sorted before rendering but are rendered as in the normal *Sorted Object Blend* transparency type. This transparency type does not guarantee "correct" transparency rendering. It is almost impossible to find an algorithm that sorts triangles correctly in all cases, and intersecting	

	triangles are not handled. Also, since each object is handled separately, the presence of two intersecting objects leads to incorrect transparency.	

Sorted Object, Sorted Triangle Blend	Render transparent objects back to front. The triangles in each object are sorted back to front before rendering. Use this transparency type when you have one or more transparent objects in which you know the triangles might overlap inside the object. This transparency type might be very slow if you have an object with many triangles as all triangles have to be sorted before rendering. Lines and points are not sorted before rendering but are rendered as in the normal *Sorted Object Blend* transparency type. This transparency type does not guarantee "correct" transparency rendering. It is almost impossible to find an algorithm that sorts triangles correctly in all cases, and intersecting triangles are not	

	handled. Also, since each object is handled separately, the presence of two intersecting objects leads to incorrect transparency.	

Home

Click the **View | Home** command or click the  button to reset the camera position, rotation, and zoom to the last saved position.

The home position is automatically set when the **Tools | Options** *Enable AutoZoom* option box is checked and when **Voxler** has performed the AutoZoom operation. The home position is set manually with the **View | Set Home** command. After setting the view to the rotation, zoom level, and position desired, the **View | Set Home** command should be used to remember this position.

Changing the view to the home position cannot be undone.

Set Home

Click the **View | Set Home** command or  button to store the current camera rotation, zoom, and position to the home position. The new home position is stored for each **Voxler** project individually and only while the project is open. After the view has changed, this saved state is restored by clicking the **View | Home** command.

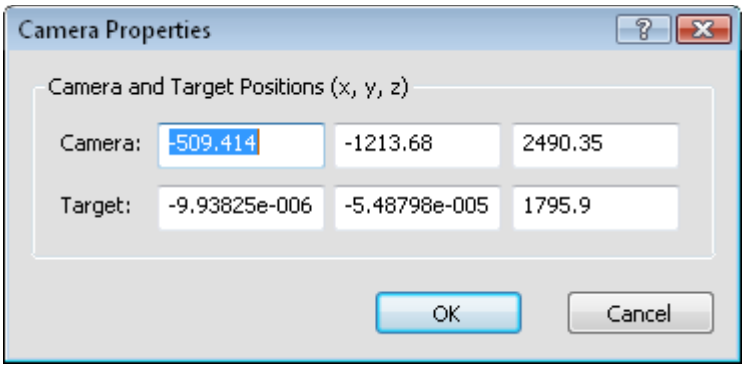
The **Set Home** command cannot be undone. This means that if you set a new view to the home position, the old position is lost. Once a **Voxler** project .VOXB file is closed and reopened, the home position is returned to the default *Top* view.

Camera Properties

The camera properties control the zoom, rotation, and angle of objects in the **Viewer** window.

Change View with Camera Properties Dialog

Click the **View | Camera Properties** command or  button to change the camera and target positions. The **Camera Properties** dialog displays.



*You can change the Camera and Target Positions (x, y, z) in the **Camera Properties** dialog.*

Camera

The *Camera* position contains the coordinates of the viewer's eye (the camera). Enter a value for each coordinate (in the same units used by your data).

Target

The *Target* position contains the coordinates of the object at which the camera is looking. Enter a value for each coordinate (in the same units used by your data).

OK or Cancel

Click *OK* to save your changes and close the dialog. The view will update to show the new camera and target position. Click *Cancel* to return to the current view without making any changes.

Change View with Mouse

To change the current view with the mouse, click the **View | Trackball** command, position the mouse in the **Viewer** window, hold down the left mouse button, and drag the mouse to the desired position.

Default Home Position

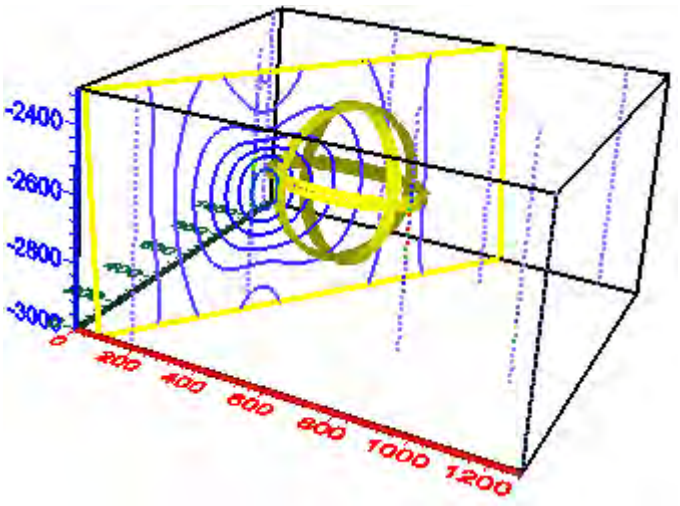
Choose the **View | Home** command to undo all value changes and return to the default home position. Change the camera properties and choose the **View | Set Home** command to set the current view as the home position.

Changing the camera and target positions cannot be undone.

Dragger

The *Slice*, *ClipPlane*, *Contours*, *ObliqueImage*, and *StreamLines* modules have a *Dragger* option in the Property Manager. The *Dragger* allows interactive positioning and rotation of the plane.

In the **Property Manager**, check the *Show Dragger* box to show the dragger— a virtual, rotatable trackball— and allow interactive positioning and rotation of the plane. The dragger allows the orientation and offset of the cutting plane to be specified.



The Dragger is shown above in yellow and can be used to interactively rotate the cutting plane.

Rotate the Dragger Trackball

Drag one of the three bands to rotate around a principal axis in the direction of the ring. Drag anywhere on the ball (between the rings) to perform an unconstrained rotation in any direction.

User Defined Rotation

To specify a user-defined rotation axis, press the SHIFT key while clicking the left mouse button and dragging. A new distinctively-colored axis is added.

Scale the Size of Trackball

To scale the size of the trackball, press the CTRL key and drag the trackball.

Offset the Plane

To offset the plane in the perpendicular direction, drag the cutting plane itself.

How can I prevent objects from spinning?

When I drag the mouse in the **Viewer** window, it makes the objects in the window spin. Is there a way to prevent them from spinning?

To prevent the object in the **Viewer** window from spinning, after dragging the mouse, hold the mouse stationary prior to releasing the left mouse button.

Chapter 18 - Voxler Worksheet

Worksheet Document

Worksheet windows are a view of the data file and are designed to display, edit, enter, and save data. The worksheet windows have several useful and powerful editing, transformation, and statistical operations available. Several import and export options are available for opening data files from other spreadsheet programs.

Worksheet Commands

The worksheet menu commands include:

File	Opens, closes, saves, and prints files
Edit	Contains undo, redo, cut, copy, paste, paste special, clear, insert, delete, find, find next, and replace
View	Controls the display of toolbars, status bar, and managers
Format	Sets cell formats, column widths, and row heights
Data	Contains commands to sort data, compute statistics, assign projection, reproject data, and calculate mathematical transformations
Tools	Adjusts program options
Window	Controls display of the windows
Help	Provides access to help topics

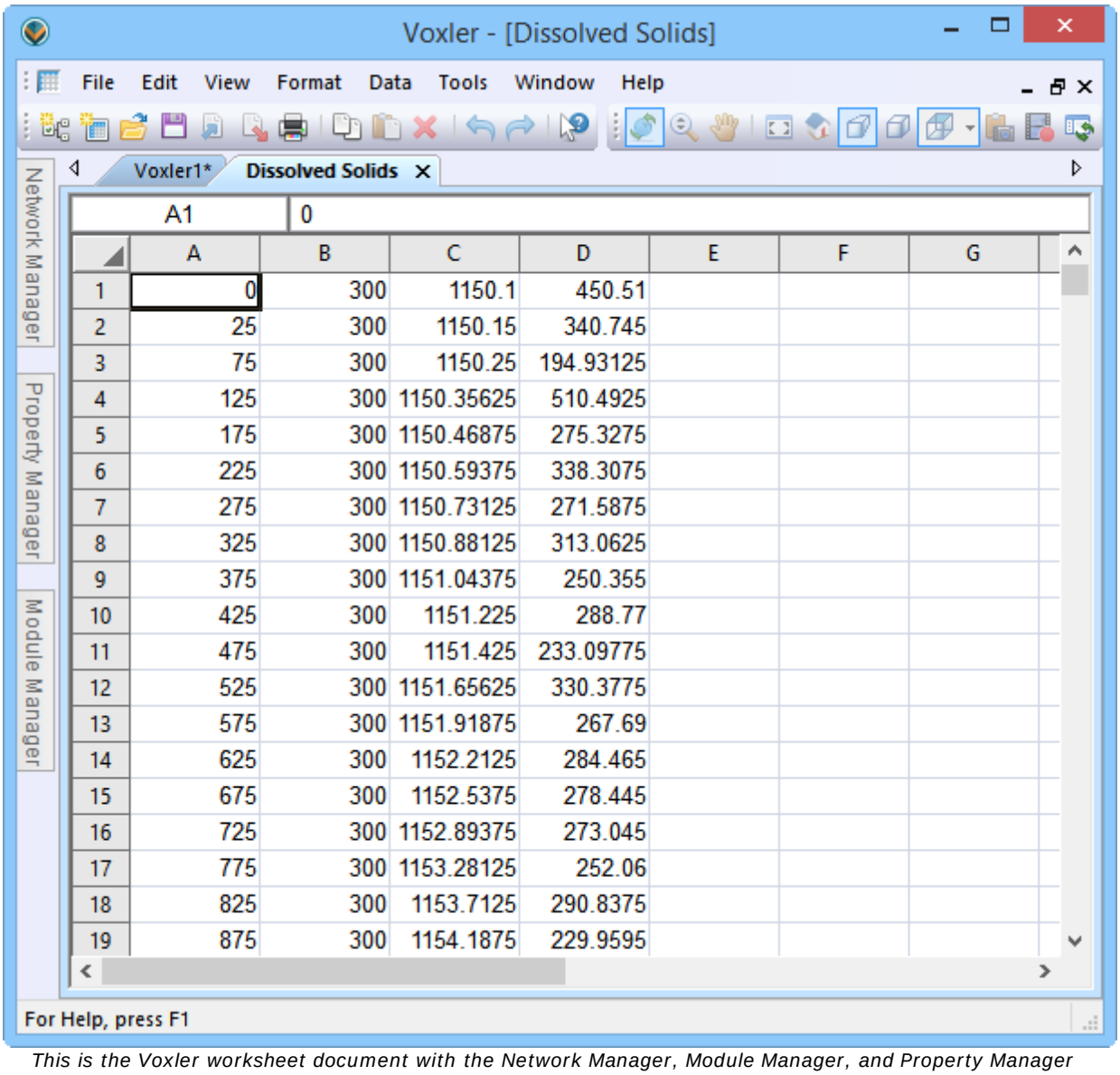
The Application/Document Control menu commands control the size and position of the application window or the document window.

Tab View

The plot, worksheet, and grid node editor windows are displayed as tabbed documents. When more than one window is open, tabs appear at the top of the document, allowing you to click on a tab to switch to a different window. The tabs may be dragged to reorder them. When a document contains unsaved changes, an asterisk (*) appears next to its tabbed name. The asterisk is removed once the changes have been saved.

Worksheet Document

The image below displays the worksheet document. See the *Worksheet Window* topic for information on the components of the worksheet window.



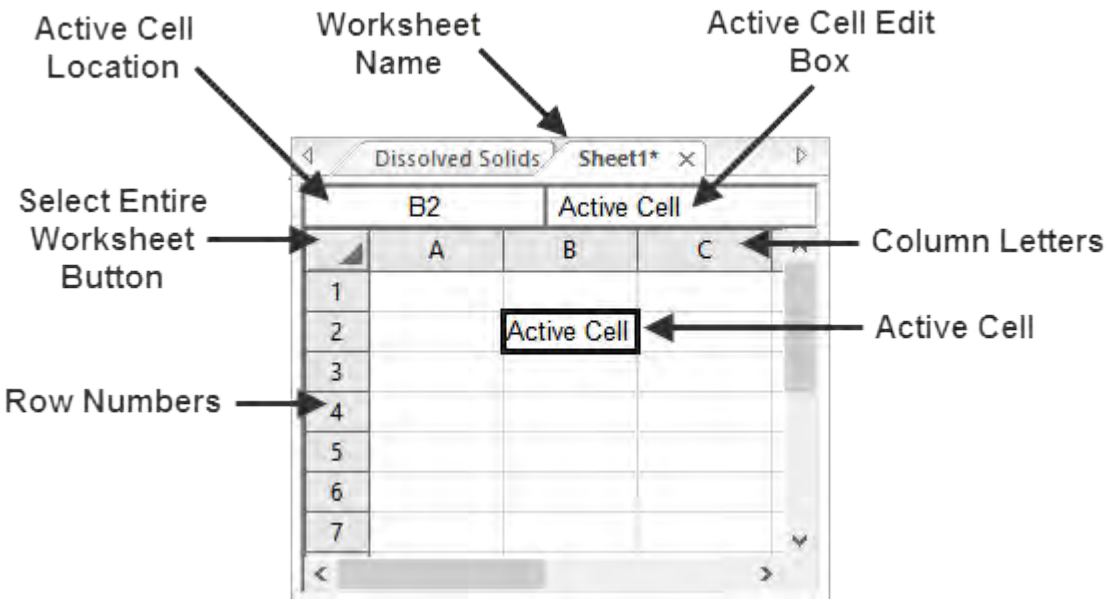
	A	B	C	D	E	F	G
1	0	300	1150.1	450.51			
2	25	300	1150.15	340.745			
3	75	300	1150.25	194.93125			
4	125	300	1150.35625	510.4925			
5	175	300	1150.46875	275.3275			
6	225	300	1150.59375	338.3075			
7	275	300	1150.73125	271.5875			
8	325	300	1150.88125	313.0625			
9	375	300	1151.04375	250.355			
10	425	300	1151.225	288.77			
11	475	300	1151.425	233.09775			
12	525	300	1151.65625	330.3775			
13	575	300	1151.91875	267.69			
14	625	300	1152.2125	284.465			
15	675	300	1152.5375	278.445			
16	725	300	1152.89375	273.045			
17	775	300	1153.28125	252.06			
18	825	300	1153.7125	290.8375			
19	875	300	1154.1875	229.9595			

This is the Voxler worksheet document with the Network Manager, Module Manager, and Property Manager in auto hide mode on the left, and the tabbed documents at the top of the worksheet.

Worksheet Window

To enter data in a worksheet, click the **File | Open** command to open an existing data file or click the **File | New | Worksheet** command to create a blank worksheet.

The components of the worksheet window are discussed below.



The components of a worksheet window shown above are described in the table below.

Component Name	Component Function
Column Letters	The letter that identifies a column of the worksheet.
Row Numbers	The number that identifies a row of the worksheet.
Active Cell	The cell highlighted with a bold outline. The active cell receives data input (numeric values or text strings) from the keyboard. Only one cell is active at a time.
Active Cell Location	The location of the active cell, specified by column letter and row number.
Active Cell Edit Box	The box displaying the data or text contained in the active cell. Data typed into an empty cell appears in both the edit box and the active cell.
Worksheet Name	The name of the data file displayed in the worksheet or the worksheet number prior to saving.
Select Entire Worksheet Button	The button used to select all cells in the worksheet. Located in the top left corner of the worksheet.

Active Cell

The active cell is displayed with a heavy border surrounding the cell. The contents of this cell are displayed in the cell edit box. You can enter or edit data in the active cell. To edit existing data, activate the desired cell and press the F2 key or highlight the information in the cell edit box.

Special Key Functions when editing the active cell :

Keyboard Command	Action
ESC	ESC cancels edit mode and restores the original contents of the active cell.
ENTER	ENTER stores the contents of the cell edit box and then moves the active cell down one cell.
CTRL+ENTER	CTRL+ENTER completes the entry and keeps the current cell active.
ARROWS (left and right)	Left and right ARROWS move within the cell's text if the F2 key has been pressed. Otherwise, these keys store the contents of the cell edit box and then move the active cell to the left or right.
ARROWS (up and down)	Up and down ARROWS store the contents of the cell edit box in the active cell and move the active cell above or below.
DELETE	DELETE deletes the character to the right of the cursor if the F2 key has been pressed. Otherwise, pressing the delete key deletes the entire contents of the cell.
BACKSPACE	BACKSPACE deletes the character to the left of the cursor if the F2 key has been pressed. Otherwise, pressing the backspace key deletes the entire contents of the cell.
PAGE UP and PAGE DOWN	PAGE UP and PAGE DOWN store the contents of the cell edit box in the active cell and move one page up or down.
TAB and SHIFT+TAB	TAB and SHIFT+TAB store the contents of the cell edit box in the active cell and move the active cell to the right or left.

Active Cell Edit Box

The cell edit box is located at the top of the worksheet window just above the column letter bar. The cell edit box shows the contents of the active cell and is used for editing cells. Use the cell edit box to see the contents of a worksheet cell when the column is too narrow to display all of the cell contents.

To begin editing the selected cell, press the F2 key. Alternatively, highlight the contents of the cell edit box to edit the cell. To overwrite the current cell contents, simply begin typing without pressing F2. If the mouse is clicked on a new cell, the new cell becomes the active cell.

Right-click in the active cell edit box to access the following commands in the context menu:

Right to left Reading order	Toggles right to left reading order on or off.
Show Unicode control characters	Toggles the display of Unicode control characters on or off.
Insert Unicode control character	Select a Unicode control character from the list, and it is inserted in the active cell edit box at the cursor location.
Open/Close IME	When a user types a phonetic representation of a word, the IME displays a candidate list on the screen. The user can select the intended word or phrase from among several different possible representations in the candidate list, and the user's selection then replaces the phonetic representation in the document. This command toggles the IME on or off.
Reconversion	IME reconversion allows users who are typing in Japanese to convert back and forth between the phonetic spelling of a word (using the standard Western keyboard) and the Japanese character that represents the word.

Special Key Functions when Editing the Active Cell:

Keyboard Command	**Action**
ESC	ESC cancels edit mode and restores the original contents of the active cell.
ENTER	ENTER stores the contents of the cell edit box and then moves the active cell down one cell.
CTRL+ENTER	CTRL+ENTER completes the entry and keeps the current cell active.
ARROWS (left and right)	Left and right ARROWS move within the cell's text if the F2 key has been pressed. Otherwise, these keys store the contents of the cell edit box and then move the active cell to the left or right.
ARROWS (up and down)	Up and down ARROWS store the contents of the cell edit box in the active cell and move the active cell above or below.
DELETE	DELETE deletes the character to the right of the cursor if the F2 key has been pressed. Otherwise, pressing the delete key deletes the entire contents of the cell.
BACKSPACE	BACKSPACE deletes the character to the left of the cursor if the F2 key has been pressed. Otherwise, pressing the backspace key deletes the entire contents of the cell.
PAGE UP and PAGE DOWN	PAGE UP and PAGE DOWN store the contents of the cell edit box in the active cell and move one page up or down.
TAB and SHIFT+TAB	TAB and SHIFT+TAB store the contents of the cell edit box in the active cell and move the active cell to the right or left.

Active Cell Location Box

The active cell location box shows the location of the active cell in the worksheet. Letters are the column labels and numbers are the row labels.

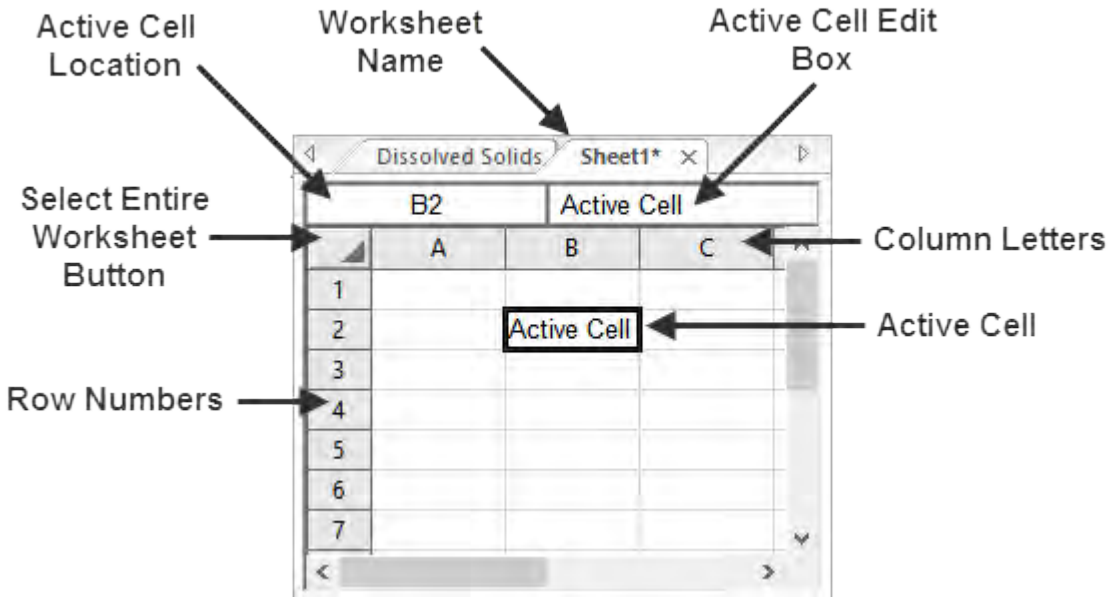
C5 | 1150.46875

	A	B	C	D
1	0	300	1150.1	450.51
2	25	300	1150.15	340.745
3	75	300	1150.25	194.93125
4	125	300	1150.35625	510.4925
5	175	300	1150.46875	275.3275
6	225	300	1150.59375	338.3075

This example shows the active cell as cell C5. The name of the active cell "C5" is listed in the active cell location box in the upper left portion of the worksheet.

Select Entire Worksheet

Clicking on the small box above the row labels and to the left of the column labels selects the entire worksheet.



The Select Entire Worksheet button is located to the left of column A and above row 1.

Selecting Cells

The keyboard and the mouse may be used to select cells. Selected cells are indicated by reverse video (white background becomes black, etc.). Hidden cells are selected if their columns or rows are within a selected block of cells. Single cells, a rectangular block of cells, one or more rows, one or more columns, or the entire worksheet can be selected.

Cells may be selected to:

- perform editing and clipboard functions,
- perform a transform function,
- sort the selected cells,
- compute statistics for selected cells, or to
- set column properties for several columns via the **Format** menu, such as column width, row height, and cell format.

There are several ways to select cells:

- Clicking on the small box above the row labels and to the left of the column label bar selects the entire worksheet.
- To deselect all selected cells, click the left mouse button anywhere within the worksheet, or move the active cell with an ARROW key. Alternatively, the PAGE UP, PAGE DOWN, HOME, and END keys may also be used to deselect the cells.
- To rapidly select a large block, first select one corner of the block, and then use the scroll bars to scroll to the opposite corner. Hold down the SHIFT key and click on the cell at the opposite corner. The PAGE UP, PAGE DOWN, HOME, and END keys may also be used, but the SHIFT key must be held down while these keys are pressed. The SHIFT key is not needed while using the scroll bars.
- To select all cells in a column or row, click the column letter or row number. To select several adjacent columns or rows, press and hold the left mouse button and drag the pointer on the column letters or row numbers.
- While holding down the CTRL key, the active cell may be repositioned for selecting a new, discontinuous block.
- The CTRL key is used to select multiple blocks and the SHIFT key is used to resize the last selected block. Details and exceptions are given in separate help sections for selecting with the mouse and selecting with the keyboard.
- If entire rows or columns are selected by clicking on the headers, some operations, such as statistics, can take a long time. Rather than clicking on the headers, only select the cells containing data.
- Clicking and holding the left mouse button while dragging the mouse in the worksheet selects a block. Similarly, using the SHIFT key plus the ARROW keys selects a block.
- The keys used with SHIFT for selecting cells are the ARROW keys, PAGE UP, PAGE DOWN, HOME, and END. TAB and SHIFT+TAB cannot be used.
- While holding down the SHIFT key, the last selected block may be resized. Use the SHIFT key and the mouse or the SHIFT key and ARROW keys.
- The active cell is at one corner (or edge) of a selected block and must first be positioned before selecting multiple cells.
- The last block cannot be resized if the active cell has been moved.
- Once selected, a block of cells cannot be unselected unless all cells are unselected.

Selecting Cells with the Keyboard

The keyboard may be used to select cells. Selected cells are indicated by reverse video (white background becomes black, etc.).

To Select	Process
Single cells	Click in the cell to select it, or use the arrow keys to select a cell. The selected cell will have a thick outline around it.
A rectangular block of cells	Move the active cell to one corner of the block. While holding down the SHIFT key, use the movement keys to position the opposite corner of the block. The movement keys include the ARROW keys, PAGE UP, PAGE DOWN, HOME, and END, but not TAB and SHIFT+TAB. When the block has been sized, release the SHIFT key. To resize the block, see the instructions below.
Several adjacent rows	Select the first or last row. Then, while holding down the SHIFT key, use the vertical movement keys. These include up ARROW, down ARROW, page up, page down, HOME, and END.
Several adjacent columns	Select the first or last column. Then, while holding down the SHIFT key, use the right and left ARROW keys.

Resize the Last Selected Block

To resize the last selected block, hold down the SHIFT key while using the movement keys (as appropriate to the type of block). The last block cannot be resized if the active cell has been moved.

Deselect All Selected Cells

To deselect all selected cells, left-click anywhere within the worksheet or move the active cell with an ARROW key or other movement key.

Selecting Cells with the Mouse

The mouse may be used to select cells. Selected cells are indicated by reverse video (white background becomes black, etc.).

To Select	Process
Single cells	Click the left mouse button on the cell. The cell will have a thick outline around it.
A rectangular block of cells	Move the active cell to one corner of the block. Click and hold the left mouse button, and drag it to the opposite corner of the block. Then release the mouse button.
An entire row	Click the mouse on the row label.
Several adjacent rows	Click and hold the mouse on the first row label and drag it to the last row. Make sure the cursor is a normal arrow cursor not the double arrow cursor used for selecting column dividing lines.
An entire column	Click the mouse on the column label.
Several adjacent columns	Click and hold the mouse on the first column label and drag it to the last column. Make sure the cursor is a normal arrow cursor not the double arrow cursor used for selecting column dividing lines.
The entire worksheet	Click on the small box above the row labels and to the left of the column label bar.

The worksheet will scroll automatically if the mouse is dragged past the visible limits of the worksheet.

Select Additional Blocks

To select additional blocks, hold down the CTRL key while clicking.

Resize the Last Selected Block

To resize the last selected block, hold down the SHIFT key while clicking and holding the left mouse button. Then, drag the edge of the last selected block to the new position. The last block cannot be resized if the active cell has been moved.

Deselect All Selected Cells

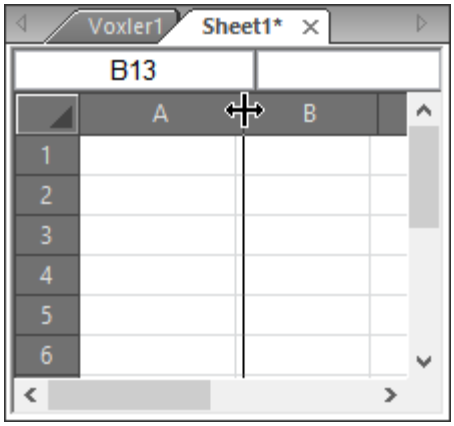
To deselect all selected cells, left-click anywhere within the worksheet or move the active cell with an arrow key or other movement key.

Selecting a Column or Row Dividing Line

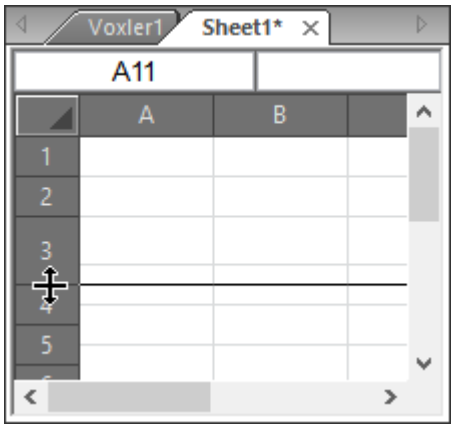
The column or row dividing lines are the lines between the column letter labels and row number labels along the borders of the worksheet. These lines divide the columns or rows. When selecting a dividing line, the cursor must be within appoximately a characther's width of the dividing line and it must be on the label bar.

Change the column width or row height by dragging the dividing line. Rows or columns can be hidden or unhidden by using the mouse. The **Format | Column Width** or **Format | Row Height** commands can also be used to set column widths or row heights.

Move the cursor to the label bar near the dividing line until the cursor changes to a  between columns, and a  between rows. The cursor must be within approximately a character's width of the dividing line and it must be on the label bar. Click and hold the left mouse button and drag the dividing line.



This example shows the cursor being used to change the width of column A.



This example shows the cursor being used to change the height of row 3.

Hiding Columns or Rows

The mouse may be used to hide columns or rows.

To hide a column, first click on the vertical dividing line to the right of the column. Drag the vertical dividing line to the left as far as it will go and then release the mouse button. If there are hidden columns to the right of this column, grab the left side of the vertical dividing line. If the right side of the vertical dividing line is selected, the vertical dividing line for the adjacent hidden column is selected.

To hide a row, first click on the horizontal dividing line at the bottom of the row. Drag the horizontal dividing line up as far as it will go and then release the mouse button. If there are hidden rows above this row, grab horizontal dividing line just below the line. If the top side of the horizontal dividing line is selected, the horizontal dividing line for the adjacent hidden row is selected.

With the Format Menu

Columns and rows can also be hidden with the **Format | Column Width** and **Format | Row Height** commands. Select the columns or rows to hide, click the **Format | Column Width** or **Format | Row Height** command, and then set the *Column Width* or *Row Height* to zero.

Displaying Hidden Columns or Rows

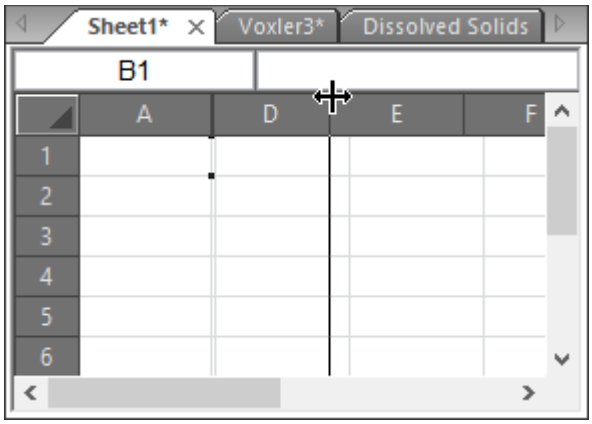
The mouse may be used to display hidden columns or rows.

To display a hidden column, first click on the vertical dividing line at the **right** of the hidden column and then drag the vertical dividing line to the new position. If several adjacent columns are hidden, only the far right column is displayed after dragging the dividing line. If the cursor is to the left of the vertical dividing line when the line is selected, then the selected vertical dividing line is for the visible column to the left and not for the hidden column.

To display a hidden row, first click on the horizontal dividing line below the hidden row and then drag the horizontal dividing line to the new position. If several adjacent rows are hidden, only the bottom row is displayed after dragging the dividing line. If the cursor is above the horizontal dividing line when the line is selected, then the horizontal dividing line will be for the visible row above the hidden rows, and not for the hidden rows.

Example

If columns B, C, and D are hidden and columns A and E are visible, then one vertical dividing line appears between columns A and E. Select that vertical dividing line with the cursor slightly to the right of the dividing line. This will select the line for column D. (If the cursor is to the left of the dividing line, then the dividing line for column A will be selected.) Drag the vertical dividing line to the right to unhide column D. Repeat for columns C and B.



This example shows the vertical line being dragged to display hidden column D.

With the Format Menu

Hidden columns and rows can also be displayed with the **Format | Column Width** and **Format | Row Height** commands. To display hidden rows or columns, select the columns or rows on both sides of the hidden columns or rows, click the **Format | Column Width** or **Format | Row Height** command, and then set the *Column Width* or *Row Height* to a number greater than zero.

Worksheet Error Codes and Special Numeric Values

There are a few different error codes and special numeric values that can appear in a worksheet cell depending on the type and nature of the data that appears.

Codes	Explanation
#########	number will not fit in the column - the column must be wider for the number to be shown
#N/A	value cannot be computed (for example, not enough data to calculate a statistic)
# DIV/0!	an attempt to divide-by-zero was made in performing a calculation
#ERROR	a value could not be computed (for example, square root of a negative number)
#OVERFLOW	the value is too large for the worksheet (largest absolute value is about 1.797E+308)
1.# INF	the value is too large for the worksheet (i.e., "infinite" value)
1.# IND	numeric value is indefinite (usually the result of performing a calculation with an infinite value or attempting to divide by zero)

Worksheet Specifications

The following technical specifications for the worksheet include the number of cells allowed in the worksheet and the nature of the numbers allowed in the worksheet.

- Maximum number of rows in a worksheet: 1 billion
- Maximum number of columns in a worksheet: 1 billion
- Maximum numeric precision: 15 digits (Counting the digits before and after the decimal place)
- Maximum numeric resolution: 2.22E-16 (The smallest detectable difference between two numbers)
- Maximum absolute value: 1.79769E+308 (The largest value that can be represented)
- Minimum absolute value: 2.22507E-308 (The smallest value that is different from zero)
- Double precision floating-point numbers can only represent approximately 15 significant decimal digits.
- Approximate memory requirements for unformatted numeric data: 10.5 bytes per cell + 24 bytes per column

Example 1

10,000 rows of numbers in 3 columns

30,000 cells x 10.5 bytes/cell = 315,000 bytes (308 Kbytes)

3 columns x 24 bytes/column = 72 bytes

TOTAL MEMORY NEEDED (in addition to memory needed to run the program): 308 Kbytes

Example 2

3 rows of numbers in 10,000 columns

30,000 cells x 10.5 bytes/cell = 315,000 bytes (308 Kbytes)

10,000 columns x 24 bytes/column = 240,000 bytes (234 Kbytes)

TOTAL MEMORY NEEDED (in addition to memory needed to run the program): 542 Kbytes

Editing the Worksheet

Edit Menu Commands

The worksheet **Edit** menu contains several standard editing options. Some menu commands are available through the right-click context menus as well as the main menus. You can right-click on selected cells to open the context menu.

The worksheet **Edit** menu has the following commands:

Undo	Undo the last action
Redo	Redo the previously undone action
Cut	Cut the selected object to the clipboard
Copy	Copy an object to the clipboard
Paste	Insert clipboard contents into the document
Paste Special	Paste clipboard contents with various formatting options
Clear	Remove the selected cells
Insert	Displace selected cells and insert new cells into the worksheet
Delete	Delete selected worksheet cells and move rows or columns
Find	Find a particular word or phrase in the worksheet
Find Next	Find the next occurrence of the word or phrase
Replace	Replace the word or phrase with alternate text

Cut

Click the **Edit | Cut** command or press CTRL+X on the keyboard to move the selected cells to the clipboard. This deletes the selected cells from the file after copying them to the clipboard. Cut objects can later be pasted with the **Paste** or **Paste Special** commands.

Only one set of data may be placed in the clipboard at a time. The next **Cut** or **Copy** command replaces the contents of the clipboard.

Copy

Click the **Edit | Copy** command, click the  button, or press CTRL+C on the keyboard to copy the selected cells to the clipboard. The original objects remain in the window. Use this command to duplicate cells in a different location in the same window, or copy the objects into a different window or application. The copied objects can later be pasted with the **Paste** or **Paste Special** commands.

Only one set of data may be placed in the clipboard at a time. The next **Cut** or **Copy** command replaces the contents of the clipboard.

Paste

Click the **Edit | Paste** command, click the  button, or press CTRL+V on the keyboard to copy the clipboard contents into the current document. The objects to be copied must first be placed in the clipboard using the **Edit | Cut** or **Edit | Copy** commands of **Voxler** or some other application. The clipboard contents remain on the clipboard until something new is cut or copied to the clipboard.

In the worksheet, the upper left corner of the pasted data is placed in the active cell. Any cells in the existing worksheet that lie to the right of and below the active cell will be overwritten with the contents of the pasted data. The following rules are used to paste into the worksheet:

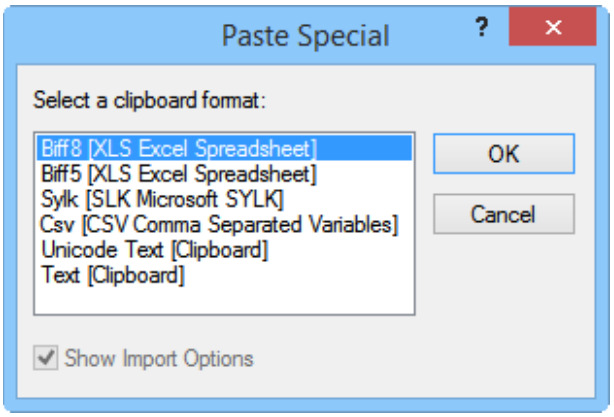
- Only the TAB character is recognized as a column separator. Spaces, commas, semi-colons, etc. are included in a text cell.
- The RETURN character is recognized as the row separator.
- Numbers paste as number values. The period can always be used as a decimal separator, and the system locale determines any other decimal separator. For example if the system locale uses a comma as the decimal separator, then both 123.456 and 123,456 are pasted as the number 123,456.
- Mixed text and numbers paste as text.
- Dates and/or times will paste as date values in a wide variety of date/time formats. Ambiguous dates are determined by the system locale setting. If dates are not pasted correctly, consider using **Paste Special** and the *Locale* settings in the **Data Import Options** dialog.

Paste Special

The **Edit | Paste Special** command in the worksheet specifies the format for the pasted contents. Clicking the **Edit | Paste Special** command opens the **Paste Special** dialog.

Paste Special Dialog

When data are copied to the clipboard, special formatting information is also copied. The **Paste Special** dialog determines the format in which the contents are pasted into the worksheet. The **Paste Special** command can also be used to remove text formatting or correctly import data with different regional formatting.



*Select a paste special format in the **Paste Special** dialog. This example shows options after copying **Voxler** worksheet data and using paste special.*

The clipboard formats displayed may vary depending the original location of the information being copied. For example, data copied from the **Voxler** worksheet may yield different options than data copied from Excel.

Biff8 [Excel Spreadsheet]

The *Biff8 [Excel Spreadsheet]* format is a Microsoft Excel Binary Interchange File Format (BIFF) version 8.

Biff5 [Excel Spreadsheet]

The *Biff5 [Excel Spreadsheet]* format is a Microsoft Excel Binary Interchange File Format (BIFF) version 5.

Biff4 [Excel Spreadsheet]

The *Biff4 [Excel Spreadsheet]* format is a Microsoft Excel Binary Interchange File Format (BIFF) version 4.

Biff3 [Excel Spreadsheet]

The *Biff3 [Excel Spreadsheet]* format is a Microsoft Excel Binary Interchange File Format (BIFF) version 3.

Biff [Excel Spreadsheet]

The *Biff [Excel Spreadsheet]* format is a Microsoft Excel Binary Interchange File Format (BIFF).

Sylk [Microsoft SYLK]

The *Sylk [Microsoft SYLK]* format is a symbolic link Microsoft file format typically used to exchange data between applications, specifically spreadsheets. The *Sylk* file format is composed of only displayable ANSI characters, allowing it to be easily created and processed by other applications, such as databases.

Unicode Text [Clipboard Text]

The *Unicode Text [Clipboard]* format is unformatted text.

Text [Clipboard Text]

The *Text [Clipboard]* format is unformatted text.

Show Import Options

When *Unicode Text [Clipboard]* or *Text [Clipboard]* is selected, the *Show Import Options* option is available. Check the box to open the **Data Import Options** dialog before importing the data.

Clear

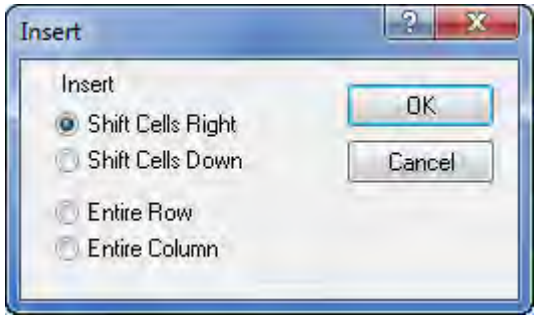
Click the **Edit | Clear** command, or press the DELETE key on the keyboard, to remove data from selected worksheet cells. The cells become empty when the data are removed. To shift the data from unselected cells into the selected cell locations, use the **Delete** command instead.

Insert

The **Edit | Insert** command inserts a single blank cell or a block of blank cells in the worksheet. Select cells in the area in which you wish to insert cells and then click **Edit | Insert**, or right-click and select **Insert** from the context menu. The **Insert** dialog appears. Specify how you want the original displaced contents moved when the blank cells are inserted.

Insert Dialog

Click the **Edit | Insert** command, or press CTRL+R on the keyboard, to open the **Insert** dialog.



*When using **Edit | Insert**, you can shift cells to the right or down to make room for the new cells.*

Shift Cells Right or Shift Cells Down

Click the *Shift Cells Down* or *Shift Cells Right* option to insert blank cells and displace the original contents either down or to the right.

Entire Row or Entire Column

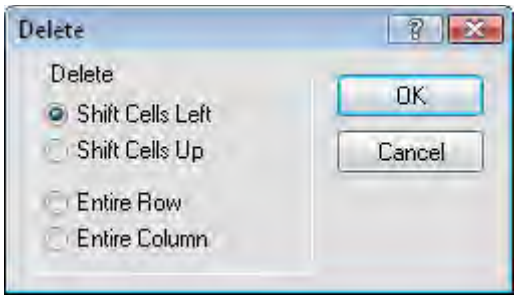
Click the *Entire Row* or *Entire Column* option to insert an entire row or column in the area that contains highlighted cells.

Delete

The **Edit | Delete** command deletes the selected worksheet cells and shifts cells up or to the left to fill in the gap. After selecting **Edit | Delete**, the **Delete** dialog appears. Specify the desired behavior of the cells and click the *OK* button. The selected cells are deleted and the contents of cells below or to the right are moved to fill the deleted block.

Delete Dialog

Click the **Edit | Delete** command, right-click and select **Delete**, or press CTRL+D on the keyboard, to open the **Delete** dialog.



*When using **Edit | Delete**, you can shift cells to the left or up to fill in the gap.*

Shift Cells Left or Shift Cells Up

Click the *Shift Cells Up* or *Shift Cells Left* option to specify if cells will be shifted to the left or up to fill in the gap after deleting the selected cells.

Delete Entire Row or Entire Column

Click *Entire Row* or *Entire Column* to delete the entire row or column that contains highlighted cells.

Leave Deleted Cells Empty

To leave the selected cells empty when the data are removed, use the **Clear** command, press the DEL key, or use the **Cut** command.

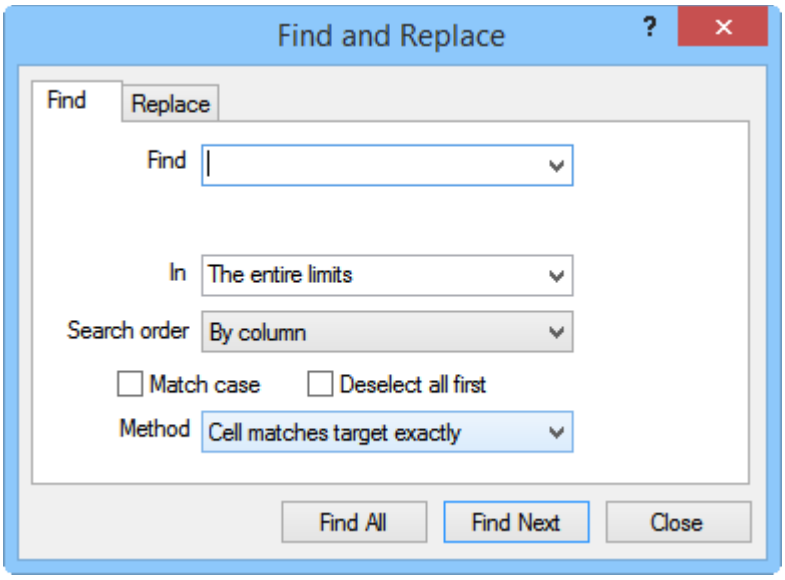
Find and Replace

The **Find and Replace** dialog displays when the **Edit | Find** or **Edit | Replace** commands are selected. The **Find and Replace** dialog is used to search and replace specific numbers or text in the worksheet.

The **Edit | Find Next** command is used to find the next instance of a particular number, word, or phrase in the worksheet. If the **Edit | Find** command was not used initially, the **Find and Replace** dialog opens so that you can define your search criteria.

The Find Page

Clicking the **Edit | Find** command, pressing CTRL+F on the keyboard, clicking the **Edit | Find Next** command, or clicking the F3 key on the keyboard, opens the **Find** page of the **Find and Replace** dialog.



Search for and replace specific text in the worksheet with the Find and Replace dialog.

Find

To find objects, type the text you want to search for in the *Find* field. Click the arrow at the right to select from a list of the most recently used text strings. The asterisk * and question mark ? wildcards can be used in the *Find* box. Click the arrow at the right to select from a list of the most recently used criteria.

- A question mark ? finds a single character in the specified location. For example, 200? finds 2009, 2008, 200a, etc.
- An asterisk * finds any number of characters at the specified location. For example, *01 finds 601, 1201, c01, etc.

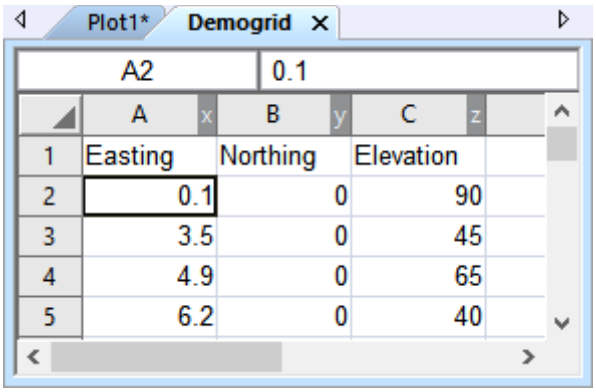
In

Next to *In*, choose the parameters of the search from the list. Choices include *The column where active cell is*, *The row where active cell is*, and *The entire limits*.

- Select *The column where the active cell is* to search only the column (i.e. column B) of the active cell (i.e. cell B2) for the information listed in the *Find* field.
- Select *The row where active cell is* to search only the row (i.e. row _2) of the active cell (i.e. cell B2) for the information listed in the *Find* field.
- Select *The entire limits* to search the entire worksheet for the information listed in the *Find* field.

Search Order

The *Search order* controls the direction of the search: down through columns by selecting *By column* or to the right across rows by selecting *By row*.



In this example, cell A2 is selected. If the Find criteria is "0", and By column is the search order, cell A4 is found first. If By row is the search order, cell B2 is found first.

Match Case

If you have case sensitive characters in the *Find* text string, check the *Match case* check box. Selecting *Match case* distinguishes between uppercase and lowercase characters. For example, a search for "Elevation" with the *Match case* option selected will not find entries for "elevation", but will find entries for "Elevation".

Deselect All First

Check the *Deselect all first* box to deselect all selected cells before performing the search. All previously selected cells will be deselected prior to the search when the *Deselect all first* check box is checked. If the *Deselect all first* box is deselected, the results of a previous search will remain highlighted when performing the next search.

Method

Choose the search *Method* from the list to determine how the search is performed. The examples assume "Golden, CO" is in the *Find* field.

- Select *Cell matches target exactly* to require that the exact criteria in the *Search* box is present in a cell before it is selected. For example, only cells that have exactly "Golden, CO" will be selected.
- Select *Cell contains target phrase* to require that the phrase in the *Search* box is present in a cell before it is selected. For example, cells that has "Golden CO", "Golden Company", or "Golden Colorado" will be selected.
- Select *Cell contains all of the target words* to require that all of the *Search* criteria words are present in a cell before it is selected. For example, cells that have "Golden" and "CO" somewhere in the cell (i.e. "Golden is the best city in Colorado" will be selected).
- Select *Cell contains any of the target words* to require that any of the *Search* criteria words are present in a cell before it is selected. For example, cells that have "Golden is a city" or "CO is a state" will be selected.

Find All Button

Click the *Find All* button to find all occurrences of the *Find* criteria in the worksheet. All of the cells that contain the *Find* criteria will be highlighted.

Find Next Button

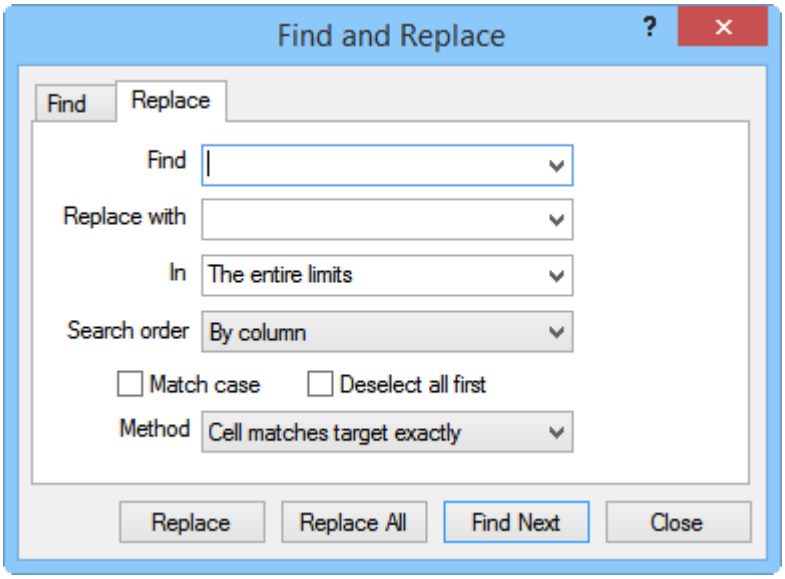
Click the *Find Next* button to find the next occurrence of the characters specified in the *Find* box. This allows you to meet the criteria one at a time. The next instance of the *Find* criteria will be highlighted.

Close Button

Click the *Close* button to exit the **Find and Replace** dialog.

The Replace Page

Clicking the **Edit | Replace** command or pressing CTRL+H on the keyboard, opens the **Replace** page of the **Find and Replace** dialog. The **Replace** page has all of the **Find** page fields, with the addition of the *Replace with* field. The **Replace** page, *Method* field has only two options.



*Use the **Find and Replace** dialog to replace numbers or text in the worksheet.*

Replace With

Type the text you want to replace in the *Find* box. To delete the characters in the *Find* box from your worksheet, leave the *Replace with* box blank. Click the arrow at the right to select from a list of the most recently searched items.

Method

Choose the search *Method* from the list to determine how the search is performed. The examples assume "Golden, CO" is in the *Find* field.

- Select *Cell matches target exactly* to require that the exact criteria in the *Search* box is present in a cell before it is selected. For example, only cells that have exactly "Golden, CO" will be selected.
- Select *Cell contains target phrase* to require that the phrase in the *Search* box is present in a cell before it is selected. For example, cells that has "Golden CO", "Golden Company", or "Golden Colorado" will be selected.

Replace Button

Click the *Replace* button to replace the selected occurrence of the criteria in the *Find* box with the criteria in the *Replace with* box, find the next occurrence of the criteria in the *Find* box, and then stop. If you want to automatically replace all occurrences of the search criteria in the worksheet, click the *Replace All* button.

Replace All Button

Click the *Replace All* button to replace all occurrences of the *Find* criteria in your document with the *Replace with* criteria. If you want to review and selectively replace each occurrence, click the *Replace* button.

Find Next

Clicking the **Edit | Find Next** command, clicking the  button, or pressing the F3 key on the keyboard, finds the next instance of a particular number, word, or phrase in the worksheet. Each cell matching the search parameters remains selected. If the **Edit | Find** command was not used initially, the **Find and Replace** dialog opens so that you can define your search criteria.

Worksheet Formatting

Format Menu Commands

The worksheet **Format** menu has the following commands:

Format Cells	Sets the numeric format, alignment, and background color for the selected cells
Column Width	Sets column widths for selected cells
Row Height	Sets row height for selected cells

Format Cells

Cell numbers, alignment, or background color can be formatted through the **Format Cells** dialog. To format a cell, select the cells to be formatted (see *Selecting Worksheet Cells*), and click the **Format | Format Cells** command. The **Format Cells** dialog opens.

The **Format Cells** dialog has three pages: **Number**, **Alignment**, and **Background**.

Number Page

Use the **Number** page to change the numeric data display in the worksheet.

Alignment Page

Use the **Alignment** page to set the cell alignment.

Background Page

Use the **Background** page to set cell background color.

Text String

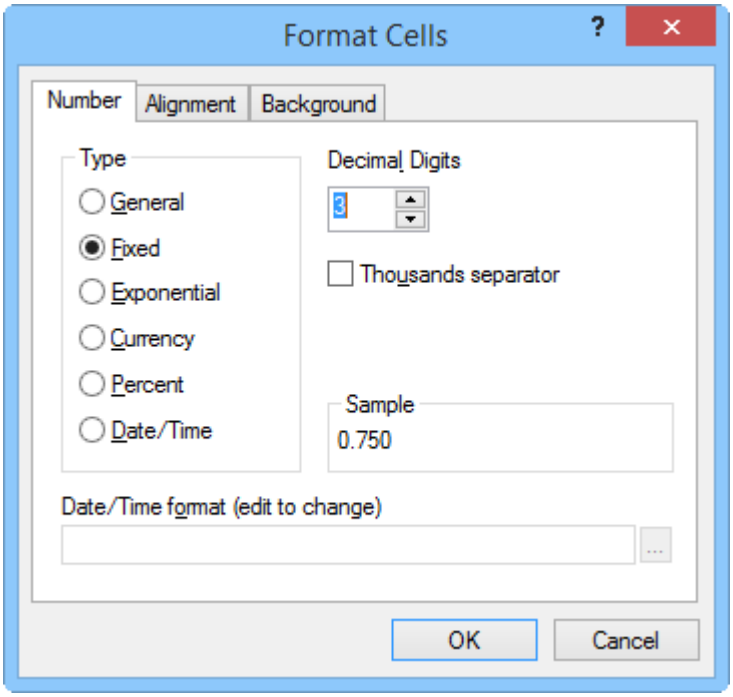
Number formatting has no effect on a numeric text string (numbers entered as text). A number with an apostrophe in front of it ('8123) is a text string. The apostrophe only shows in the active cell edit box. For example, an ASCII data file might contain the digits "8123" (digits surrounded by quotes), '8123 (digits preceded with an apostrophe), numbers with letters, or numbers with symbols (i.e. a blackslash "\"). These "numbers" are read as text and not as a number. The **Data | Transform** command can be used to perform a mathematical function, such as ATOI(X), to convert a text string to an integer value. Alternatively, the cell can be converted to numbers with the **Data | Text to Number** command.

Preserving Cell Format

The only formats that preserve cell-formatting information are Excel or SYLK SLK. ASCII file formats (.CSV, .TXT, .DAT, .BNA, .BLN) do not preserve file format information.

Format Cells - Number

Cell numbers, alignment, or background color can be formatted through the **Format Cells** dialog. To format a cell, select the cells to be formatted (see *Selecting Worksheet Cells*), and then click the **Format | Format Cells** command or click the  button. Use the **Number** page to change the numeric data display in the worksheet. Number formatting has no effect on a numeric text string (numbers entered as text). For example, an ASCII data file might contain the numbers "8123" (numbers surrounded by quotes) which are read as text and not as a number. The **Data | Transform** command can be used to perform a mathematical function, such as ATOI(X), to convert a text string to an integer value. Alternatively, the cell can be converted to numbers with the **Data | Text to Number** command.



*Select the number format options on the **Number** page of the **Format Cells** dialog.*

Type

The *Type* section contains the numeric format for the selected cells. Available options are *General*, *Fixed*, *Exponential*, *Currency*, *Percent*, and *Date/Time*. Click on the desired option.

- *General* displays numbers as fixed or exponential, whichever is shorter.
- *Fixed* displays numbers as d.ddd. The number to the left of the decimal can vary. Set the number to the right of the decimal in the *Decimal Digits* box.
- *Exponential* displays numbers as d.ddde+dd. Set the number of digits to the right of the decimal in the *Decimal Digits* box.
- *Currency* displays fixed numbers with a currency symbol such as the dollar sign ($).
- *Percent* displays numeric values (such as 0.13) as percentages with a percent symbol suffix (13%).
- *Date/Time* formats the cells as date and/or time. Select *Date/Time* and then type or select the *Date/Time format*.

Decimal Digits

The *Decimal Digits* controls the number of digits to the right of the decimal when the *Type* is set to *Fixed*, *Exponential*, *Currency*, or *Percent*. To change the *Decimal Digits*, highlight the existing value and type a new value. Alternatively, click the  to increase or decrease the value.

Thousands Separator

The *Thousands separator* option controls whether a comma appears in the number, indicating thousands. When checked, a comma appears every three digits to the left of the decimal point. When unchecked, the number appears without the comma. Do not type a comma when entering data as this causes the number to be read as text.

Sample

The *Sample* box displays the current number format.

Date/Time Format

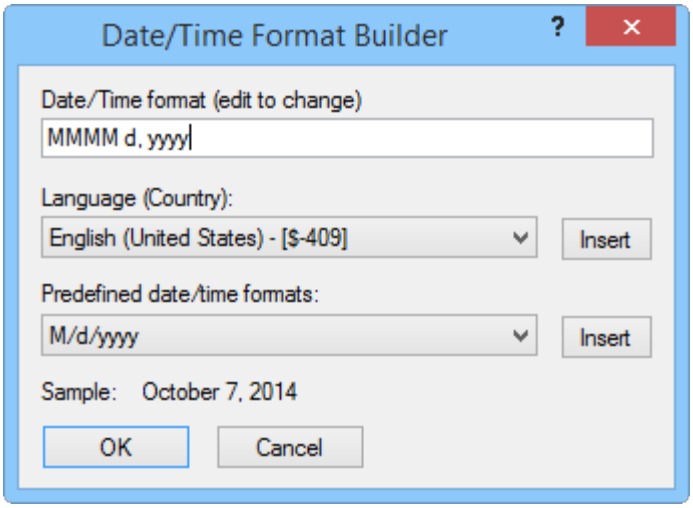
When the *Type* is set to *Date/Time*, the *Date/Time format* option becomes available. Type the desired format into the *Date/Time format* field, or click the  button to insert a date/time format with the **Date/Time Format Builder** dialog. Available formats are made of combinations of day, month, year, AD/BC or CE/BCE designation, hours, minutes, seconds, and AM/PM designation. Days are shown as d or dd. Months are shown as M, MM, MMM, MMMM, or MMMMM. Years are shown as yy or yyyy. Hours are shown as h or hh. Minutes are shown as m or mm. Seconds are shown as ss. Sub-seconds are displayed as ss.0 or ss.000. AM/PM designation are shown as tt or TT. BC/AD designation is shown as gg or GG. BCE/CE designation is shown as g, ggg, G, or GGG. Total elapsed time is shown as [h]. After clicking on the *Date/Time format*, the *Sample* updates to show a value in the selected format.

OK or Cancel

Click *OK* to make the change to the cell format. Click *Cancel* to return to the worksheet without making the change.

Date/Time Format Builder Dialog

From the worksheet, click the ... button in the **Format Cells** dialog **Number** page to open the **Date/Time Format Builder** dialog.



*The **Date/Time Format Builder** dialog is used to insert or create date/time formats for worksheet cells or text objects.*

Date/Time Format

Type a *Date/Time Format* into the *Date/Time format (edit to change)* field to set the date/time format. You can also use the *Language (Country)* and *Predefined date/time formats* lists to insert multiple date/time formats and languages.

Language (Country)

By default, the program will use the computer's default language settings for displaying the date/time options in the worksheet. The computer default is controlled by the Windows Control Panel. Refer to your Windows documentation for information about setting the locale. The *Language (Country)* uses the same codes to override the display. For instance, if the date/time values should always be displayed in English, regardless of locale, you could select *English (United States) - [$-409]* and click the *Insert* button. Insert the locale setting first in the *Date/Time format* box. Any cells with the specified language will appear in that language. In addition, the options in the *Predefined date/time formats* will change to show the common formats for that locale. Locale IDs are input as [$-####] in the *Date/Time format* field, where the #### is the locale identifier.

Note: The *Insert* button must be clicked after selecting the *Language (Country)* option. Simply selecting the *Language (Country)* does not change the *Date/Time format*. The *Date/Time format* does not change until *Insert* is clicked.

Predefined Date/Time Formats

The *Predefined date/time formats* list contains the common formats for the selected *Language (Country)* option or for your Windows locale. Available formats are made of combinations of year, month, day, hours, minutes, seconds, and AM/PM designation. Years are shown as yy or yyyy. Months are shown as M, MM, MMM, MMMM, or MMMMM. Days are shown as d, dd, ddd, or dddd. Hours are shown as h, hh, H, HH, or [h]. Minutes are shown as m, mm, or [mm]. Seconds are shown as ss, ss.0, ss.00, ss.000, ss.0000, or [ss]. AM/PM designation is shown as tt or TT. BC/AD designation is shown as gg or GG. BCE/CE designation is shown as g, G, ggg, or GGG. Refer to formats for information about each specific option.

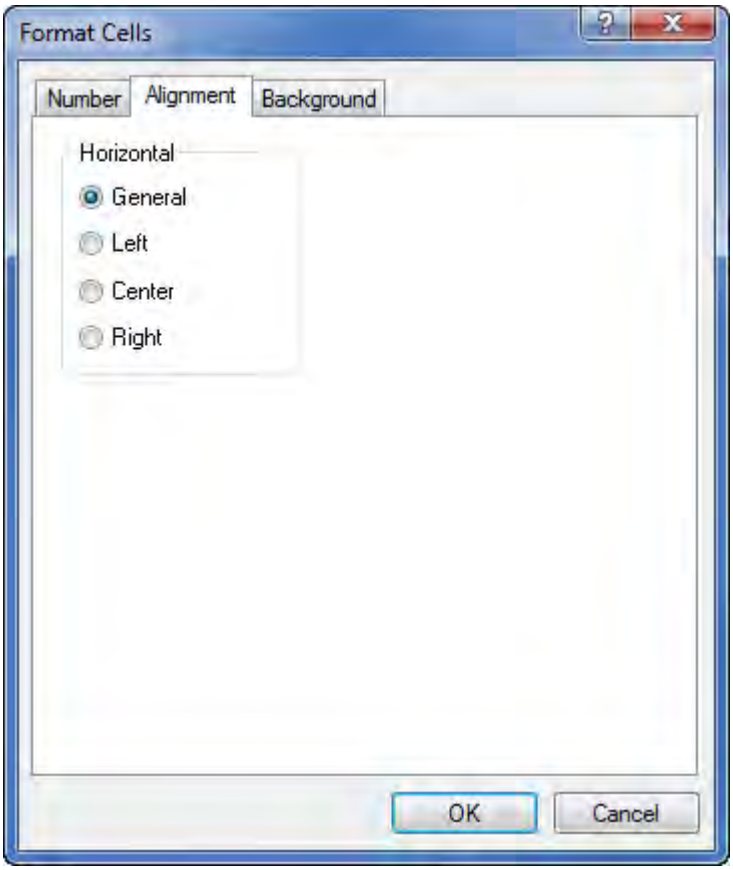
Note: The *Insert* button must be clicked after selecting the *Predefined date/time formats* option. Simply selecting the *Predefined date/time formats* does not change the *Date/Time format*. The *Date/Time format* does not change until *Insert* is clicked.

Sample

The *Sample* text updates to show a sample of the current entry in the *Date/Time format (edit to change)* field.

Format Cells - Alignment

Cell numbers, alignment, or background color can be formatted through the **Format Cells** dialog. To format a cell, select the cells to be formatted (see *Selecting Worksheet Cells*), and then click the **Format | Format Cells** command or click the button. Use the **Alignment** page to align the cell in one of four ways. By default, imported ASCII files automatically align numbers to the right and text to the left.



*Select the alignment options on the **Alignment** page of the **Format Cells** dialog.*

General

General aligns text on the left side of the cell and numbers on the right side of the cell.

Left

Left aligns text and numbers with the left side of the cell.

Center

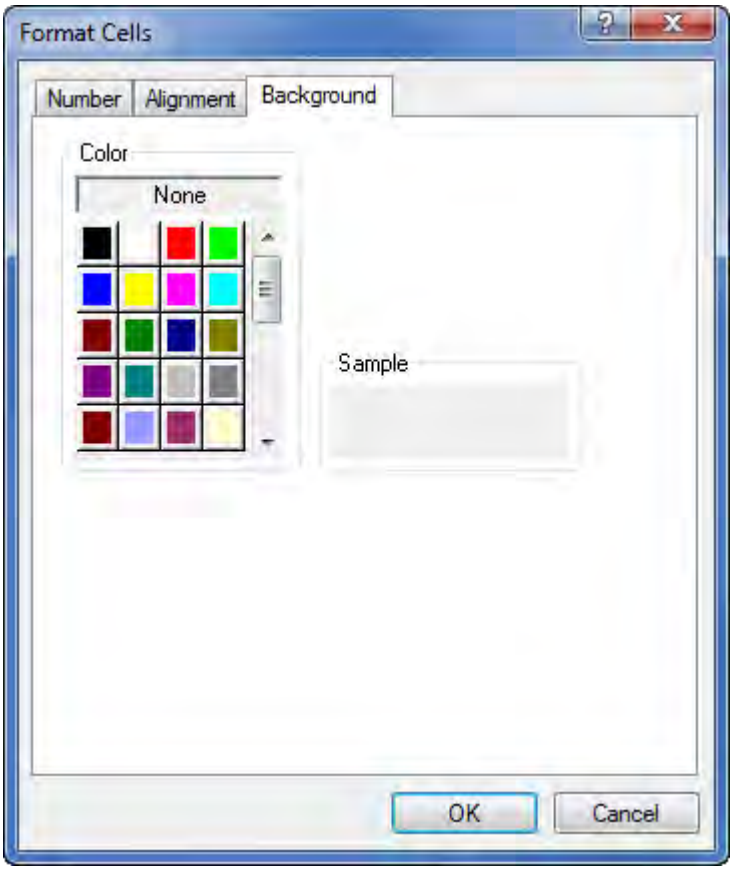
Center aligns text and numbers with the center of the cell.

Right

Right aligns text and numbers with the right side of the cell.

Format Cells - Background

Cell numbers, alignment, or background color can be formatted through the **Format Cells** dialog. To format a cell, select the cells to be formatted (see *Selecting Worksheet Cells*), and then click the **Format | Format Cells** command or click the  button. You can set cell background color on the **Background** page. Save the worksheet in Excel format to save background color in the file.



*Select the cell background color the **Background** page of the **Format Cells** dialog.*

None

Click the *None* button to remove any previously assigned background colors.

Color Palette

Select a cell background color from the color palette.

Sample

A sample of the color is displayed in the *Sample* box.

Date Time Formats

Date and time formats can be set from the worksheet. Date and time options are case sensitive.

When dates are parsed during input/import, the month and day of week names must match those of the local language as set in the Windows Control Panel, otherwise the entry will not be recognized as a valid date and will be treated as a text string.

Dates that are not in the same format as the local language setting in the Windows Control Panel can be parsed correctly by specifying the *Locale* settings in the **Data Import Options** dialog. Access the **Data Import Options** dialog by importing a TXT file with the worksheet **File | Import** command. Alternatively, you can copy the dates in **Voxler** or another program, then use the **Edit | Paste Special** command and select the *Text [Clipboard]* or *Unicode Text [Clipboard]* options.

Date/Time formats are made of combinations of locale, year, month, day, hours, minutes, seconds, BC/AD or BCE/CE designation, and AM/PM designation. Years are shown as yy or yyyy. Months are shown as M, MM, MMM, MMMM, or MMMMM. Days are shown as d, dd, ddd, or dddd. Hours are shown as h, hh, H, HH, or [h]. Minutes are shown as m, mm, or [mm]. Seconds are shown as ss, ss.0, ss.00, ss.000, ss.0000, or [ss]. AM/PM designation is shown as tt or TT. BC/AD designation is shown as gg or GG. BCE/CE designation is shown as g, G, ggg or GGG.

To add new date/time designations, use any combination of the following codes:

d	9	Single digit day, excluding leading zero
dd	09	Double digit day, including leading zero
ddd	Wed	Shortened day of week name
dddd	Wednesday	Full day of week name
M	7	Single digit month, excluding leading zero
MM	07	Double digit month, including leading zero
MMM	Jul	Shortened month name
MMMM	July	Full month name
MMMMM	J	First letter of month name
yy	98	Two digit year
yyyy	1998	Full year
g		Before Common Era designator - Includes space and bce or nothing if ce, lower case
gg	ad	BC/AD designator - Includes space and bc or ad, lower case
ggg	ce	Before Common Era designator - Includes space and bce or ce, lower case
G		Before Common Era designator - Includes space and BCE or nothing if CE, upper case
GG	AD	BC/AD designator - Includes space and BC or AD, upper case

GGG	CE	Before Common Era designator - Includes space and BCE or CE, upper case
h	6	Single digit hours - 1-12, excluding leading zero
hh	06	Double digit hours - 01-12, including leading zero
H	18	Hours - 0-23 military, excluding leading zero
HH	18	Hours - 00-23 military, including leading zero
[h]	1003914	Hours portion of total time, excludes leading zeros
m	45	Minutes - 0-60, excluding leading zero
mm	45	Minutes - 00 to 60, including leading zero
[mm]	45	Minutes portion of total time, includes leading zeros
ss	44	Seconds - 0-60, rounded to the nearest second
ss.0	44.1	Seconds - 0-60, rounded to the nearest tenth of a second
ss.00	44.12	Seconds - 0-60, rounded to the nearest hundredth of a second
ss.000	44.123	Seconds - 0-60, rounded to the nearest millisecond
ss.0000	44.12345	Seconds - 0-60, maximum precision
[ss]	44	Seconds portion of total time, includes leading zeros
tt	pm	am or pm designator, lower case
TT	PM	AM or PM designator, upper case
\		escape character - output next character verbatim
'...'		output ALL characters between single quotes verbatim, including escape character
[$-xxxx]	[$-409]	xxxx is an up to four hex digit representation of a locale ID

Custom Date/Time Example

mm/dd/yy h:mm:ss tt	Month double digits, Day double digits, Year double digits, Hour in standard format, Minutes, Seconds and AM/PM designation	04/14/09 6:45:44 PM

When dates are parsed during input/import, the month and day of week names must match those of the local language as set in the Windows Control Panel, otherwise the entry will not be recognized as a valid date and will be treated as a text string.

When the recognized format is ambiguous (i.e. 10/7/12), the month, day, and year order is determined by the Windows locale. In some countries, this will be recognized as M/d/yy, in others as d/M/yy, and in others as YY/M/d. It is important to use non-ambiguous date/time formats when the Windows locale may change.

The tables below show many examples of date/time format strings.

Date Formats

All rows below use the date September 7, 1998 for the *Example*.

Date/Time Code	Example	Description
(None)		Date not displayed
M/d/yy	9/7/98	Single digit month and day, two digit year, separated with /
MM/dd/yy	09/07/98	Double digit month, day, and year, separated with /
M/d/yyyy	9/7/1998	Single digit month and day, full year, separated with /
MMM dd, yyyy	Sep 07, 1998	Shortened month name, double digit day, full year, separated with spaces and comma
MMMM dd, yyyy	September 07, 1998	Full month name, double digit day, full year, separated with spaces and comma
MMMM-d-yyyy	September-7-1998	Full month name, single digit day, full year, separated with -
d MMMM yyyy	7 September 1998	Single digit day, full month name, full year, separated with spaces
d-MMM-yy	7-Sep-98	Single digit day, shortened month name, two digit year, separated with -
dd-MMM-yy	07-Sep-98	Double digit day, shortened month name, two digit year, separated with -
d-MMM-yyyy	7-Sep-1998	Single digit day, shortened month name, full year, separated with -
d-MMM	7-Sep	Single digit day, shortened month name, separated with -
MMM-yy	Sep-98	Shortened month name, two digit year, separated with -
MMM-yyyy	Sep-1998	Shortened month name, full year, separated with -
MMMM-yy	September-98	Full month name, two digit year, separated with -
MMMM-yyyy	September-1998	Full month name, full year, separated with -
MM-dd-yy	09-07-98	Double digit month and day, two digit year, separated with -
yyyy	1998	Full year
yyyy gg	1998 ad	Full year with lowercase bc/ad designation
yyyy GGG	1998 CE	Full year with uppercase BCE/CE designation
yy	98	Two digit year
MMMMM	S	First letter of month name

MMMM	September	Full month name
MMM	Sep	Shortened month name
MM	09	Double digit month
M	9	Single digit month
MMMMM-yy	S-98	First letter of month name, two digit year, separated with -
MMM-d	Sep-7	Shortened month name, single digit day, separated with -
M/d	9/7	Single digit month and day, separated with /
dddd	Monday	Full day of week name
ddd	Mon	Shortened day of week name
dd	07	Double digit day
d	7	Single digit day
d/M/yy	7/9/98	Single digit day and month, two digit year, separated with /
d.M.yy	7.9.98	Single digit day and month, two digit year, separated with .
dd/MM/yy	07/09/98	Double digit day and month, two digit year, separated with /
dd/MM/yyyy	07/09/1998	Double digit day and month, full year, separated with /
yy/MM/dd	98/09/07	Two digit year, double digit month and day, separated with /
yyyy-MM-dd	1998-09-07	Full year, double digit month and day, separated with -

Time Formats

All rows below use the time 2:45:44.12 PM for the *Example*.

Date/Time Code	Example	Description
(None)		Time not displayed
h:mm tt	2:45 PM	Hour in 0-12 (standard format), two digit Minutes 00 to 60, then a space and AM or PM
h:mm	14:45	Hour in 0-23 (military time), two digit Minutes 00 to 60
hh:mm	14:45	Two digit Hour 00-23 (military time), two digit Minutes 00 to 60
h:mm:ss tt	2:45:44 PM	Hour in 0-12 (standard format), two digit Minutes 00 to 60
h:mm:ss	14:45:44	Hour in 0-23 (military time), two digit Minutes 00 to 60, two digit Seconds 00 to 60

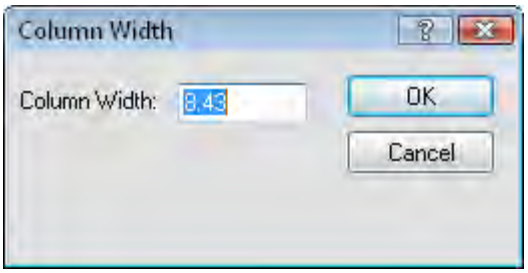
hh:mm:ss	14:45:44	Two digit Hour 00-23 (military time), two digit Minutes 00 to 60, two digit Seconds 00 to 60
m:ss	45:44	Single digit Minutes 0 to 60, two digit Seconds 00 to 60
mm:ss	45:44	Two digit Minutes 00 to 60, two digit Seconds 00 to 60
m:ss.0	45:44.1	Single digit Minutes 0 to 60, two digit Seconds 00 to 60, fractional seconds rounded to the nearest tenth of a second
mm:ss.0	45:44.1	Two digit Minutes 00 to 60, two digit Seconds 00 to 60, fractional seconds rounded to the nearest tenth of a second
h:mm:ss.000	14:45:44.12	Hour in 0-23 (military time), two digit Minutes 00 to 60, two digit Seconds, 00 to 60, fractional seconds with full precision
m:ss.000	45:44.12	Single digit Minutes 0 to 60, two digit Seconds 00 to 60, fractional seconds with full precision
mm:ss.000	45:44.12	Two digit Minutes 00 to 60, two digit Seconds 00 to 60, fractional seconds with full precision
[h]:mm:ss	865094:45:44	Total hours (day value plus hour value), two digit Minutes 00 to 60, two digit Seconds 00 to 60. Example Explanation: Date value 865080 = September 7, 1998 Hour value = 14, added to 865080 = 865094

Column Width

You can change the column width of selected cells by clicking the **Format | Column Width** command or by using the mouse to resize the column. The Excel XLS or SYLK SLK file format must be used to save the column width in the file since ASCII file formats (.CSV, .TXT, .DAT, .BNA, .BLN) do not preserve file format information.

The Column Width Dialog

To set column widths or to hide columns, select either the entire column or individual cells within the columns (see *Selecting Worksheet Cells*), and then click the **Format | Column Width** command or click the  button. Enter the width for the selected column or cells into the **Column Width** dialog. Columns can range from zero to 512 characters wide. The value zero (0) hides the column.



*Change the column width by selecting columns, clicking Format | Column Width, and then entering a number into the **Column Width** dialog.*

Changing Column Widths with the Mouse

Column width can also be changed using the mouse. When the cursor is moved to the line that defines the right boundary of the column header, the cursor changes to a ↔. Press and hold the left mouse button and move the cursor to the left or right to change the width of the column.

Hide a Column with the Mouse

You can hide a column by moving the cursor to the left until the next dividing line is reached.

Display Hidden Columns with the Mouse

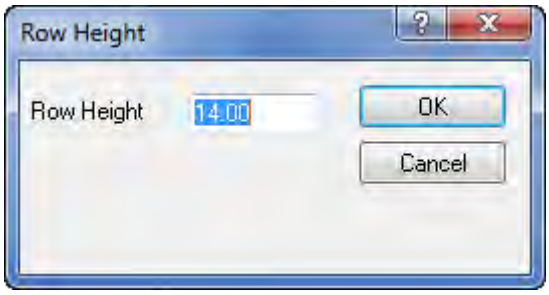
To display hidden columns, press and hold the left mouse button at the right edge of the hidden column and move the cursor to the right to widen the column.

Row Height

You can change the **Row Height** of selected cells by clicking the **Format | Row Height** command or by using the mouse to size the row. The Excel XLS or SYLK SLK file format must be used to save the row height and numeric format information with the file since ASCII file formats (.CSV, .TXT, .DAT, .BNA, .BLN) do not preserve file format information.

The Row Height Dialog

To set the row height or hide rows, select either the entire row or individual cells within the rows (see *Selecting Worksheet Cells*), and click the **Format | Row Height** command or click the button. Enter the height for the selected row or cells in the **Row Height** dialog. Rows can range from zero to 512 points in height. A value of zero (0) hides the row.



*Change the row height by selecting rows, choosing **Format | Row Height**, and then entering a number into the **Row Height** dialog.*

Changing Row Heights with the Mouse

Row height can also be changed using the mouse. When the cursor is moved to the line that defines the lower boundary of the row header, the cursor changes to a line with two arrows . Press and hold the left mouse button and move the cursor up or down to change the height of the row.

Hide a Row with the Mouse

You can hide a row by moving the cursor up until the next dividing line is reached.

Display Hidden Row with the Mouse

To display hidden rows, press and hold the left mouse button at the bottom of the hidden row and move the cursor down to stretch the row height.

Data Menu Commands

Selected cells of data are sorted or transformed using the worksheet **Data** menu commands. The **Data** menu also contains a statistics option. Various statistical results generated from the data can be saved in the worksheet. Coordinate and projection information is assigned with **Data** menu commands.

The worksheet **Data** menu has the following commands:

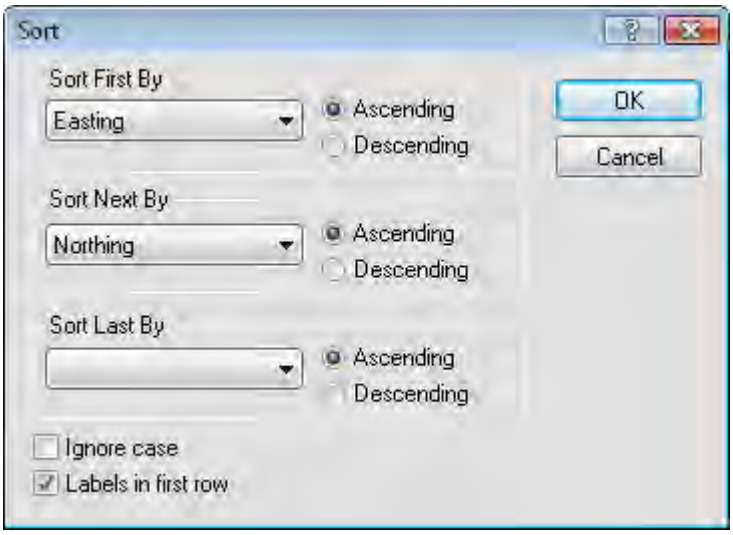
Sort	Sorts selected cells
Transform	Applies a mathematical transform to columns
Statistics	Computes statistics on selected cells
Text to Number	Convert the text in selected cells to numbers
Transpose	Convert columns to rows and rows to columns
Assign Coordinate System	Specifies the existing coordinate system
New Projected Coordinates	Specifies the columns containing the source X, Y coordinates and target X, Y columns for a new coordinate system

Sort

The **Data | Sort** command arranges data according to rank in user-specified sort columns. Sorting rank is based on numbers, ASCII characters, and punctuation. Sort numeric data, text, or mixed columns. Sorting specifications are made in the **Sort** dialog.

The Sort Dialog

Click the **Data | Sort** command in the worksheet to open the **Sort** dialog.



*Click the **Data | Sort** command to sort data on multiple columns.*

Selecting Cells to Sort

Sorting is performed only on the selected columns (see *Selecting Cells*). If only one column is selected, only that column is sorted. To keep records (rows of data) together, select all columns containing data even if only one column is sorted. To decrease sort time, select a block of cells rather than clicking on the row or column labels.

Sort Order

The *Sort First By* option defines the primary column on which the rows are sorted. The positions of the sorted rows are determined by the *Ascending* or *Descending* rank in the *Sort First By* column.

Secondary Sort

When two or more rows have identical entries in the *Sort First By* column, the *Sort Next By* column can further organize the data set. Duplicates in the *Sort First By Column* are then sorted according to the rank in the *Sort Next By* column.

Final Sort

The *Sort Last By* column can be used when the *Sort Next By* column contains duplicates.

Ascending or Descending Sort

The sort order in an *Ascending* sort is based on the ASCII table. Numeric values are placed first, followed in order by cells starting with a space character, common punctuation, numeric text (numbers entered as text), uppercase letters, less common punctuation, lower case letters,

uncommon punctuation, and blank cells. *Descending* order is the opposite of ascending order although blank cells are still listed last.

0	1	2	3	4	5	6	7	8	9
space	!	"	#	$	%	&	'	(	)
*	+	,	-	.	/	"0"	"1"	"2"	"3"
"4"	"5"	"6"	"7"	"8"	"9"	:	;	<	=
>	?	@	A	B	C	D	E	F	G
H	I	J	K	L	M	N	O	P	Q
R	S	T	U	V	W	X	Y	Z	[
\	]	^	_	`	a	b	c	d	e
f	g	h	i	j	k	l	m	n	o
p	q	r	s	t	u	v	w	x	y
z	{	|	}	~	blank				

This ASCII table shows the sort order in the worksheet.

Ignore Case

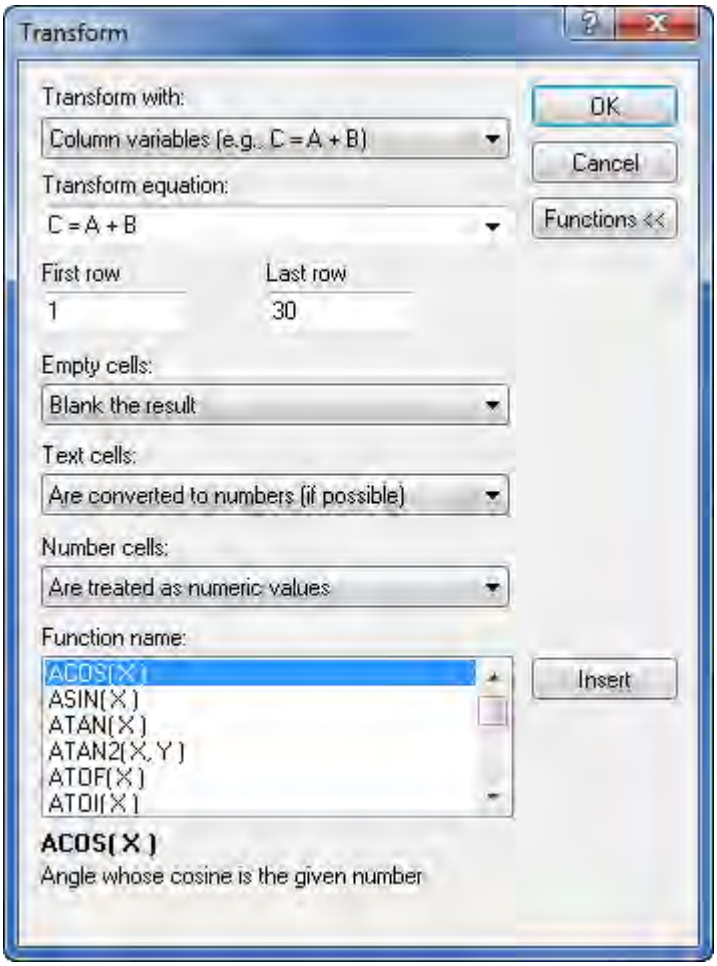
Because sorting is based on an ASCII table, upper and lowercase letters are treated differently. For example, "A" is sorted separately from "a." If the letters are to be treated as the same during the sort, check the *Ignore case* option. When this check box is activated, "A" is considered identical to "a" in the sorting rank.

Labels in First Row

The data set may contain text identifying the data in the column (header information) in Row 1. In this case, click the *Labels in first row* option to exclude the label row from the sort process.

Transform

Click the **Data | Transform** command to open the **Transform** dialog, where you can apply mathematical transformations to the columns, rows, or cells. Valid math operators include addition (+), subtraction (-), multiplication (*), and division (/) as well as a large library of built-in mathematical functions. Parentheses should be used to override precedence or for clarification.



*Use the **Transform** dialog to apply math functions to data. The dialog options update to reflect the option selected for Transform with field.*

Transform With

Select the type of transform from the *Transform with* list. *Column variables (e.g., C = A + B)* applies the transform equation to the specified rows in the *Transform equation* column. *Row variables (i.e., _3 = _1 + _2)* applies the transform equation to the specified columns in the *Transform equation* row. *Cell variables (i.e., C3 = A1 + B2)* applies the transform equation only to the cell specified in the *Transform equation*.

Transform Equation

Type the formula into the *Transform equation* box. Formulas consist of a destination column, row, or cell on the left side of the equation and a mathematical manipulation on the right side of the equation. Use the column label letters, row numbers, or cell locations on both sides of the equation. Click the down arrow to use previously entered equations. For columns, a sample equation may be

C = A + B. For rows, a sample equation is _4=_1+_2. For cells, a sample equation would look like C2=A1+B1-C1.

If the transform method is by column, the range functions (sum, avg, std, rowmin and rowmax) take column indices only, i.e., sum(A...C). If transform method is by variable rows, the range functions take row indices only, i.e., sum(_1..._3). If transform method is by variable cells, the range functions are not supported.

The last ten functions are stored in the *Transform equation* field. After ten functions are included in the list, the oldest function is replaced when a new function is added. The *Transform equations* are stored between sessions. To use a stored function, click the  in the *Transform equation* box and select a function from the list. Note that the *First row* and *Last row* or *First col* and *Last col* values are not saved with the stored *Transform equations*.

First and Last Columns and Rows

When calculating transformations on columns, enter the *First row* and the *Last row* to limit the calculation to the specified rows. When calculating transformations on rows, enter the *First col* and *Last col* to limit the calculation to the specified columns. When calculating transformations on cells, the *First row*, *Last row*, *First col*, and *Last col* options are not available.

By default, these are set to the first row and last row (or first column and last column) with text or numbers entered into a cell for the entire worksheet.

Empty Cells

The *Empty cells* option controls how empty cells are treated in the calculations of formulas. Available options are *Blank the result*, *Are treated as the number zero (0)*, and *Are treated as empty text ("")*. The default option is *Blank the result*, which results in the formula not being calculated for any row that contains a blank cell in any of transform equation rows or columns.

- Setting the *Empty cells* option to *Blank the result* results in a blank cell for the transform when the cells on the right side of the equation are empty.
- Setting the *Empty cells* option to *Are treated as the number zero (0)* results in the transform creating a number when all of the cells on the right side of the equation are empty or numeric. When the right side of the equation combines text and blank cells, the equation is blank.
- Setting the *Empty cells* option to *Are treated as empty text ("")* results in the transform creating a text string when all of the cells on the right side of the equation are empty or text. When the right side of the equation combines numeric and blank cells, the equation is blank.

Text Cells

The *Text cells* option controls how text cells are treated in the calculations of formulas. Available options are *Blank the result*, *Are treated as text*, *Are converted to numbers (if possible)*, and *Are treated as the number zero (0)*. The default option is *Blank the result*, which results in the formula not being calculated for any row that contains a text cell in any of transform equation rows or columns.

- Setting the *Text cells* option to *Blank the result* results in a blank cell for the transform when any of the cells on the right side of the equation contain text strings (including numbers formatted as text).
- Setting the *Text cells* option to *Are treated as text* results in the transform creating a text string when all of the cells on the right side of the equation are text (or treated as text). If a mix of text cells and numbers or empty cells (that are not treated as text) are in the cells on

the right side of the equation, the transform results in a blank cell. This option allows text strings to be concatenated.

- Setting the *Text cells* option to *Are converted to numbers (if possible)* results in the transform creating a number when all of the cells on the right side of the equation are numeric or treated as numbers. Any cells with numbers formatted as text are treated as the number. For example, the text string '05 would be treated as the number 5 if this option is selected.
- Setting the *Text cells* option to *Are treated as the number zero (0)* results in the transform creating a number when all of the cells on the right side of the equation are numeric or treated as numbers. Any cells with text are replaced with the value zero for the transform. For example, if you are using the equation C=A+B and A has *Colorado* and B has *45*, the value in cell C will be 45.

Number Cells

The *Number cells* option controls how numeric cells are treated in the calculations of formulas. Available options are *Blank the result*, *Are treated as numeric values*, *Are converted to text*, and *Are treated as empty text ("").* The default option is *Are treated as numeric values*, which results in the formula being calculated for any row that contains numbers in any of transform equation rows or columns.

- Setting the *Number cells* option to *Blank the result* results in a blank cell for the transform when any of the cells on the right side of the equation contain numbers. This option is useful when you only want to combine text cells or blank cells.
- Setting the *Number cells* option to *Are treated as numeric values* results in the transform creating a number when all of the cells on the right side of the equation are number (or treated as numbers). If a mix of text cells and numbers or empty cells (that are not treated as numbers) are in the cells on the right side of the equation, the transform results in a blank cell.
- Setting the *Number cells* option to *Are converted to text* results in the transform creating a text string when all of the cells on the right side of the equation are text or treated as text. Any cells with numbers are treated as the text string of the number. For example, number 5 is in the cell, so the text string would appear as '5 if this option is selected.
- Setting the *Number cells* option to *Are treated as empty text ("")* results in the transform creating a text string when all of the cells on the right side of the equation are text or treated as text. Any cells with numbers are replaced with "" for the transform. For example, if you are using the equation C=A+B and A has *Colorado* and B has *45*, the value in cell C will be *Colorado*.

Combining Text, Numbers, and Empty Cells

Many possible combinations of the *Empty cells*, *Text cells*, and *Number cells* exist to allow combining these different types of cells in a *Transform equation*. If the transform result is not what you expect, check the settings for these options and adjust if necessary.

Functions

Click the *Functions* >> button to display a list of predefined mathematical functions. Click the *Functions* << button again to hide the list of predefined mathematical functions.

To use a function, place the cursor in the location to add a function, select a function from the list, click the *Insert* button, and then replace the X in the function with a column letter (A); underscore and row number (_1); or cell location (A1). Also, be sure to use proper mathematical operators (+_*/) between the function and the rest of the equation. The definition of the function is listed below the *Function name* list when a function is selected.

Insert

When the *Functions* are expanded, the *Insert* button is visible. Click the *Insert* button to add a function to the *Transform equation* box. In the *Transform equation* box, manually change the variable (i.e. X or Y) in the listed functions to a column letter, row number (_1), or cell location.

Examples

An example of a column formula is C = A + B. Columns A and B are added and inserted into column C with this equation. The formula adds the contents of A and B in each row and places the results in column C for that row.

An example of a row formula is _4=_1+_2. Rows 1 and 2 are added and inserted into row 4 with this equation. The formula adds the contents of the 1 and 2 in each column listed between the *First col* and *Last col* values and places the results in row 4 for that column.

An example of a cell formula is C2=A1+B1-C1. The value in C1 is subtracted from the sum of the values in cells A1 and B1. The result is inserted into cell C2 with this equation.

Example Functions

This example shows how to use the built in functions. Consider, for example, taking the cosine of data in column C. Column D is the first empty column, so we will use column D as the destination column.

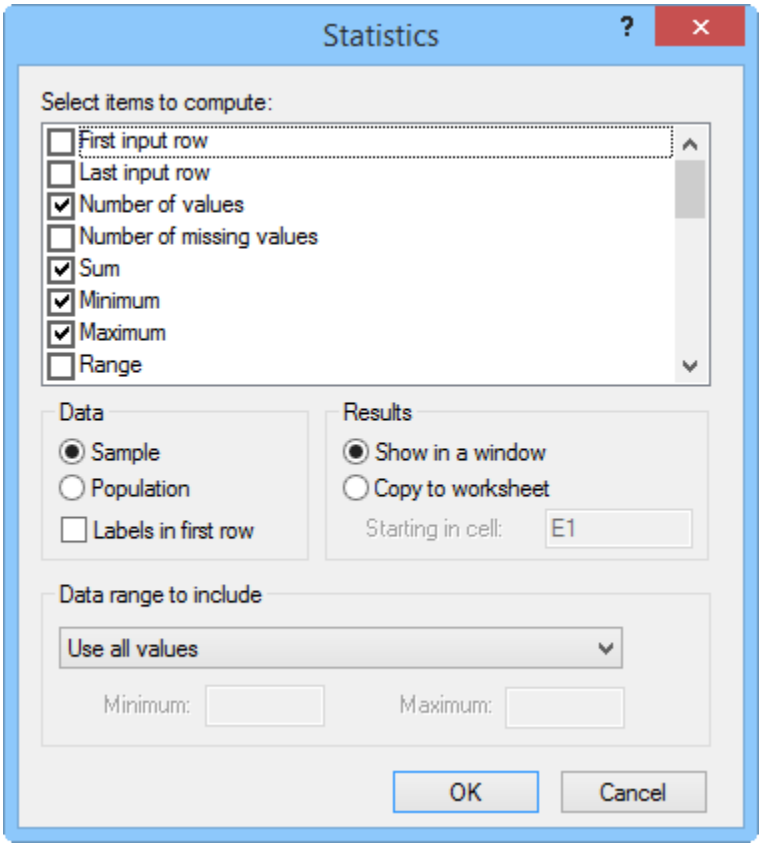
1. Click the **Data | Transform** command to open the **Transform** dialog. You do not need to highlight any columns before selecting **Transform**.
2. In the *Transform equation* box, type "D = " without the quotes.
3. Click the *Functions* button.
4. Double-click on the function name COS(X) in the *Function name* group. Alternatively, you could select a *Function name* and click the *Insert* button.
5. COS(X) is automatically placed in the equation as "D = COS(X)" without the quotes.
6. Replace the X in the function with the column letter containing the data to be transformed (column C). The equation will be "D = COS(C)" without the quotes.
7. Change the *First row* and *Last row* if you wish.
8. Make sure that *Empty cells* and *Text cells* are set to *Blank the result* to only calculate values with numbers.
9. Click *OK* to create a new data column with column C's data transformed with the cosine.

Statistics

The **Data | Statistics** command calculates statistical values for a group of selected numeric cells (see *Selecting Cells*). Select an entire column or a continuous group of cells in a column to use the **Statistics** command. If a rectangular block of rows and columns are selected, the **Statistics** command calculates the statistics for each column separately. A warning message appears if a group of cells cannot be used with the **Statistics** command. Non-numeric cell entries (empty cells or text) are ignored in statistics calculations.

The Statistics Dialog

Use the **Data | Statistics** command in the worksheet to open the **Statistics** dialog.



Click in the box adjacent to the statistics name to compute the statistics for the selected columns.

Select Items to Compute

Click in the check boxes next to the statistics options to calculate the statistics for the selected data:

- *First input row* reports the first row number in the selection. If the *Labels in first row* option is checked, the *First input row* is the second row in the selection.
- *Last input row* reports the last row number containing data in the column.
- *Number of values* indicates the number of numeric cells in the column.
- *Number of missing values* indicates the number of non-numeric cells in the selection. If columns are selected by clicking the column letters, the number of missing values includes blank values up to the last used row in the worksheet, which may be different than the last

used row in the selected column. If cells are selected by highlighting specific cells, then only the blank cells within the selection are counted.

- *Sum* is the sum of all numeric cells in the column.
- *Minimum* indicates the minimum value in the column.
- *Maximum* indicates the maximum value in the column.
- *Range* indicates the range of the numeric values in the column (*Maximum – Minimum*).
- *Mean* is the arithmetic average of the data values. It is the sum of the data values divided by the number of data values.
- *Median* is the middle value among the data values. Half of the data values are larger than the median and half are smaller than the median. When there are an even number of data values the median is the average of the two middle values.
- *Mode* is the value that appears most often in a data set. If the data set contains multiple modes, the modes will be displayed separated by a comma. #N/A will be displayed if no mode exists for the data set, i.e. there is an equal number of each data value.
- *First quartile (25th percentile)* is the value such that one-fourth of the data values are smaller than the quartile and three-fourths of the data values are larger than the first quartile.
- *Third quartile (75th percentile)* is the value such that three-fourths of the data values are smaller than the quartile and one-fourth of the data values are larger than the third quartile.
- *Standard error of the mean*
- *95% confidence interval for the mean*
- *99% confidence interval for the mean*
- *Variance*
- *Average deviation*
- *Standard deviation*
- *Coefficient of variation*
- *Coefficient of skewness*
- *Coefficient of kurtosis*
- *Kolmogorov-Smirnov goodness of fit for normal distribution*
- *Critical value of K-S statistic at 90% significance level*
- *Critical value of K-S statistic at 95% significance level*
- *Critical value of K-S statistic at 99% significance level*

The Data Group

The *Data* group is used to select *Sample* or *Population* statistics. Select *Sample* or *Population* statistics, depending on whether the data represent a statistical sample or the complete set of all possible members of a population.

The *Labels in first row* option is also specified in the *Data* group. Check the *Labels in first row* box if the first row of the selection contains descriptive labels. If this box is checked the label appears at the top of the statistics report for each column.

Results Group

The *Results* group is used to show the statistics report in a window or copy the results to a new location of the worksheet. Select *Show in a window* to write the statistics results to a **Statistics Results** dialog. The results in this dialog can be copied to the clipboard to paste to other locations.

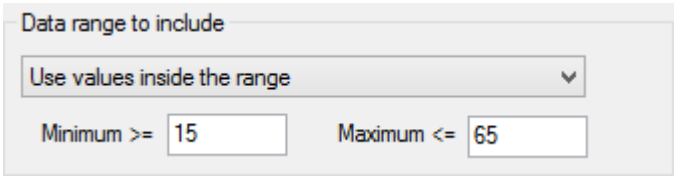
Select *Copy to worksheet* to write the statics report to a new location in the worksheet. Use the *Starting in cell* box to specify the cell for the upper left corner of the statistics report. If the destination cells contain data, a warning is displayed that data will be overwritten. Click *OK* to overwrite the data, or click *Cancel* to set a new *Starting in cell* location.

Data Range to Include Group

The *Data range to include* contains options to limit the values where the statistics are calculated. Available options are *Use all values*, *Use values inside the range*, *Use values outside the range*, and *Use all values except*.

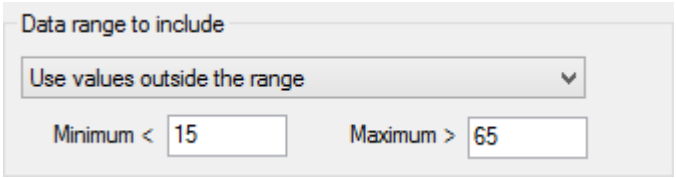
When the *Data range to include* is set to *Use all values*, all of the values in the highlighted section are used to calculate the statistics.

When the *Data range to include* is set to *Use values inside the range*, the *Minimum* >= and *Maximum* <= options are available. Type in the data values that bracket the range of values where the statistics should be calculated. For instance, if the *Minimum* >= is set to 15 and the *Maximum* <= is set to 65, only data points between (and including) 15 and 65 are used for calculating the statistics.



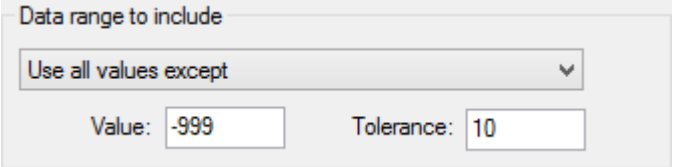
Only the values that are inside the range are included in the calculated statistics.

When the *Data range to include* is set to *Use values outside the range*, the *Minimum* < and *Maximum* > options are available. Type in the data values that bracket the range of values where the statistics should be calculated. For instance, if the *Minimum* < is set to 15 and the *Maximum* > is set to 65, only data points below 15 or greater than 65 (and excluding 15 and 65) are used for calculating the statistics.



Only the values that are outside the range are included in the calculated statistics.

When the *Data range to include* is set to *Use all values except*, the *Value* and *Tolerance* options are available. Type in the data value that should be excluded in the *Value* box. The *Tolerance* value gives a range on either side of the *Value*. Everything in the range *Value-Tolerance* to *Value+Tolerance* is excluded from the statistics calculation. For instance, if the *Value* is set to -999 and the *Tolerance* is set to 10, all values between -1009 and -989 are excluded from the statistics. This means that all values less than -1009 and greater than -989 are included in the statistics calculations.



Only the values that are outside the range Value-Tolerance to Value+Tolerance are included in the calculated statistics.

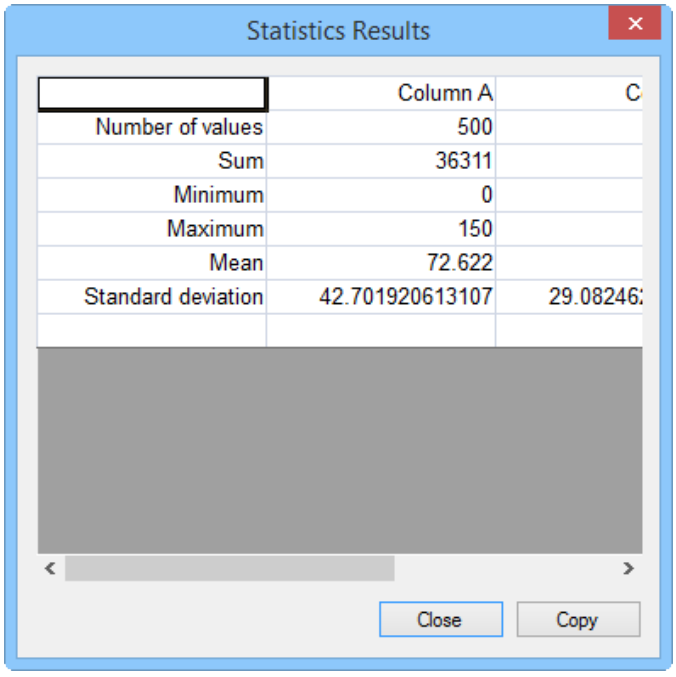
The *Use all values except* option can be used to ignore blanking values. For example, the default blanking value for grids generated in Golden Software's **Surfer** is 1.70141E+38. Two methods exist for ignoring the blanking value:

- One method is to enter the exact blanking value into the *Value* field of the **Statistics** dialog. Next set the *Tolerance* to 0.
- Another method is to enter an approximation of the blanking value and a tolerance that will include the actual blanking value while excluding the desired data. Using this method with the **Surfer** default value, 1.70E+38 is entered into the *Value* field and 1E+37 is entered into the *Tolerance* field.

Both of these methods result in the **Statistics** command returning results for the non-blanked values only.

Statistics Results Dialog

The **Statistics Results** dialog is displayed when the *Show in a window* option is selected in the *Results* section of the **Statistics** dialog.



*The **Statistics Results** dialog displays the results from the **Statistics** command.*

Click *Copy* to copy the results to the clipboard. The results can be pasted in another program. If you wish to paste the results into a **Voxler** worksheet, close the **Statistics Results** dialog first.

Click *Close* to close the **Statistics Results** dialog.

Statistics Definitions

95% and 99% Confidence Interval for the Mean

If CI is the value of the confidence interval reported by the worksheet, the range of values between the sample mean minus CI and the sample mean plus CI is expected to include the true mean of the underlying population 95% of the time (for the 95% confidence interval) or 99% of the time (for the 99% confidence interval). This formula assumes that the data set is sufficiently large for the central limit theorem to apply.

95% Confidence Interval for the Mean

$$\pm t_{(n-1),\alpha=.05}(SE)$$

99% Confidence Interval for the Mean

$$\pm t_{(n-1),\alpha=.01}(SE)$$

where

tv,α = the value of the Student's t distribution with v degrees of freedom such that difference between the cumulative probability function evaluated at tv,α and - tv,α is equal to 1- α.

SE = the standard error of the mean

Average Deviation

The average deviation is the average of the difference between the absolute values of data points and the mean.

Population Mean Deviation (MD)

$$MD = \frac{1}{N}\sum_{i=1}^{N}\left|(x_i - \mu)\right|$$

Sample Mean Deviation (MD)

$$MD = \frac{1}{n}\sum_{i=1}^{n}\left|(x_i - \bar{x})\right|$$

Definition of Population Mean (μ)

Definition of Sample Mean ($\bar{x}$)

where

N = number of data values for a population

n = number of data values for a sample

x_i = i^{th} data value

Coefficient of Kurtosis

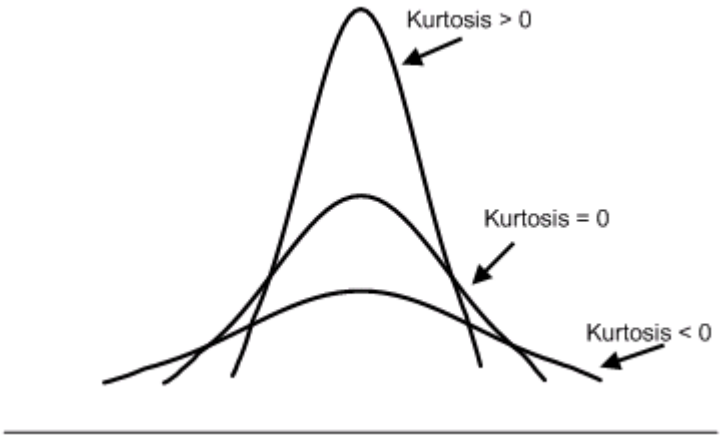
Kurtosis is a measure of how sharp the data peak is. Traditionally the value of this coefficient is compared to a value of 0.0, which is the coefficient of kurtosis for a normal distribution (i.e. the bell-shaped curve). A value greater than 0 indicates a peaked distribution and a value less than 0 indicates a flat distribution. Without a very large sample size, the use of this coefficient is of questionable value.

Population Kurtosis (γ_2)

$$\gamma_2 = \left(\frac{1}{N\,\sigma^4} \sum_{i=1}^{N} (x_i - \mu)^4 \right) - 3$$

Sample Kurtosis (g_2)

$$g_2 = \left\{ \frac{n\,(n+1)}{(n-1)(n-2)(n-3)} \sum \left(\frac{x_i - \bar{x}}{s} \right)^4 \right\} - \frac{3\,(n-1)^2}{(n-2)(n-3)}$$



(adapted from King and Julstrom, 1982)

where

σ = Population Standard Deviation

s = Sample Standard Deviation

μ = Population Mean

$\bar{x}$ = Sample Mean

N = number of data values for a population

n = number of data values for a sample

x_i = i^{th} data value

Coefficient of Skewness

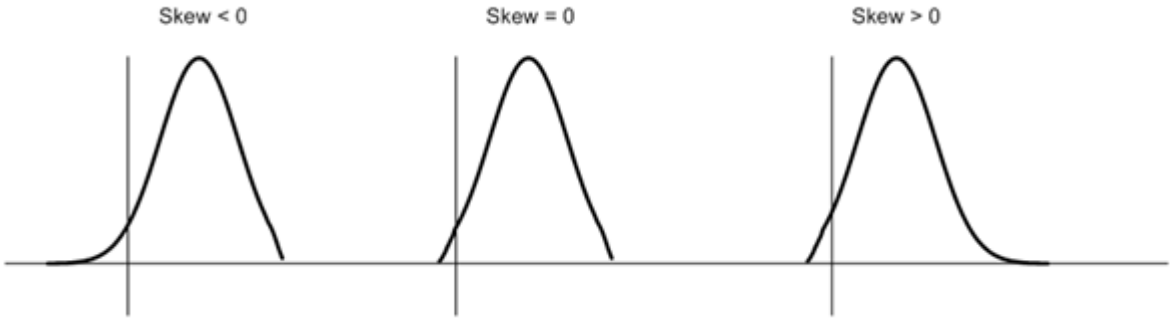
Skew is a measure of asymmetry in the distribution. A positive skew indicates a longer tail to the right, while a negative skew indicates a longer tail to the left. A perfectly symmetric distribution, like the normal distribution, has a skew equal to 0. For small data sets this measure is unreliable.

Population Skew (γ_1)

$$\gamma_1 = \frac{1}{N\,\sigma^3}\sum_{i=1}^{N}(x_i - \mu)^3$$

Sample Skew (g_1)

$$g_1 = \frac{n}{(n-1)(n-2)}\sum\left(\frac{x_i - \bar{x}}{s}\right)^3$$



(adapted from King and Julstrom, 1982 and Hildebrand, 1986)

where

σ = Population Standard Deviation

s = Sample Standard Deviation

μ = Population Mean

$\bar{x}$ = Sample Mean

N = number of data values for a population

n	= number of data values for a sample
x_i	= i^{th} data value

Coefficient of Variation

The coefficient of variation is the standard deviation divided by the mean. The worksheet reports the quotient: it does not convert the value to a percentage. The coefficient of variation is a dimensionless measure of variation. This statistic is not defined for the case of a zero mean and this measure is only useful when dealing with strictly positive data.

Population Coefficient of Variation (V)

$$V = \sigma / \mu$$

Sample Coefficient of Variation (V)

$$V = s / \bar{x}$$

where

σ	= Population Standard Deviation
s	= Sample Standard Deviation
μ	= Population Mean
$\bar{x}$	= Sample Mean

Critical Value of K-S Statistic at 90%, 95%, and 99% Significance Level

The critical value of K-S statistic at 90%, 95%, or 99% significance level are indicators of normal distributions. For example, if a sample collected from a population has a normal frequency distribution, the K-S statistic for that sample is less than the critical value 90, 95, or 99 percent of the time. If the K-S statistic is larger than the critical value, the hypothesis that the underlying population is distributed normally with a mean of $\bar{x}$ and a standard deviation of s should be rejected.

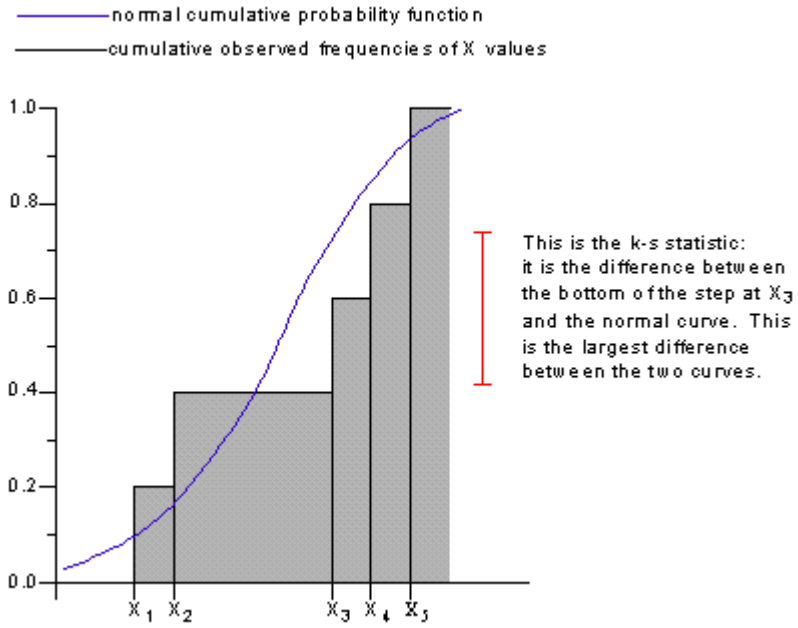
Kolmogorov-Smirnov Goodness of Fit for Normal Distribution

The Kolmogorov-Smirnov statistic is the largest difference between an expected cumulative probability distribution and an observed frequency distribution. The expected distribution used here is the normal probability distribution with mean and variance equal to the mean and variance of the sample data. The observed frequency distribution is a stepped function that increases by 1/n with each step, where n is the number of values in the data set.

For example, suppose that there are five values in a data set. The observed frequency distribution is 0 to the left of the first data point. At the first data point the observed distribution function jumps to 0.2 (since there are five data values, the size of the step at each value is one divided by five). At each successive data value the observed frequency distribution jumps by 0.2.

The K-S statistic is calculated as the largest difference (in absolute value) between the normal cumulative probability function and the observed frequency distribution, as shown below. Note that

at each step it's necessary to compute the difference between bottom of the step and the normal curve and also between the top of the step and the normal curve.



Mean

The mean is the arithmetic average of the data values. It is the sum of the data values divided by the number of data values.

Population Mean (μ)

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$$

Sample Mean ($\bar{x}$)

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

where

N	= number of data values for a population
n	= number of data values for a sample
x_i	= i^{th} data value

Standard Deviation

The standard deviation is the square root of the variance.

Population Standard Deviation (σ)

$$\sigma = \sqrt{\sigma^2}$$

Sample Standard Deviation (*S*)

$$s = \sqrt{s^2}$$

where

σ^2 = Population Variance

s^2 = Sample Variance

Standard Error of the Mean

The standard error of the mean is an estimate of the standard deviation of means that would be found if many samples of *n* items were repeatedly collected from the same population.

An alternate description: Suppose many samples of size *n* were repeatedly collected from the same population and the means of these many samples were calculated. The means of the samples would themselves form a data set. The standard error of the mean is an estimate of the standard deviation of this theoretical sample of means.

Standard Error of the Mean (SE)

$$SE = s / \sqrt{n}$$

where

S = Sample Standard Deviation

n = number of data values for a sample

Variance

The population variance is the average of the squared deviations from the mean. The sample variance is the sum of the squared deviations from the mean divided by one less than the number of data values.

Population Variance (σ^2)

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2$$

Sample Variance (S^2)

$$s^2 = \frac{1}{(n-1)}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

where

μ	= Population Mean
$\bar{x}$	= Sample Mean
N	= number of data values for a population
n	= number of data values for a sample
x_i	= i^{th} data value

Statistics References

Clark, P. and F. Evans. (1954) Distance to nearest neighbor as a measure of spatial relationships in populations. Ecology, 35:445–453.

Cressie, N. A. C. (1991) *Statistics for Spatial Data.* New York: John Wiley & Sons, Inc.

Gilbert, Richard O. (1987) *Statistical Methods For Environmental Pollution Monitoring*. New York: Van Nostrand Reinhold.

Hildebrand, David K. (1986) Statistical Thinking for Behavioral Scientists. Duxbury Press.

King, Ronald S. and Bryant Julstrom. (1982) *Applied Statistics Using the Computer*. Sherman Oaks, California: Alfred Publishing Company.

Press, William H. et al. (1992) *Numerical Recipes in C: The Art of Scientific Computing*. New York: Cambridge University Press.

Montgomery, D. C. and G. C. Runger. (1994) *Applied Statistics and Probability for Engineers.* New York: John Wiley & Sons, Inc.

Ryan, T. (1997) *Modern Regression Methods*. New York: John Wiley & Sons, Inc.

Sheskin, D.J. (2000) Handbook of Parametric and Nonparametric Statistical Procedures, Second Edition. Boca Raton, Florida: Chapman & Hall/CRC.

Skellam, J. (1952) Studies in Statistical Ecology. I: Spatial Pattern. *Biometrika*, 39:346–362.

Sokal, Robert R. and F. James Rohlf. (1981) *Biometry: Principles and Practices of Statistics in Biological Research*. New York: Freeman and Co.

Zar, Jerrold H. (1974) *Biostatistical Analysis*. Englewood Cliffs, New Jersey: Prentice Hall.

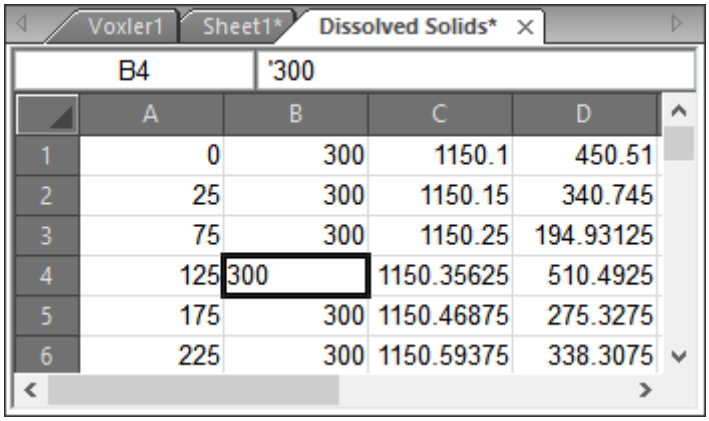
Text To Number

Click the **Data | Text to Number** command to convert text strings in selected cells to numbers. This command will eliminate extraneous zeros and convert text to simplified numeric format.

To determine if the cell contains a number formatted as text, click on the cell to select it. The cell edit box displays the number with an apostrophe (') before the number. For instance, in the image below, the number 300 appears as '300, if the number if formatted as text.

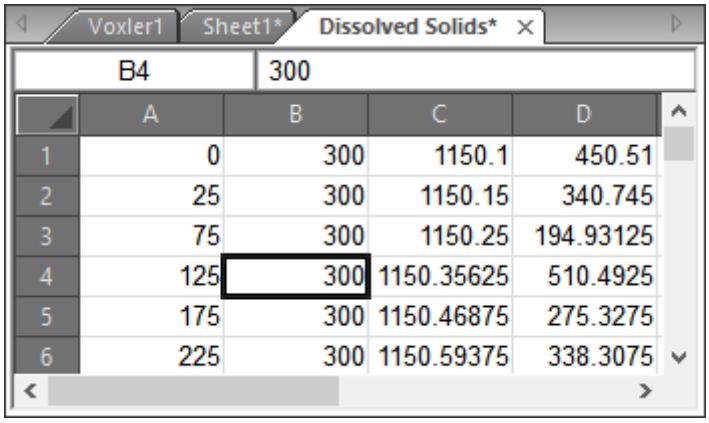
To convert a cell:

1. Click on the cell to select it.



Click on the cell to select it. Notice the cell edit box contains an ' before the number.

1. Click the **Data | Text to Number** command.
2. The cell converts to numeric format, if the cell had been a text number. The cell edit box now displays the number without the ' before it.



The cell now shows the number right aligned and the cell edit box does not contain an ' before the number.

Transpose

The **Data | Transpose** command rearranges data from columns to rows or from rows to columns. To quickly switch the layout of your data, highlight the data that should be flipped. Click the **Data | Transpose** command and the columns become rows and the rows become columns.

For example, consider the following data:

	A	B	C	D	E
1	Category	Spring	Summer	Fall	Winter
2	A	12	14	15	21
3	B	13	5	23	12
4	C	51	21	12	32

Categories A, B, and C are displayed with each category in a row.

Highlight the rows 1-4. Click the **Data | Transpose** command and the data appears in columns:

	A	B	C	D
1	Category	A	B	C
2	Spring	12	13	51
3	Summer	14	5	21
4	Fall	15	23	12
5	Winter	21	12	32

Categories A, B, and C are now displayed with each category in a separate column.

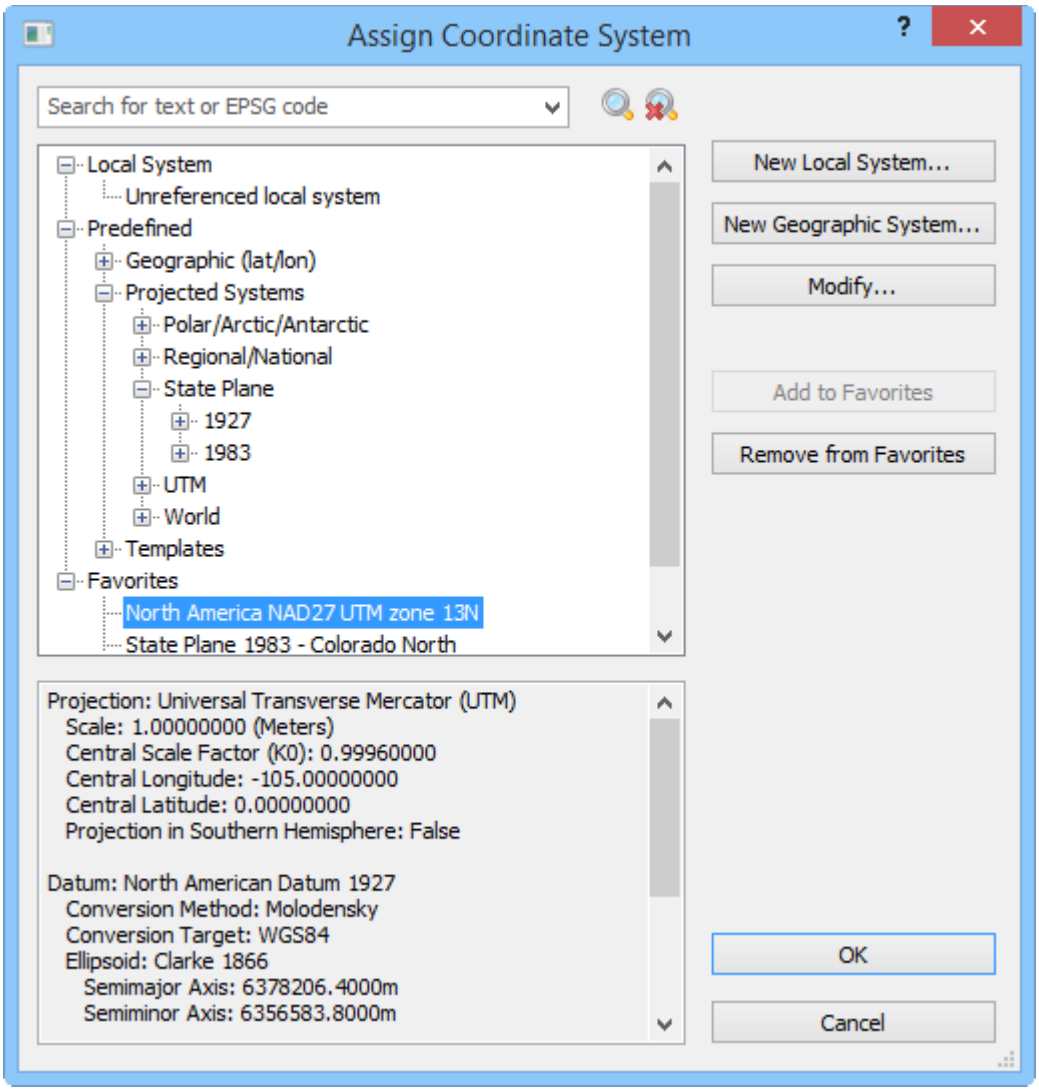
Assign Coordinate System

The **Data | Assign Coordinate System** command links a data file to a specific coordinate system. Click the **Data | Assign Coordinate System** button to specify the coordinate system for the data file in the **Assign Coordinate System** dialog. Once the coordinate system is defined for the data file, a Golden Software Georeference .GSR2 file is created. This file contains all the relevant projection information that **Voxler** needs to load the data in the proper projection.

Assign Coordinate System Dialog

The **Assign Coordinate System** dialog is accessed from several locations. It links a file to a specific coordinate system. Once a coordinate system is defined for a file, a Golden Software Georeference .GSR2 file is created. This file contains all the relevant coordinate system information that **Voxler** needs to load the file in the proper coordinate system in the future.

When a .GSR2 file is created for a data file, it is read when creating grids or maps from the data file. The resulting grid or map layer has the same coordinate system as the original data file. It is recommended to check the *GS Reference (Version 2) file* if you intend to use the grid file in **Surfer**, as the GSR2 retains all of the information needed.



*Assign a projection to your file or map layer in the **Assign Coordinate System** dialog.*

Search for Coordinate Systems

Voxler has over 2500 coordinate systems in the **Assign Coordinate System** dialog. To search for a specific coordinate system type a partial name, complete name, or EPSG code into the *Search for text or EPSG code* field. Next click the  button or press ENTER to search for the coordinate system. The number of returned search results will be displayed below the search bar. The search results will replace the full coordinate system list. Navigate through the search results by clicking the ⊞ button to expand the categories in the **Assign Coordinate System** dialog. Click the  button to clear the search results and display all of the coordinate systems in the **Assign Coordinate System** dialog.

When searching in the **Assign Coordinate System** dialog, the search string must exactly match a portion of the desired coordinate system name or EPSG code. However, the search string does not need to be the complete name or EPSG code. For example, searching for *System 1984* will return the *World Geodetic System 1984* coordinate system, but searching for *World 1984* returns no results.

Projection Categories

Click the ⊞ button to expand the options in the **Assign Coordinate System** dialog. Click the ⊟ button to collapse the options.

Local System

Expand *Local System* to select the *Unreferenced local system*, which contains a *Projection* of None, a *Datum* of None, and a *Warp* of None. For example, you may have a data set with an arbitrary coordinate system (i.e. not real world X, Y coordinates). You would assign this data to a *Local System*, if a coordinate system is necessary.

Predefined

Expand the *Predefined* section by clicking the + button. The *Predefined* section includes all coordinate systems that have been predefined for **Voxler**.

Geographic (lat/lon)

Expand *Geographic (lat/lon)* to select a Latitude/Longitude coordinate system and datum that fits your needs. Detailed information about each projection is listed at the bottom of the dialog when the system is selected. If your data are currently in a form of lat/lon, you would want to select one of the options in *Geographic (lat/lon)*.

Projected Systems

Expand *Projected Systems* to select a predefined *Polar/Arctic/Antarctic*, *Regional/National*, *State Plane*, *UTM*, or *World* coordinate system. Detailed information about each projection is listed to the right when the datum is selected.

Templates

Expand the *Templates* section to select a predefined template. Click the desired template and press the *Modify* button to change the properties. Once modified, the new coordinate system is added to the *Custom* section.

Favorites

Select a coordinate system and click the *Add to Favorites* button to add a coordinate system to your *Favorites* list. Alternatively, right-click on a selected coordinate system and choose **Add to Favorites** from the menu. Select a coordinate system and click the *Remove from Favorites* button to remove a coordinate system from your *Favorites* list. By default, no favorite coordinate systems are specified. It is recommended that systems that you use frequently and *Custom* systems that you intend to use more than once be added to the *Favorites* section for ease of navigation.

Custom

Expand *Custom* to see the custom coordinate system you have defined for the current file. *Custom* systems can be defined by clicking the *New* button. By default, no custom coordinate systems are specified. Custom coordinate systems are only listed in the dialog when the file using the custom coordinate system is open.

If a *Custom* coordinate system is defined and intended to be used more than once, it is highly recommended that the system be added to the *Favorites* section by clicking the *Add to Favorites* button.

New Custom Local Coordinate System

The linear units and offsets can be specified for local coordinate systems. Click the *New Local System* button to open the **Define Unreferenced Coordinate System** dialog and define a custom local unreferenced coordinate system. Alternatively, click the *Modify* button while *Unreferenced local system* is selected in the *Select coordinate system* list to open the **Define Unreferenced Coordinate System** dialog.

New Custom Geographic Coordinate System

Click the *New Geographic System* button to open the **Define Coordinate System** dialog and define a custom projection and datum. Alternatively, right-click on an existing coordinate system and choose *New Geographic System* to create a new projection based off the selected projection. The new custom projection and datum will be listed in the *Custom* section with the name you defined.

Modify Coordinate System

Select any coordinate system from the *Select a coordinate system* list. Click the *Modify* button to open the **Define Coordinate System** or **Define Unreferenced Coordinate System** dialog. Alternatively, right-click on the coordinate system and choose *Modify* from the menu. Modify any properties and click *OK*. The modified coordinate system is added to the *Custom* list.

If the *Modify* button is unavailable, make sure that a specific coordinate system is selected. If a category of systems, such as *Predefined* or *Geographic (lat/lon)* is selected, the *Modify* button is unavailable.

Add to Favorites

Click the *Add to Favorites* button to add a projection to your *Favorites* list. Alternatively, right-click on a coordinate system and choose *Add to Favorites.* This is very useful if you frequently use the same projection, such as World Geodetic System 1984. This is also useful for *Custom* systems that will be used on multiple project files. Adding the projection to the *Favorites* list makes selecting the projection easier in the future.

If the *Add to Favorites* button is unavailable, make sure that a specific coordinate system is selected. If a category of systems, such as *Predefined* or *Geographic (lat/lon)* is selected, the *Add to Favorites* button is unavailable.

Remove

Select a coordinate system in the *Favorites* list and click the *Remove from Favorites* button to delete the system. Alternatively, right-click on a coordinate system in the *Favorites* section and choose *Remove.* The coordinate system is removed from the *Favorites* section.

If the *Remove from Favorites* button is unavailable, make sure that a specific coordinate system is selected in the *Favorites* list. If a category of systems, such as *Predefined* or *Geographic (lat/lon)* is selected or if a coordinate system is selected in a category other than *Favorites*, the *Remove from Favorites* button is unavailable.

Example 1: Select a Predefined Coordinate System (i.e. UTM)

1. In the **Assign Coordinate System** dialog, click the ⊞ button to the left of *Predefined*.
2. Click the ⊞ button to the left of *Projected Systems.*
3. Click the ⊞ button to the left of *UTM*.

4. Click the ⊞ button to the left of *WGS84*.
5. Select WGS84 UTM zone 13N.
6. Click *OK*. The projection for this data is now set to WGS84 UTM zone 13N.

Example 2: Create and Select a Custom Coordinate System (i.e. Lambert Conformal Conic)

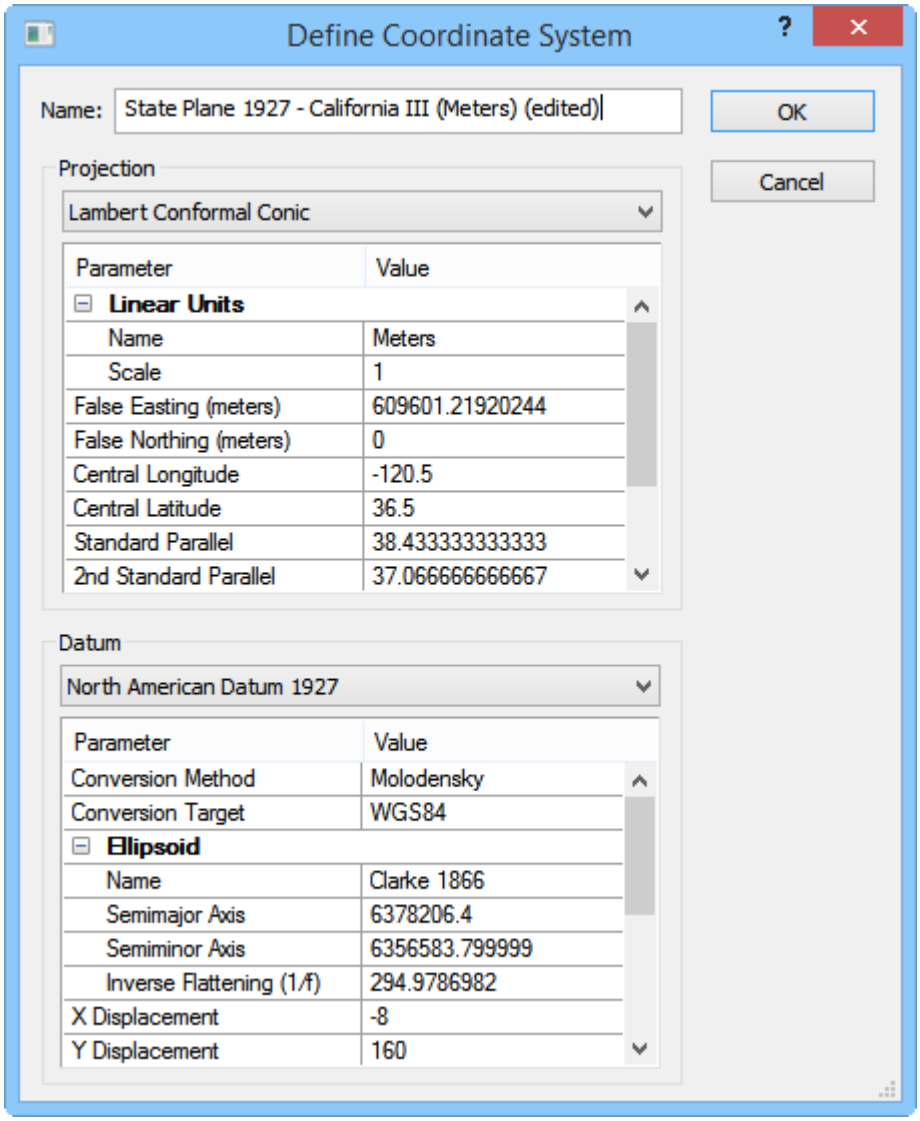
1. In the **Assign Coordinate System** dialog, click the *New* button to open the **Define Coordinate System** dialog.
2. Change the Name to Lambert Conformal Conic Custom.
3. From the Projection list, select Lambert Conformal Conic.
4. Make any necessary changes to the *Parameter* or *Values* in the *Projection* section.
5. Make any necessary changes to the *Datum* section.
6. Click *OK* and the *Lambert Conformal Conic Custom* projection is added to the *Custom* section.
7. Click the custom projection and click *OK* to apply the projection to the data.

Example 3: Saving a Custom Coordinate System

1. In the **Assign Coordinate System** dialog, click the ⊞ button to the left of *Custom.*
2. Select the coordinate system in the *Custom* section and click the *Add to Favorites* button.
3. Click the ⊞ button to the left of *Favorites* to open the *Favorites* section. The custom projection is saved here.
4. Click *OK* to close the dialog.

Define Coordinate System

Click the *New Geographic System* or *Modify* buttons in the **Assign Coordinate System** dialog to open the **Define Coordinate System** dialog. The **Define Coordinate System** dialog allows you to create a *Custom* coordinate system. Select a *Projection*, enter the projection *Parameters*, and specify a *Datum*. Click *OK* and the new coordinate system will be added to the **Assign Coordinate System** dialog *Custom* list for future use.



*Define a custom coordinate system in the **Define Coordinate System** dialog.*

Name

Specify a custom name for the new coordinate system in the *Name* box. When editing a predefined coordinate system, the name is automatically appended with *(edited)* after the existing name.

Projection

Select a projection from the *Projection* list. Specify custom *Parameter* and *Value* options. *Unprojected Lat/Long* does not have *Parameter* and *Value* options to specify.

The projection options include: Unprojected Lat/Long, Albers Equal Area Conic, Azimuthal Equidistant, Bonne, Cassini, Eckert IV, Eckert VI, Equidistant Conic, Equidistant Cylindrical, Gnomonic, Hotine Oblique Mercator, Lambert Azimuthal Equal Area, Lambert Conformal Conic, Mercator, Miller Cylindrical, Mollweide, Oblique Mercator, Orthographic, Polyconic, Robinson, Robinson-Sterling, Sinusoidal, Stereographic, Transverse Mercator, Universal Transverse Mercator (UTM), and Van Der Grinten.

Datum

Select a datum from the *Datum* list. Specify custom *Parameter* and *Value* options to fit your needs.

OK or Cancel

Click *OK* to create your new custom coordinate system. The coordinate system will be listed in the *Custom* section of the **Assign Coordinate System** dialog for this file. If you want to save the custom coordinate system for future use, add the custom coordinate system to the *Favorites* section. Click *Cancel* to return to the Assign Coordinate System dialog without creating a custom coordinate system.

Example 1: Defining a Custom Coordinate System

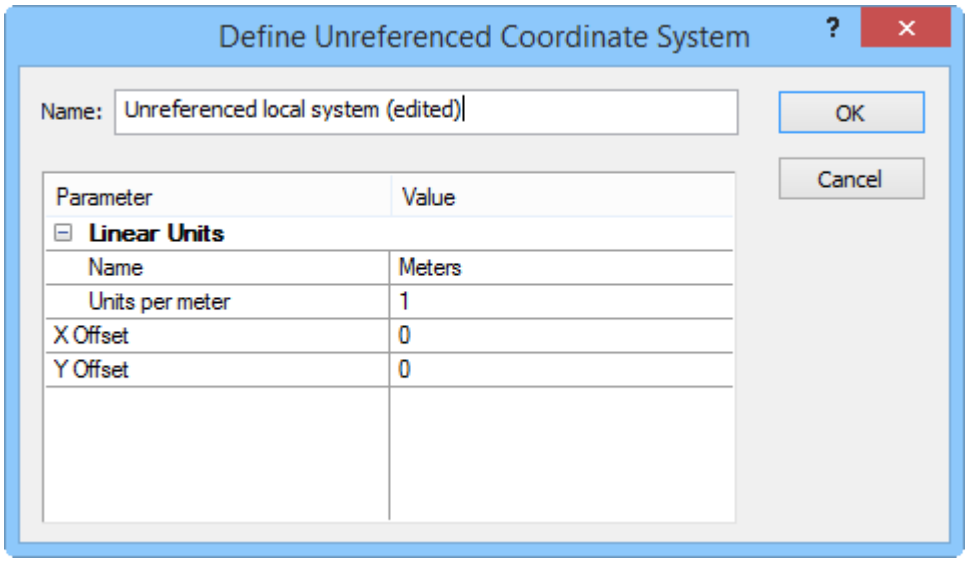
1. In the **Assign Coordinate System** dialog, click the *New Geographic System* button to open the **Define Coordinate System** dialog.
2. Change the Name to Lambert Conformal Conic Custom.
3. From the Projection list, select Lambert Conformal Conic.
4. Make any necessary changes to the *Parameter* or *Values*.
5. Make any necessary changes to the *Datum* section.
6. Click *OK* and the *Lambert Conformal Conic Custom* projection is added to the *Custom* section.
7. In the **Assign Coordinate System** dialog, select the custom projection and click *OK*.

Example 2: Saving a Custom Coordinate System

1. In the **Assign Coordinate System** dialog, click the ⊞ button to the left of *Custom*.
2. Select the coordinate system in the *Custom* section and click the *Add to Favorites* button.
3. Click the ⊞ button to the left of *Favorites* to open the *Favorites* section. The custom projection is saved here.
4. Click *OK* to close the dialog.

Define Unreferenced Coordinate System

Click the *New Local System* button, or *Modify* button with a local coordinate system selected, in the **Assign Coordinate System** dialog to open the **Define Unreferenced Coordinate System** dialog. The **Define Unreferenced Coordinate System** dialog creates a *Custom* coordinate system. Specify the linear units and offset for the unreferenced coordinate system. Click *OK* and the new coordinate system will be added to the **Assign Coordinate System** dialog *Custom* list for future use.



Define a custom local coordinate system in the ***Define Unreferenced Coordinate System*** *dialog.*

Name

Specify a custom name for the new coordinate system in the *Name* box. When editing a predefined coordinate system, the name is automatically appended with *(edited)* after the existing name.

Linear Units

Specify the units for the new coordinate system in the *Name* field. Click the current selection and select the desired units from the list.

Alternatively, specify custom units by typing the number of units per meter in the *Units per meter* box. For example, to use the uncommon unit UK Metric Foot (defined as 300mm), type *3.33333333* into the *Units per meter* field.

Offset

Apply an offset by typing a value in the *X Offset* or *Y Offset* fields.

OK or Cancel

Click *OK* to create your new custom coordinate system. The coordinate system will be listed in the *Custom* section of the **Assign Coordinate System** dialog for this file. If you want to save the custom coordinate system for future use, add the custom coordinate system to the *Favorites* section. Click *Cancel* to return to the **Assign Coordinate System** dialog without creating a custom coordinate system.

Example 1: Defining Units for an Unreferenced Coordinate System

1. In the **Assign Coordinate System** dialog, click the *New Local System* button to open the **Define Unreferenced Coordinate System** dialog.
2. Change the Name to Local Meters (no offset).
3. From the *Name* list in the *Linear Units* section, select *Meters*.
4. Click *OK* and the *Local Meters (no offset)* coordinate system is added to the *Custom* section.
5. In the **Assign Coordinate System** dialog, select the custom projection and click *OK*.

Example 2: Saving a Custom Coordinate System

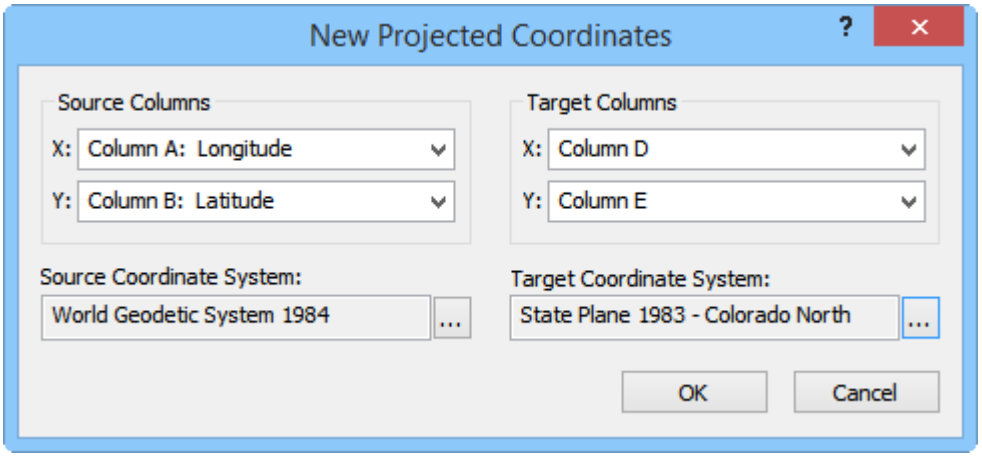
1. In the **Assign Coordinate System** dialog, click the ⊞ button to the left of *Custom.*
2. Select the coordinate system in the *Custom* section and click the *Add to Favorites* button.
3. Click the ⊞ button to the left of *Favorites* to open the *Favorites* section. The custom projection is saved here.
4. Click *OK* to close the dialog.

New Projected Coordinates

The **Data | New Project Coordinates** command allows you to specify a *Source Coordinate System* and project the X and Y coordinate data to a new *Target Coordinate System*. The input *Source Columns* and output *Target Columns* allow you to either overwrite the original coordinate columns, or write the new coordinates to new columns in you worksheet. For example, this command can be useful if you have coordinate data in latitude and longitude (i.e. *Source Coordinate System*) and need to project the coordinates to UTM (i.e. *Target Coordinate System*).

The New Projected Coordinates Dialog

The **Data | New Projected Coordinates** command in the worksheet opens the New Projected Coordinates dialog.



*Change coordinates in the **New Projected Coordinates** dialog.*

Source Columns

Specify the columns containing the X and Y coordinates in the *Source Columns X* and *Y*. Click the arrow to see a list of the data columns in your worksheet. The *Source Columns* contain the X and Y values in the existing coordinate system.

Target Columns

Specify the X and Y destination columns in the *Target Columns X* and *Y*. The projected output coordinates will be located in the target columns specified. Click the arrow to see a list of the data columns in your worksheet.

Source Coordinate System

Assign the *Source Coordinate System* by clicking the  button to open the **Assign Coordinate System** dialog. The *Source Coordinate System* is the system that your X, Y coordinate data are currently using.

Target Coordinate System

Assign the *Target Coordinate System* by clicking the  button to open the **Assign Coordinate System** dialog. The *Target Coordinate System* is the new coordinate system that you want to use to project your X, Y coordinate data.

Projections

What is a Coordinate System?

A coordinate system is method of defining how a file's point locations display on a map. Different types of coordinate systems exist that control how the coordinates are shown on the map. In **Voxler**, a data file can be unreferenced in local coordinates, referenced to a geographic lat/long coordinate system, or referenced to a known projection and datum.

A local coordinate system generally is considered unreferenced. A local system has a location that begins numbering at an arbitrary location and increments numbers from this location. This is frequently referred to as a *Cartesian coordinate system*.

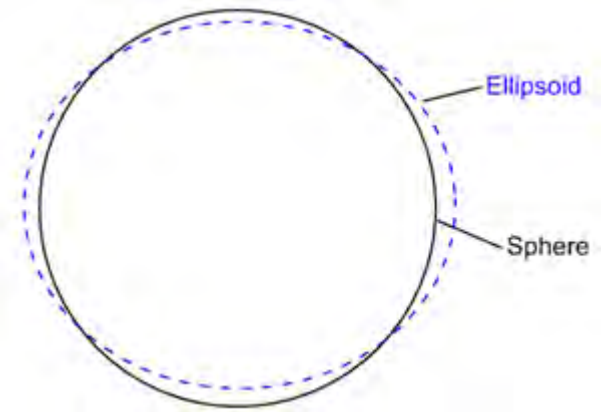
A *Geographic coordinate system* uses a spherical surface to define locations on the earth. Geographic coordinate systems are commonly called unprojected lat/long. **Voxler** has several predefined geographic coordinate systems available. Each system has a different datum. The same latitude and longitude value will plot in different locations depending on the datum.

A *Projected* coordinate system consists of a projection and a datum. Each projection distorts some portion of the map, based on the ellipsoid and datum specified. Coordinates can be lat/long, meters, feet, or other units. Different projections cause different types of distortion. It is recommended that you do not use projected coordinate systems if you do not need to convert between coordinate systems or if all your data are in the same coordinate system.

Ellipsoids

For maps of the Earth where accuracy is not of particular concern, we can safely assume that the Earth is perfectly spherical in shape. For small-scale maps, the difference between a sphere and ellipsoid is not detectable on the map.

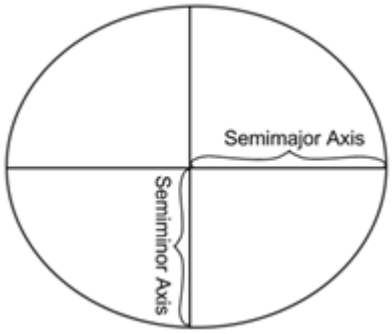
However, the Earth is actually somewhat ellipsoidal (or egg-shaped), approximately 1/300th wider than it is tall, assuming a vertical orientation with respect to the axis of rotation. This shape needs to be taken into account to produce larger scale maps of any significant accuracy.



This is an example of an ellipsoid (dashed blue line) superimposed on a sphere (solid black line).

While an ellipsoid is a closer approximation of the Earth's shape than a sphere, the Earth's surface is not entirely uniform in curvature, so any ellipsoidal representation of the Earth is still only an approximation. This being the case, cartographers have historically used a number of slightly different ellipsoidal representations in attempts to produce more accurate maps of different regions of the Earth.

Ellipsoids are defined by the ellipse being used and by the amount of rotation of the ellipse. An ellipse has two axes. The longer axis is the *Semimajor Axis*. The shorter axis is the *Semiminor Axis*. Rotating the ellipse around the semiminor axis creates the ellipsoid.



The diagram shows the semimajor and semiminor axes on an ellipse.

In **Voxler**, the ellipsoid can be defined by the *Semimajor Axis* and the *Semiminor Axis* or by the *Semimajor Axis* and the *Inverse Flattening (1/f)* value. The flattening value ranges from zero to 1, so the *Inverse Flattening (1/f)* value must be larger than 1. The flattening value is determined by:

f = (Semimajor Axis - Semiminor Axis)/Semimajor Axis

In the **Assign Coordinate System** dialog, click the *New* button to define a new coordinate system. In the **Define Coordinate System** dialog, the *Datum* group allows you to specify parameters for the datum. The *Ellipsoid* can be customized in terms of *Name*, *Semimajor Axis*, *Semiminor Axis*, and *Inverse Flattening (1/f)*. The *Semimajor Axis*, *Semiminor Axis*, and *Inverse Flattening* ratio should be set in meters. If you do not understand ellipsoids and datum definitions, it is

recommended you use the defaults. For more information on these subjects, see the projection references.

Datums

Since coordinates on the Earth's surface can be recorded under widely varying assumptions about the shape and size of the Earth and the locations of the poles and prime meridian, cartographers have developed a standard for identifying the frame of reference for a coordinate system. This standard is called the *datum*. Because the frames of reference differ, a coordinate recorded in one datum usually has slightly different latitude and longitude values from the same point recorded in any other datum.

Some datums are designed to provide a marginally accurate representation of coordinates spanning the entire Earth, while other datums are designed to provide more accurate results in a particular region at the expense of lesser accuracy in other parts of the world. For example, the *South American Datum of 1969 (SA69)* is tailored to provide good results for maps of the South American continent and surrounding areas, but, as a consequence, provides poor results for the rest of the world.

When combining data from multiple sources into a single map, it is important that all of the coordinate systems being combined specify the projection and datum accurately. Since each datum has slightly different latitude and longitude values for the same coordinates, mixing coordinates from multiple datums together without fully defining the datum introduces inaccuracies into the map.

A *datum conversion* can be used to convert coordinates from one datum to another using the **Data | New Projected Coordinates** command. Click the **New Projected Coordinates** command to open the **New Projected Coordinates** dialog. Next click the  to specify the source and target coordinate systems in the **Assign Coordinate System** dialog.

If a custom datum definition is required, click *New* in the **Assign Coordinate System** dialog to define a new projection and datum. The **Define Coordinate System** dialog has the *Conversion Method* and *Ellipsoid* parameters necessary to allow you to define a coordinate system with a custom datum.

Several different *Conversion Methods* may be used for converting coordinates from one datum to another:

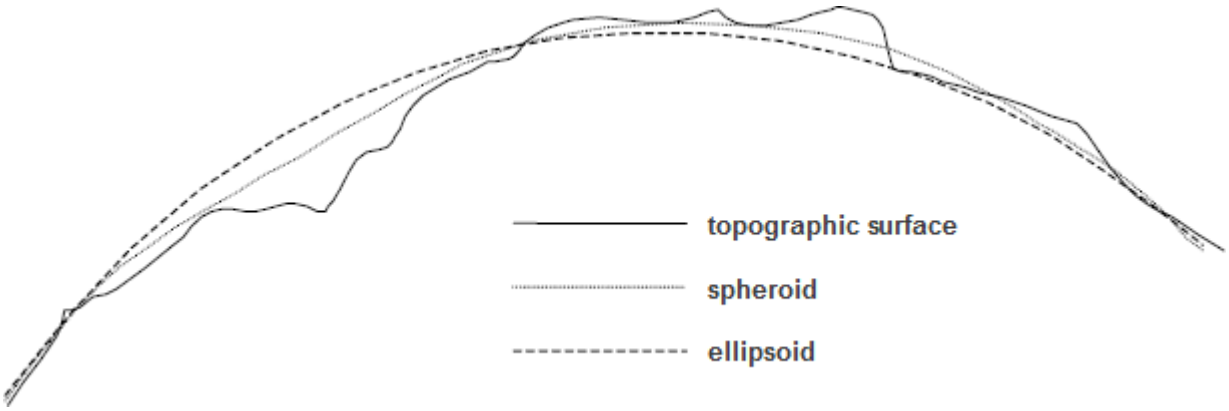
Molodensky	The *Molodensky* method is the most widely used method of datum conversion. It adjusts latitude and longitude coordinates by taking into account the displacement between two datum's ellipsoids on all three axes. It does not take into account any rotational differences between the two ellipsoids.
Bursa-Wolfe	The *Bursa-Wolfe* method is similar to the *Molodensky* method, but in some instances it produces more accurate results because it takes into account both displacement and rotational differences between two ellipsoids. **Surfer** supports the *Bursa-Wolfe* method for conversions from the WGS84 datum to the following datums: World Geodetic System 1972, DHDN-1, DHDN, Australian Geodetic 1984, ANS84, MRT - Everest Modified, Switzerland - CH1903, NTF France - Paris Meridian, and Pulkovo 1942 - Hungary.

Voxler supports the NTv2 datum conversion method. NTv2 is the Canadian government's officially sanctioned method of converting Canadian map data from the old NAD27 datum to the NAD83 datum. NTv2 is based on a hierarchical database of interpolation grids of different resolutions for different regions of the country. NTv2 datum conversions cannot be performed unless an NTv2 grid shift file is installed in the same folder/directory as the **Voxler** program. Visit Natural Resources Canada's Geodetic Reference Systems page on the web for more information about this datum and obtaining an NTv2 grid shift file. See the **Golden Software** How to convert from NAD27 to NAD83 using NTv2 help topic for detailed steps on using the NTv2 datum conversion.

Voxler supports conversions for over 200 different predefined datums.

Understanding Local Datums

To understand how local datum transforms affect data, you need to understand the ellipsoid, the spheroid, and how both relate to the datum. The spheroid is the a perfect circular object, located at mean sea level. This is constant everywhere. Because of local variations in the gravitational field, the shape actually has local variations.



The image from Verhoogan, shows the variation in the geoid, ellipsoid, and actual topographic surface.

The ellipsoid is used to approximate the global differences. But additional differences exist because of the actual topographic surface. Local datums were created to locally account for these differences. The datum includes the ellipsoid, the prime meridian shift, and any offsets in the X or Y direction. Because local datums align the ellipsoid with a particular location on the earth's surface, local datums are not suitable for use outside the designed area.

In the **Define Coordinate System** dialog, you can choose how datums are converted from one datum to another. This is the *Conversion Method*. The *Molodensky* method is the most widely used method of datum conversion. It adjusts latitude and longitude coordinates by taking into account the displacement between two datum's ellipsoids on all three axes. It does not take into account any rotational differences between the two ellipsoids. The *Bursa-Wolfe* method is similar to the *Molodensky* method, but in some instances it produces more accurate results because it takes into account both displacement and rotational differences between two ellipsoids. **Voxler** supports the *Bursa-Wolfe* method for conversions from the WGS84 datum to the following datums: World Geodetic System 1972, DHDN-1, DHDN, Australian Geodetic 1984, ANS84, MRT - Everest Modified, Switzerland - CH1903, NTF France - Paris Meridian, and Pulkovo 1942 - Hungary.

Changing the datum incorrectly can cause maps to appear wrong or not appear at all. It is adviced that chaning the datum be done with caution and a basic understanding of the local datums is advised.

Custom Datum Definition

Click the *New Geographic System* button in the **Assign Coordinate System** dialog to open the **Define Coordinate System** dialog. Use the *Datum* group to define a custom datum and specify the exact datum parameters and values.

Datum

The *Datum* list contains predefined datums. Select a datum to populate the *Parameter* and *Value* columns. Customize the parameters and values as needed.

Conversion Method

The *Conversion Method* controls the method of datum conversion and the conversion parameters. The conversion methods include *Molodensky*, *Bursa-Wolfe*, and *None*.

- The *Molodensky* method is the most widely used method of datum conversion. It adjusts latitude and longitude coordinates by taking into account the displacement between two datum's ellipsoids on all three axes. It does not take into account any rotational differences between the two ellipsoids.
- The *Bursa-Wolfe* method is similar to the *Molodensky* method, but in some instances it produces more accurate results because it takes into account both displacement and rotational differences between two ellipsoids.
- Choose *None* if the predefined methods do not suite your purpose. If you select <*custom*> from the *Ellipsoid Name* list, you can specify your own ellipsoid model parameters.

Conversion Parameter and Values

You can edit the conversion parameter and values directly by selecting the value you want to edit and typing a new value over it. Parameters vary depending on the selected *Conversion Method*.

Ellipsoid

The *Ellipsoid* group contains options for defining the ellipsoid. Use these settings to define the ellipsoid model that best approximates the curvature of the Earth's shape in the map region. If you do not understand ellipsoids and datum definitions, it is recommended you use the defaults.

- The *Name* list contains a collection of ellipsoid models that cartographers have historically used in attempts to produce more accurate maps of different regions of the Earth. The *Name* list also contains the option to create a <*custom*> ellipsoid.
- The *Semimajor Axis* box can be edited to define the major axis.
- The *Semiminor Axis* box can be edited to define the minor axis.
- The *Inverse Flattening (1/f)* can be edited to define the flattening ratio.
- The *X Displacement*, *Y Displacement*, and *Z Displacement* are the axis displacements in meters.

Prime Meridian

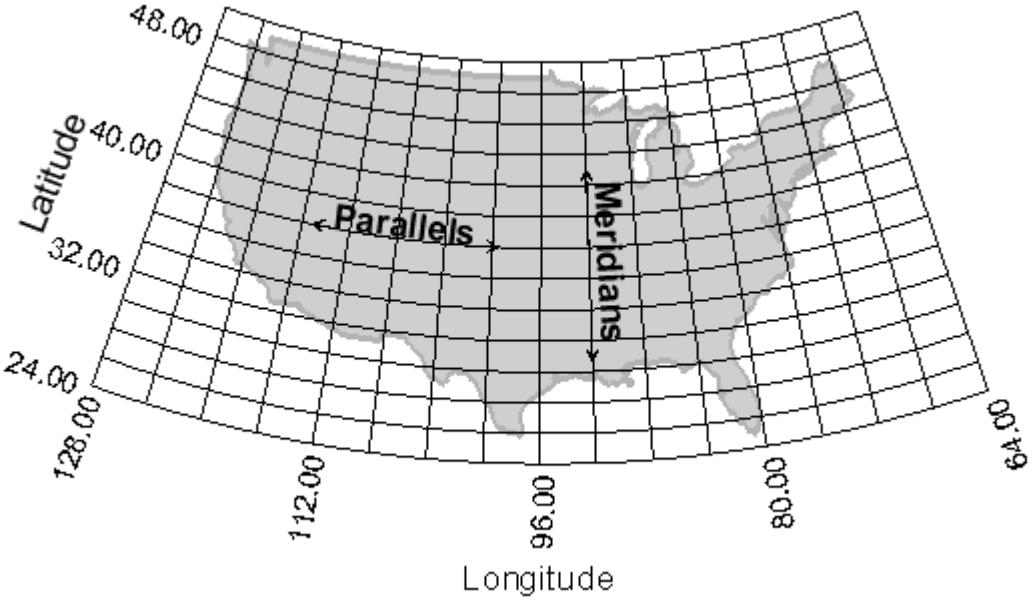
The *Prime Meridian* section contains options for defining the prime meridian *Name* and *Prime Meridian Shift*. A meridian is a line of constant longitude running north-south on a map. The zero meridian or prime meridian is used as a reference line from which longitude east and west is measured. The prime meridian passes through Greenwich, England. The *Prime Meridian Shift* is the shift from the prime meridian, typically 0 degrees, in decimal degrees.

Latitude and Longitude Coordinates

In **Voxler**, worksheet data can be projected using a projection. The projected data can then be used to create a *Data Source* module.

Latitude and longitude are spherical coordinates used to locate a point on the earth. Many maps do not need to take the curvature of the earth into account. For maps covering relatively small land areas, such as a state or small group of states, the earth can be assumed to be flat. In these cases, the latitude/longitude coordinates can be plotted on a Cartesian coordinate system. Maps plotted in this way must use different scaling in the two dimensions to minimize distortion on the map.

Voxler only plots latitude and longitude coordinates in decimal degrees. You can see *Latitude and Longitude Coordinates in Decimal Degrees* for information on converting degrees-minutes-seconds to decimal degrees.



Parallels define lines of constant latitude. Meridians define lines of constant longitude. This Albers projected map makes use of graticule lines to indicate the relationship.

Latitude

Latitude is the Y coordinate and defines north-south global position measured from the equator. Lines of constant latitude are called parallels because they define a series of rings parallel to the equator. Parallels run east-west, but define north-south position on the globe. Parallels are designated in degrees from 0° at the Equator to 90° at the poles. **Voxler** uses the convention that parallels are positive north of the equator (north latitudes), and negative south of the equator (south latitudes). Designations such as 45° indicate a position 45 degrees north of the equator, while -65° indicates a position 65 degrees south of the equator. At any position on the globe, the distance covered by a degree of latitude remains nearly constant.

Longitude

Longitude is the X coordinate and indicates east-west position on the globe. Lines of constant longitude are called meridians. Meridians lie at right angles to the parallels and are half-circles drawn from the North Pole to the South Pole. One meridian is designated as the prime meridian.

The prime meridian most commonly in use in the United States runs through Greenwich, England, although there are several other prime meridians in use throughout the world. Longitude is measured 180° east and 180 degrees west from the prime meridian. In **Voxler**, longitude is positive east (east longitude) of the prime meridian, and negative west of the prime meridian (west longitude). A designation such as -105° is used to indicate a location 105 degrees west of the prime meridian. Meridians converge at the poles so the distance covered by one degree of longitude decreases as you move north or south from the equator.

Latitude and Longitude in Decimal Degrees

Latitude and Longitude coordinates are often presented in degrees, minutes, and second, such as 39°25'30" (39 degrees, 25 minutes, 30 seconds). However, **Voxler** can only plot values in decimal degrees. So, for example, 39°25'30" is referred to as 39.425 in **Voxler**.

Converting from degrees, minutes, and seconds is actually quite easy. There are 60 minutes in one degree and 3600 seconds in one degree. To convert minutes and seconds to decimal degrees, divide minutes by 60, divide seconds by 3600, and then add the results to obtain the decimal equivalent.

Conversion Equation:

Decimal Degrees = Degrees + (Minutes / 60) + (Seconds / 3600)

Example

Consider the latitude value 39°25'30". This value needs to be converted to decimal degree in order to use it in **Voxler**.

To convert 39°25'30" to decimal degrees:

1. First, convert minutes (25') and seconds (30") to their degree equivalents and add the results.

25'/60 = 0.4167

30"/3600 = 0.0083

0.4167 + 0.0083 = 0.425

2. Then, add this number to the number of degrees.

39 + 0.425 = 39.425

3. The final result is the decimal degree value.

39°25'30" = 39.425°

Types of Projections

Most forms of projection operate by projecting Earth coordinates onto a geometric shape that can be easily flattened to a two-dimensional image. This mathematical transformation is commonly referred to as a map projection. A map projection systematically projects locations from the surface of the spheroid to represent positions on the geometric shape.

Three geometric shapes are frequently used:

Type of Shape	How it works	Characteristics of Projection	Examples of Projection
Cylinder	Earth coordinates may be projected onto a cylinder. The cylinder is cut lengthwise and unrolled to make a two-dimensional map. This type of projection is called a cylindrical projection.	▪ Lines of longitude are parallel to each other. ▪ Lines of latitude are parallel to each other. ▪ Lines of longitude are at right angles to lines of latitude. ▪ Regions near the equator or selected standard parallels are minimally distorted. ▪ Regions near the poles are highly distorted.	Cassini, Equidistant Cylindrical, Hotine Oblique Mercator, Mercator, Miller Cylindrical, New Zealand Map Grid, Oblique Mercator, Transverse Mercator, and Universal Transverse Mercator
Cone	Earth coordinates may be projected onto a cone. The point of the cone is usually directly above the pole and the sides of the cone pass through the globe at two user-defined latitudes, called the Standard Parallels. At the standard parallels, there is no difference between the east-west and north-south scales. The cone is cut from tip to base and unrolled to make a two-dimensional map. This type of projection is called a conic projection.	• Lines of latitude form concentric arcs. • Lines of longitude are straight and radiate outward from the tip of the imaginary cone.	Albers Equal Area, Equidistant Conic, Lambert Conformal Conic, Polyconic, and Bonne

Plane	Earth coordinates may be projected directly onto a flat plane. This type of projection is called an azimuthal projection. Projections of this type are recommended for maps of polar regions because cylindrical and conic projections generally either have severe distortion in polar regions or are unable to project coordinates in polar regions.	• The side of the Earth that is facing away from the center of the projection is not visible.	Azimuthal Equidistant, Gnomonic, Orthographic, Stereographic, and Lambert Azimuthal Equal Area
Other	Projections in this category are pseudocylindrical, pseudoconic, or based on some other mathematical projection or mathematical tables.		Eckert IV, Eckert VI, Mollweide, Robinson, Robinson-Sterling, Sinusoidal, State Plane*, Unprojected Lat/Long, and Van der Grinten

* The State Plane Coordinate System uses Transverse Mercator, Lambert Conformal Conic, or Hotine Oblique Mercator, depending on the zone.

Characteristics of Projections

Some projections are imbued with characteristics that tell us if certain types of measurements (e.g. measurements of distance, area, etc.) are accurate on the projected map. Some of these characteristics include the following:

Type of Projection	Characteristic of Projection	Drawbacks	Example Projections
Equal Area	A *equal area* projection is when the area of any given part of the map is preserved. This means that the any object that covers the same area on the Earth as any other part of the map will be the same size. For example, if a one inch diameter circle on the map covers a 100 mile diameter circle on the Earth's surface, then we know that a one inch diameter circle anywhere else on the map is known to cover another 100 mile diameter circle on the Earth. In maps of smaller regions, shapes may not be obviously distorted.	In order for a projection to be equal area, however, consistency in the shapes, scales, and/or angles across the map must be sacrificed. Meridians and parallels may not intersect at right angles.	Albers Equal Area, Bonne, Eckert IV, Eckert VI, Lambert Azimuthal Equal Area, Mollweide, and Sinusoidal
Conformal	A *conformal* projection preserves local shapes. This means that when the local angles for points on the map are represented accurately. This means that the angles between any given point and any nearby points are accurate, but are not necessarily accurate for widely separated points on the map. A side effect is that conformal projections preserve the precise perpendicular intersections between parallels and meridians on the map. When mapping smaller areas, relative shape is preserved.	In order for a projection to be conformal, however, consistency in the surface areas, shapes, and/or scales across the map must be sacrificed. An area enclosed by a series of arcs may be greatly distorted.	Hotine Oblique Mercator, Lambert Conformal Conic, Mercator, Oblique Mercator, State Plane Coordinate System, Transverse Mercator, and Universal Transverse Mercator

Equidistant	A *equidistant* projection is when the scale between at least one specific origin point on the map with respect to every other point on the map is represented accurately.	In order for a projection to be equidistant, however, consistency in the surface areas, shapes, and/or angles across the map must be sacrificed.	Azimuthal Equidistant, Equidistant Cylindrical, Equidistant Conic, and Cassini
Azimuthal	A *azimuthal* projection is when the direction of (or angle to) all points on the map are accurate with respect to the center point of the projection.	In order for a projection to be azimuthal, areas, shapes, and angles may be sacrificed at areas not close to the center of the map.	Azimuthal Equidistant, Gnomonic, Lambert Azimuthal Equal Area, Orthographic, and Stereographic
Other	Some projections try to minimize the effects of all distortions and as a result do not minimize any one distortion in particular.		Polyconic, Robinson and Robinson-Sterling, Unprojected Lat/Long, and Van der Grinten

In addition to the characteristics described above, some projections have highly specialized characteristics that may be useful in certain applications. For example, on maps made with a Mercator projection, all lines of constant direction (rhumb lines) are known to be straight, thereby making such maps very desirable as navigational charts.

Supported Projections

Albers Equal Area Conic Projection



World Map
Albers Equal Area Conic Projection
Central Longitude: 0
Central Latitude: 0
Standard Parallel: 45
2nd Standard Parallel: 0

Projection Characteristics

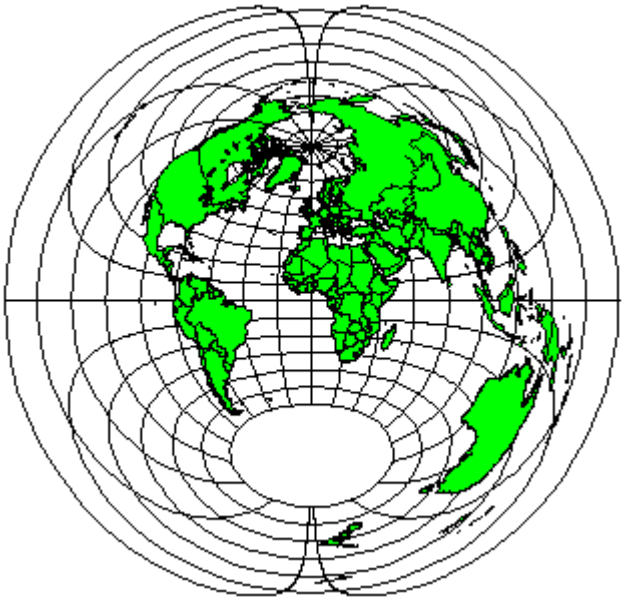
The Albers Equal Area Conic projection scale is constant along any given parallel and accurate along the two specified standard parallels. This projection is used in the National Atlas of the United States. The Albers Equal Area Conic projection is useful for equal area maps of low-aspect regions (regions that are wider than they are tall).

Projection Parameters

Parameter	Description
Name	Specifies the units used in the map.
Scale	Specifies the unit scale of the projected map, relative to *meters*. For example, a scale of one means one unit in the projected map equals one meter; a scale of two means two units in the projected map equal one meter; etc.
False Easting	Specifies the false easting, or horizontal offset, of the projected coordinates, in meters. *False Eastings* and *False Northings* are added to the underlying "projected" coordinates as a way to arbitrarily offset their internal XY coordinates after the projection. Unless you have a reason for using these offset values, do not use them. These values do not affect the latitude/longitude coordinates for the map, only the internal coordinates used to plot the map on the screen. If you use *False Easting* and *False Northing* offsets for a map, any subsequent boundaries you append to the map must also use these same offsets if you want the imported boundaries to be drawn in the correct relative position to the existing boundaries.
False Northing	Specifies the false northing, or vertical offset, of the projected coordinates, in meters. See above.

Central Longitude	Specifies the central longitude of the projection in degrees. The *Central Longitude* value typically should be defined as the longitudinal center of the map you are going to produce. For example, the value -95.5 represents the geographic center of the United States, so the map of the U.S. is drawn upright.
Central Latitude	Specifies the central latitude of the projection in degrees. The *Central Latitude* value should be defined as the latitudinal center of the map you are going to produce. This value is only significant when you define *False Easting* and *False Northing* values.
Standard Parallel	Specifies the latitude of the first of two standard parallels, in degrees. The standard parallels typically are defined at approximately one-sixth of the distance inside the north and south limits of the map. For example, if your map latitude ranges from 30° to 36°, you could place your *Standard Parallels* at 31° and 35°. There are alternative methods for determining the best position of the standard parallels. Please see Snyder for more information.
2nd Standard Parallel	Specifies the latitude of the second of two standard parallels, in degrees. See above.

Azimuthal Equidistant Projection



World Map
Azimuthal Equidistant Projection
Standard Parallel: 0
Central Longitude: 0

Projection Characteristics

The *Azimuthal Equidistant* projection is a planar projection. Directions and scale are true from the center point of the map. Shapes are true at the center of the map, but are distorted the further you move from the center. When using a polar view of this projection, all meridians are straight lines. When using an equatorial view, the central longitude and equator are straight lines, otherwise, only the central longitude is a straight line. This projection is typically used in polar hemispheric maps.

Projection Parameters

Parameter	Description
Name	Specifies the units used in the map.
Scale	Specifies the unit scale of the projected map, relative to *meters*. For example, a scale of one means one unit in the projected map equals one meter; a scale of two means two units in the projected map equal one meter; etc.
False Easting	Specifies the false easting, or horizontal offset, of the projected coordinates, in meters. *False Eastings* and *False Northings* are added to the underlying "projected" coordinates as a way to arbitrarily offset their internal XY coordinates after the projection. Unless you have a reason for using these offset values, do not use them. These values do not affect the latitude/longitude coordinates for the map, only the internal coordinates used to plot the map on the screen. If you use *False Easting* and *False Northing* offsets for a map, any subsequent boundaries you append to the map must also use these same offsets if you want the imported boundaries to be drawn in the correct relative position to the existing boundaries.
False Northing	Specifies the false northing, or vertical offset, of the projected coordinates, in meters. See above.
Central Longitude	Specifies the central longitude of the projection in degrees. The *Central Longitude* value typically should be defined as the longitudinal center of the map you are going to produce.
Standard Parallel	Specifies the central latitude of the projection in degrees. The *Standard Parallel* value typically should be defined as the latitudinal center of the map you are going to produce.

Bonne Projection



North America Map
Bonne Projection
Central Longitude: -100
Standard Parallel: 40

Projection Characteristics

The Bonne projection is a pseudo-conical, equal area projection. The scale is constant along any given parallel, and accurate along the specified standard parallels. The Bonne projection is distortion-free along the central longitude and the parallels. This projection is used for continental and topographic mapping.

Projection Parameters

Parameter	Description
Name	Specifies the units used in the map.
Scale	Specifies the unit scale of the projected map, relative to meters. For example, a scale of one means one unit in the projected map equals one meter; a scale of two means two units in the projected map equal one meter; etc.
False Easting	Specifies the false easting, or horizontal offset, of the projected coordinates, in meters. *False Eastings* and *False Northings* are added to the underlying "projected" coordinates as a way to arbitrarily offset their internal XY coordinates after the projection. Unless you have a reason for using these offset values, do not use them. These values do not affect the latitude/longitude coordinates for the map, only the internal coordinates used to plot the map on the screen. If you use *False Easting* and *False Northing* offsets for a map, any subsequent boundaries you append to the map must also use these same offsets if you want the imported boundaries to be drawn in the correct relative position to the existing boundaries.
False Northing	Specifies the false northing, or vertical offset, of the projected coordinates, in meters. See above.
Central Longitude	Specifies the central longitude of the projection in degrees. The *Central Longitude* value typically should be defined as the longitudinal center of the map you are going to produce.
Standard Parallel	Specifies the central latitude of the projection in degrees. The *Standard Parallel* value typically should be defined as the latitudinal center of the map you are going to produce.

Cassini Projection



Map of North America
Cassini Projection
Central Longitude = -100
Central Latitude = 40

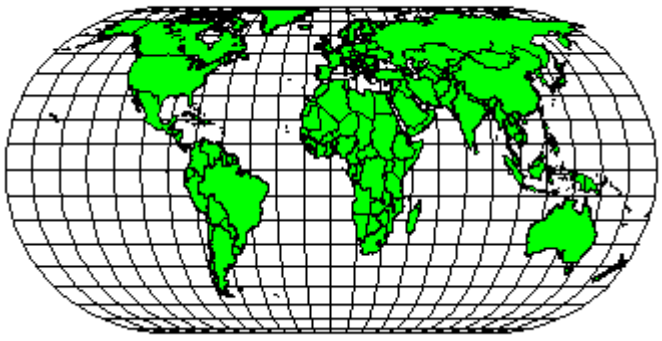
Projection Characteristics

The Cassini projection is a cylindrical projection. The scale is accurate along the central longitude and along latitude lines perpendicular to the central longitude. This projection is useful for high-aspect regions (regions taller than they are wide).

Projection Parameters

Parameter	Description
Name	Specifies the units used in the map.
Scale	Specifies the unit scale of the projected map, relative to meters. For example, a scale of one means one unit in the projected map equals one meter; a scale of two means two units in the projected map equal one meter; etc.
False Easting	Specifies the false easting, or horizontal offset, of the projected coordinates, in meters. *False Eastings* and *False Northings* are added to the underlying "projected" coordinates as a way to arbitrarily offset their internal XY coordinates after the projection. Unless you have a reason for using these offset values, do not use them. These values do not affect the latitude/longitude coordinates for the map, only the internal coordinates used to plot the map on the screen. If you use *False Easting* and *False Northing* offsets for a map, any subsequent boundaries you append to the map must also use these same offsets if you want the imported boundaries to be drawn in the correct relative position to the existing boundaries.
False Northing	Specifies the false northing, or vertical offset, of the projected coordinates, in meters. See above.
Central Latitude	Specifies the central latitude of the projection in degrees. The *Central Latitude* value typically should be defined as the latitudinal center of the map you are going to produce.
Central Longitude	Specifies the central longitude of the projection in degrees. The *Central Longitude* value typically should be defined as the longitudinal center of the map you are going to produce.

Eckert IV Projection



World Map
Eckert IV Projection
Central Longitude: 0

Projection Characteristics

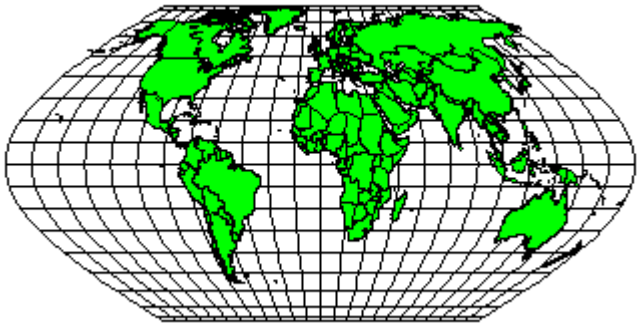
The scale is constant along any given parallel and accurate along the parallels 40°30' north and south in the Eckert IV projection. This is a pseudo-cylindrical, equal area projection with the central longitude half the length of the equator. The poles are represented by lines half the distance of the

equator as well. The meridians are semi-ellipses. Eckert IV is designed to produce aesthetically pleasing world maps.

Projection Parameters

Parameter	Description
Name	Specifies the units used in the map.
Scale	Specifies the unit scale of the projected map, relative to meters. For example, a scale of one means one unit in the projected map equals one meter; a scale of two means two units in the projected map equal one meter; etc.
False Easting	Specifies the false easting, or horizontal offset, of the projected coordinates, in meters. *False Eastings* and *False Northings* are added to the underlying "projected" coordinates as a way to arbitrarily offset their internal XY coordinates after the projection. Unless you have a reason for using these offset values, do not use them. These values do not affect the latitude/longitude coordinates for the map, only the internal coordinates used to plot the map on the screen. If you use *False Easting* and *False Northing* offsets for a map, any subsequent boundaries you append to the map must also use these same offsets if you want the imported boundaries to be drawn in the correct relative position to the existing boundaries.
False Northing	Specifies the false northing, or vertical offset, of the projected coordinates, in meters. See above.
Central Longitude	Specifies the central longitude of the projection in degrees. The *Central Longitude* value typically should be defined as the longitudinal center of the map you are going to produce.

Eckert VI Projection



World Map
Eckert VI Projection
Central Longitude: 0

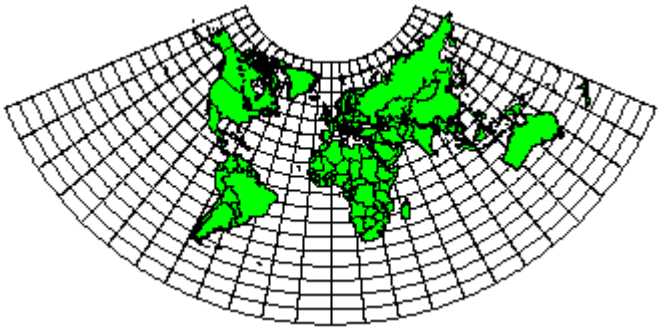
Projection Characteristics

The scale is constant along any given parallel, and accurate along the parallels 49°16' north and south in the Eckert VI projection. This is a pseudo-cylindrical, equal area projection with the central longitude half the length of the equator. The poles are represented by lines half the distance of the equator as well. The meridians are sinusoidal. The Eckert VI projection is useful for world maps.

Projection Parameters

Parameter	Description
Name	Specifies the units used in the map.
Scale	Specifies the unit scale of the projected map, relative to meters. For example, a scale of one means one unit in the projected map equals one meter; a scale of two means two units in the projected map equal one meter; etc.
False Easting	Specifies the false easting, or horizontal offset, of the projected coordinates, in meters. *False Eastings* and *False Northings* are added to the underlying "projected" coordinates as a way to arbitrarily offset their internal XY coordinates after the projection. Unless you have a reason for using these offset values, do not use them. These values do not affect the latitude/longitude coordinates for the map, only the internal coordinates used to plot the map on the screen. If you use *False Easting* and *False Northing* offsets for a map, any subsequent boundaries you append to the map must also use these same offsets if you want the imported boundaries to be drawn in the correct relative position to the existing boundaries.
False Northing	Specifies the false northing, or vertical offset, of the projected coordinates, in meters. See above.
Central Longitude	Specifies the central longitude of the projection in degrees. The *Central Longitude* value typically should be defined as the longitudinal center of the map you are going to produce.

Equidistant Conic Projection



World Map
Equidistant Conic Projection
Central Longitude: 0
Central Latitude: 0
Standard Parallel: 45
2nd Standard Parallel: 0

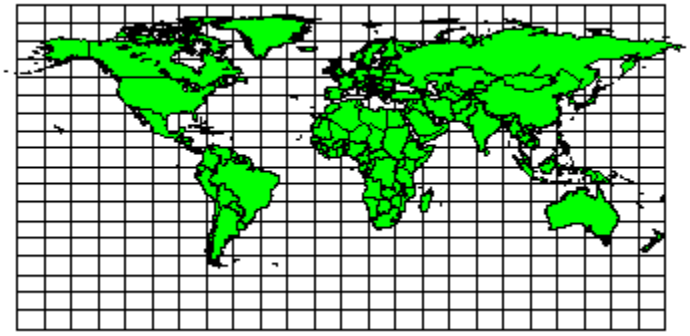
Projection Characteristics

There is no distortion in scale, shape, or area along the standard parallels in an Equidistant Conic projection. Scale is true along all longitudes and along the standard parallels. Direction is locally true along the standard parallels. This projection is useful for maps of low-aspect regions (regions that are wider than they are tall).

Projection Parameters

Parameter	Description
Name	Specifies the units used in the map.
Scale	Specifies the unit scale of the projected map, relative to meters. For example, a scale of one means one unit in the projected map equals one meter; a scale of two means two units in the projected map equal one meter; etc.
False Easting	Specifies the false easting, or horizontal offset, of the projected coordinates, in meters. *False Eastings* and *False Northings* are added to the underlying "projected" coordinates as a way to arbitrarily offset their internal XY coordinates after the projection. Unless you have a reason for using these offset values, do not use them. These values do not affect the latitude/longitude coordinates for the map, only the internal coordinates used to plot the map on the screen. If you use *False Easting* and *False Northing* offsets for a map, any subsequent boundaries you append to the map must also use these same offsets if you want the imported boundaries to be drawn in the correct relative position to the existing boundaries.
False Northing	Specifies the false northing, or vertical offset, of the projected coordinates, in meters. See above.
Central Longitude	Specifies the central longitude of the projection in degrees. The *Central Longitude* value typically should be defined as the longitudinal center of the map you are going to produce. For example, the value -95.5 represents the geographic center of the United States, so the map of the U.S. is drawn upright.
Central Latitude	Specifies the central latitude of the projection in degrees. The *Central Latitude* value should be defined as the latitudinal center of the map you are going to produce. This value is only significant when you define *False Easting* and *False Northing* values.
Standard Parallel	Specifies the latitude of the first of two standard parallels, in degrees. The standard parallels typically are defined at approximately one-sixth of the distance inside the north and south limits of the map. For example, if your map latitude ranges from 30° to 36°, you could place your *Standard Parallels* at 31° and 35°. There are alternative methods for determining the best position of the standard parallels. Please see Snyder for more information.
2nd Standard Parallel	Specifies the latitude of the second of two standard parallels, in degrees. See above.

Equidistant Cylindrical Projection



World Map
Equidistant Cylindrical Projection
Central Longitude: 0
Standard Parallel: 0

Projection Characteristics

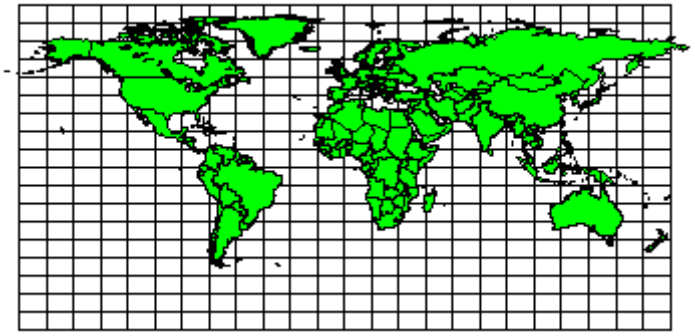
The coordinates are equidistant with respect to the center of the Equidistant Cylindrical projection. Distortion is minimal at the specified standard parallel and increases dramatically with distance north or south from this parallel. Typically, this projection is used with maps covering small areas.

Projection Parameters

Parameter	Description
Name	Specifies the units used in the map.
Scale	Specifies the unit scale of the projected map, relative to meters. For example, a scale of one means one unit in the projected map equals one meter; a scale of two means two units in the projected map equal one meter; etc.
False Easting	Specifies the false easting, or horizontal offset, of the projected coordinates, in meters. *False Eastings* and *False Northings* are added to the underlying "projected" coordinates as a way to arbitrarily offset their internal XY coordinates after the projection. Unless you have a reason for using these offset values, do not use them. These values do not affect the latitude/longitude coordinates for the map, only the internal coordinates used to plot the map on the screen. If you use *False Easting* and *False Northing* offsets for a map, any subsequent boundaries you append to the map must also use these same offsets if you want the imported boundaries to be drawn in the correct relative position to the existing boundaries.
False Northing	Specifies the false northing, or vertical offset, of the projected coordinates, in meters. See above.
Central Longitude	Specifies the central longitude of the projection in degrees. The *Central Longitude* value typically should be defined as the longitudinal center of the map you are going to produce. This value is only significant when you define *False Easting* and *False Northing* values, and has no apparent effect on the map.

Standard Parallel	Specifies the central latitude of the projection in degrees. The *Standard Parallel* value typically should be defined as the latitudinal center of the map you are going to produce.

Geographic Coordinate System



World Map
Unprojected Lat/Long

Projection Characteristics

This is a simplistic cylindrical projection. If the coordinates in a **Voxler** map layer are stored in a geographic coordinate system (also known as Unprojected Lat/Long or latitude/longitude), the map is displayed on the screen by simply treating the longitudes as horizontal offsets and the latitudes as vertical offsets.

When plotting latitude/longitude coordinates, **Voxler** constructs the map coordinate system by first determining the latitude for the center of the map. Then the appropriate scale is determined for the east-west (longitude) relative to the north-south (latitude) dimension of the map. See latitude/longitude for more information on relative scaling of latitude versus longitude.

The features of a geographic coordinate system (Unprojected Lat/Long) map are:

- Meridians (lines of constant longitude) are equally spaced and are drawn perpendicular to the parallels.
- Parallels (lines of constant latitude) are equally spaced over the entire map and are drawn perpendicular to the meridians. The further your map area is from the equator, the further apart the parallels are spaced.
- The spacing between meridians is different than the spacing between parallels except when the equator is at the north-south center of the map.
- At the center of the map, the scale is accurate in both the north-south and east-west direction.
- Scale is accurate in the north-south direction over the extent of the map.
- East-west scale increases towards the poles, and decreases towards the equator.

There are no parameters for this projection.

Gnomonic Projection



Antarctica
Gnomonic Projection
False Easting: 0
False Northing: 0
Central Longitude: 19.0
Standard Parallel: 2.4

Projection Characteristics

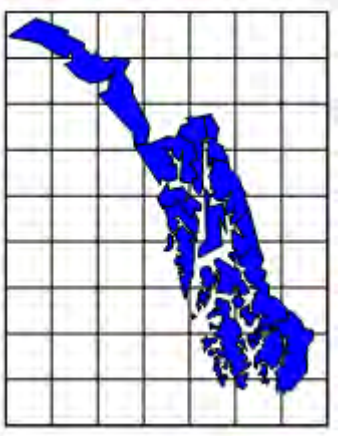
The Gnomonic projection is an azimuthal projection. It is represented as a plane tangent to the globe. At this point of tangency, which is called the standard parallel, all major characteristics are retained. When you move away from the standard parallel in any direction the map is not conformal, not equal-area, and distances are not true to scale. Only areas of less than a hemisphere can be shown and distortion increases noticeably as you move further from the standard parallel.

Projection Parameters

Parameter	Description
Name	Specifies the units used in the map.
Scale	Specifies the unit scale of the projected map, relative to meters. For example, a scale of one means one unit in the projected map equals one meter; a scale of two means two units in the projected map equal one meter; etc.

False Easting	Specifies the false easting, or horizontal offset, of the projected coordinates, in meters. *False Eastings* and *False Northings* are added to the underlying "projected" coordinates as a way to arbitrarily offset their internal XY coordinates after the projection. Unless you have a reason for using these offset values, do not use them. These values do not affect the latitude/longitude coordinates for the map, only the internal coordinates used to plot the map on the screen. If you use *False Easting* and *False Northing* offsets for a map, any subsequent boundaries you append to the map must also use these same offsets if you want the imported boundaries to be drawn in the correct relative position to the existing boundaries.
False Northing	Specifies the false northing, or vertical offset, of the projected coordinates, in meters. See above.
Central Longitude	Specifies the central longitude of the projection in degrees. The *Central Longitude* value typically should be defined as the longitudinal center of the map you are going to produce. This value is only significant when you define *False Easting* and *False Northing* values, and has no apparent effect on the map.
Standard Parallel	Specifies the central latitude of the projection in degrees. The *Standard Parallel* value typically should be defined as the latitudinal center of the map you are going to produce.

Hotine Oblique Mercator 2-Point Projection



Aleutian Islands, Alaska
Hotine Oblique Mercator Projection
Scale = 1
False Easting = 0 False Northing = 0
Central Scale Factor = 1 Central Latitude = 0
Standard Parallel = 56 2nd Standard Parallel = 58
1st Meridian = -133 2nd Meridian = -135
Rotate U/V to X/Y = True
Offset by U = True

Projection Characteristics

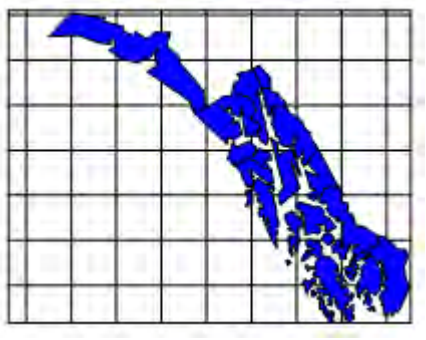
The Hotine Oblique Mercator 2-Point projection is a cylindrical, conformal projection. The scale is accurate along the chosen central line by the longitude/latitude settings below. This projection is useful for oblique areas (areas that do not follow lines of latitude and longitude), such as the Alaska panhandle because the central line does not have to follow a line of constant longitude. This projection is typically used with areas that are state or province sized and it is not suitable for maps

of the world. There are two forms of the Hotine projection equation. The *Hotine Oblique Mercator 2-Point* projection is defined by selecting two points to form a line (Alternate A, *Snyder*).

Projection Parameters

Parameter	Description
Name	Specifies the units used in the map.
Scale	Specifies the unit scale of the projected map, relative to meters. For example, a scale of one means one unit in the projected map equals one meter; a scale of two means two units in the projected map equal one meter; etc.
False Easting	Specifies the false easting, or horizontal offset, of the projected coordinates, in meters. *False Eastings* and *False Northings* are added to the underlying "projected" coordinates as a way to arbitrarily offset their internal XY coordinates after the projection. Unless you have a reason for using these offset values, do not use them. These values do not affect the latitude/longitude coordinates for the map, only the internal coordinates used to plot the map on the screen. If you use *False Easting* and *False Northing* offsets for a map, any subsequent boundaries you append to the map must also use these same offsets if you want the imported boundaries to be drawn in the correct relative position to the existing boundaries.
False Northing	Specifies the false northing, or vertical offset, of the projected coordinates, in meters. See above.
Central Scale Factor (KO)	Specifies the central scaling factor for the projection. This value is often set to 1.0, but may be set to another value for specific applications.
Central Latitude	Specifies the central latitude of the projection in degrees. The *Central Latitude* value typically should be defined as the latitudinal center of the map you are going to produce, and should typically be defined as the center of the map.
Standard Parallel	Defines one end of the Y extent for the central line.
1st Meridian	Defines one end of the X extent for the central line.
2nd Standard Parallel	Defines the other end of the Y extent for the central line.
2nd Meridian	Defines the other end of the X extent for the central line.
Rotate U/V to X/Y	u,v are unrectified coordinates that follow the central line of the projection. x,y are rectified rectangular coordinates. When True, u,v are rotated to x,y. See *Snyder* page 70.
Offset by U	When True, u coordinates are offset to remove the *Us* center component, to normalize the origin of the u axis. This is typically required for State Plane coordinate systems that use the Hotine Oblique Mercator projection method.

Hotine Oblique Mercator Projection



Alaska Panhandle
Hotine Oblique Mercator Projection
Scale = 1
False Easting = 818676.73440112
False Northing = 575097.68887519
Central Scale Factor = .9999
Azimuth (Alpha) = -36.8698976
Central Latitude = 57
1st Meridian = -133.6667
Rotate U/V to X/Y = True
Offset by U = True

Projection Characteristics

The Hotine Oblique Mercator projection is a cylindrical, conformal projection. The scale is accurate along the chosen central line by the longitude/latitude settings below. This projection is useful for oblique areas (areas that do not follow lines of latitude and longitude), such as the Alaska panhandle because the central line does not have to follow a line of constant longitude. This projection is typically used with areas that are state or province sized and it is not suitable for maps of the world. There are two forms of the Hotine projection equation. The *Hotine Oblique Mercator* projection is defined by a point and an azimuth that defines a line (Alternate B, Snyder).

Projection Parameters

Parameter	Description
Name	Specifies the units used in the map.
Scale	Specifies the unit scale of the projected map, relative to meters. For example, a scale of one means one unit in the projected map equals one meter; a scale of two means two units in the projected map equal one meter; etc.
False Easting	Specifies the false easting, or horizontal offset, of the projected coordinates, in meters. *False Eastings* and *False Northings* are added to the underlying "projected" coordinates as a way to arbitrarily offset their internal XY coordinates after the projection. Unless you have a reason for using these offset values, do not use them. These values do not affect the latitude/longitude coordinates for the map, only the internal coordinates used to plot the map on the screen. If you use *False Easting* and *False Northing* offsets for a map, any subsequent boundaries you append to the map must also use these same offsets if you want the imported boundaries to be drawn in the correct relative position to the existing boundaries.

False Northing	Specifies the false northing, or vertical offset, of the projected coordinates, in meters. See above.
Central Scale Factor (KO)	Specifies the central scaling factor for the projection. This value is often set to 1.0, but may be set to another value for specific applications.
Azimuth (Alpha)	The angle in degrees in which to rotate the central line. Zero is north, and rotation is clockwise. Value must be greater than -360 and less than +360. Value cannot equal 0, 360, or -360. Value cannot equal 90, 270, -90, or -270 if *Offset by U* is set to *False*.
Central Latitude	Specifies the central latitude of the projection in degrees. The *Central Latitude* value typically should be defined as the latitudinal center of the map you are going to produce, and should typically be defined as the center of the map.
1st Meridian	Defines one end of the X extent for the central line.
Rotate U/V to X/Y	u,v are unrectified coordinates that follow the central line of the projection. x,y are rectified rectangular coordinates. When True, u,v are rotated to x,y. See *Snyder* page 70.
Offset by U	When True, u coordinates are offset to remove the *Us* center component, to normalize the origin of the u axis. This is typically required for State Plane coordinate systems that use the Hotine Oblique Mercator projection method.

Lambert Azimuthal Equal Area Projection



World Map
Lambert Azimuthal Equal Area Projection
Central Longitude: -90
Standard Parallel: 45

Projection Characteristics

Scale on a Lambert Azimuthal Equal Area projected map is accurate only from the center to any other point on the map. Distortion is minimal near the center and increases with distance from the center. This projection is useful for continents, polar regions (hemispheres), or smaller regions. The Lambert Azimuthal Equal Area projection is not generally used for world maps due to extreme distortion outside the center of the map.

Projection Parameters

Parameter	Description
Name	Specifies the units used in the map.
Scale	Specifies the unit scale of the projected map, relative to meters. For example, a scale of one means one unit in the projected map equals one meter; a scale of two means two units in the projected map equal one meter; etc.
False Easting	Specifies the false easting, or horizontal offset, of the projected coordinates, in meters. *False Eastings* and *False Northings* are added to the underlying "projected" coordinates as a way to arbitrarily offset their internal XY coordinates after the projection. Unless you have a reason for using these offset values, do not use them. These values do not affect the latitude/longitude coordinates for the map, only the internal coordinates used to plot the map on the screen. If you use *False Easting* and *False Northing* offsets for a map, any subsequent boundaries you append to the map must also use these same offsets if you want the imported boundaries to be drawn in the correct relative position to the existing boundaries.
False Northing	Specifies the false northing, or vertical offset, of the projected coordinates, in meters. See above.
Central Longitude	Specifies the central longitude of the projection in degrees. The *Central Longitude* value typically should be defined as the longitudinal center of the map you are going to produce.
Standard Parallel	Specifies the central latitude of the projection in degrees. The *Standard Parallel* value typically should be defined as the latitudinal center of the map you are going to produce.

Lambert Conformal Conic Projection



World Map
Lambert Conformal Conic Projection
Central Longitude: 0
Central Latitude: 0
Standard Parallel: 45
2nd Standard Parallel: 0

Projection Characteristics

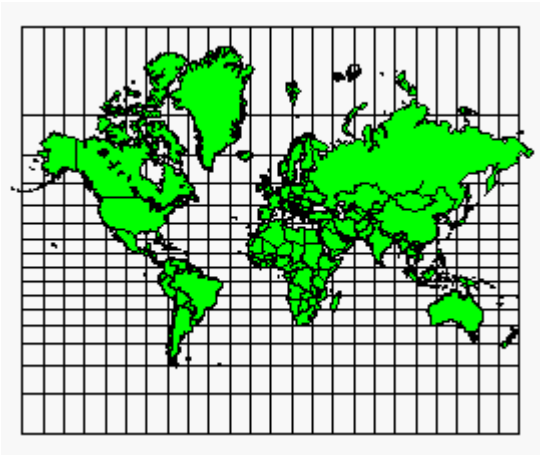
In a Lambert Conformal Conic projection, scale is constant along any given parallel and accurate along the specified standard parallels. Scale is the same in all directions at any given point. This projection is useful for equal area maps of low-aspect regions (regions that are wider than they are

tall). The pole is a point in the hemisphere containing the standard parallels and the graticules stretch to infinity in the other hemisphere. The Lambert Conformal Conic projection is used for many of the zones in the State Plane Coordinate System.

Projection Parameters

Parameter	Description
Name	Specifies the units used in the map.
Scale	Specifies the unit scale of the projected map, relative to meters. For example, a scale of one means one unit in the projected map equals one meter; a scale of two means two units in the projected map equal one meter; etc.
False Easting	Specifies the false easting, or horizontal offset, of the projected coordinates, in meters. *False Eastings* and *False Northings* are added to the underlying "projected" coordinates as a way to arbitrarily offset their internal XY coordinates after the projection. Unless you have a reason for using these offset values, do not use them. These values do not affect the latitude/longitude coordinates for the map, only the internal coordinates used to plot the map on the screen. If you use *False Easting* and *False Northing* offsets for a map, any subsequent boundaries you append to the map must also use these same offsets if you want the imported boundaries to be drawn in the correct relative position to the existing boundaries.
False Northing	Specifies the false northing, or vertical offset, of the projected coordinates, in meters. See above.
Central Longitude	Specifies the central longitude of the projection in degrees. The *Central Longitude* value typically should be defined as the longitudinal center of the map you are going to produce. For example, the value -95.5 represents the geographic center of the United States, so the map of the U.S. is drawn upright.
Central Latitude	Specifies the central latitude of the projection in degrees. The *Central Latitude* value should be defined as the latitudinal center of the map you are going to produce. This value is only significant when you define *False Easting* and *False Northing* values.
Standard Parallel	Specifies the latitude of the first of two standard parallels, in degrees. The *Standard Parallels* typically are defined at approximately one-sixth of the distance inside the north and south limits of the map. For example, if your map latitude ranges from 30° to 36°, you could place your *Standard Parallels* at 31° and 35°. There are alternative methods for determining the best position of the standard parallels. Please see Snyder for more information.
2nd Standard Parallel	Specifies the latitude of the second of two standard parallels, in degrees. See above.

Mercator Projection



World Map
Mercator Projection
Central Longitude: 0
Central Latitude: 0

Projection Characteristics

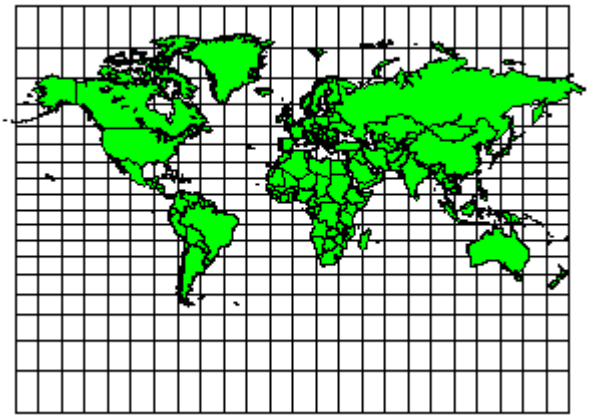
The Mercator projection is a cylindrical projection and it is conformal. In a Mercator projection, scale is constant along any given parallel and accurate along the specified center latitude. Scale is the same in all directions near any given point. Distortion is minimal near the center parallel, but becomes extreme toward the poles. All lines of constant direction (rhumb lines) are known to be straight, thereby making this projection very desirable for producing navigational charts. A limitation of this projection is that coordinates at or near the poles cannot be projected due to constraints of the mathematical formulas used.

Projection Parameters

Parameter	Description
Name	Specifies the units used in the map.
Scale	Specifies the unit scale of the projected map, relative to meters. For example, a scale of one means one unit in the projected map equals one meter; a scale of two means two units in the projected map equal one meter; etc.
False Easting	Specifies the false easting, or horizontal offset, of the projected coordinates, in meters. *False Eastings* and *False Northings* are added to the underlying "projected" coordinates as a way to arbitrarily offset their internal XY coordinates after the projection. Unless you have a reason for using these offset values, do not use them. These values do not affect the latitude/longitude coordinates for the map, only the internal coordinates used to plot the map on the screen. If you use *False Easting* and *False Northing* offsets for a map, any subsequent boundaries you append to the map must also use these same offsets if you want the imported boundaries to be drawn in the correct relative position to the existing boundaries.
False Northing	Specifies the false northing, or vertical offset, of the projected coordinates, in meters. See above.

Central Longitude	Specifies the central longitude of the projection in degrees. The *Central Longitude* value typically should be defined as the longitudinal center of the map you are going to produce.
Central Latitude	Specifies the central latitude of the projection in degrees. The *Central Latitude* value typically should be defined as the latitudinal center of the map you are going to produce.

Miller Cylindrical Projection



World Map
Miller Cylindrical Projection
Central Longitude: 0

Projection Characteristics

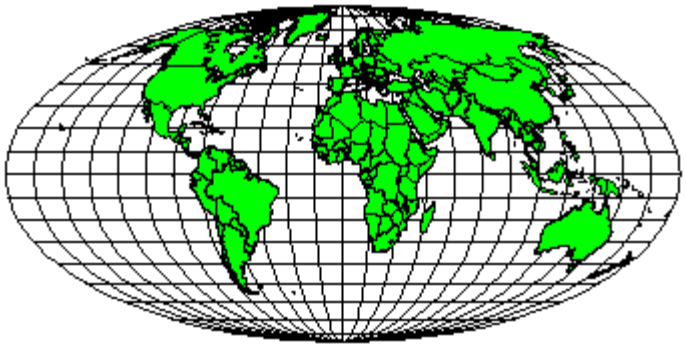
Scale is constant along any given parallel and accurate along the equator in a Miller Cylindrical projection. Scale is the same in all directions near any given point. Miller Cylindrical projection maps use variable latitudinal scale as a way to minimize distortion as you move north or south from the equator. The method effectively corrects for the relative distances covered by one degree of longitude relative to one degree of latitude as you move away from the equator. Distortion is minimal near the equator, but becomes extreme toward the poles. Miller Cylindrical maps do not represent relative land areas accurately, but do approximate the relative shapes of individual land areas. Miller Cylindrical projection maps are useful for displaying the entire world.

Projection Parameters

Parameter	Description
Name	Specifies the units used in the map.
Scale	Specifies the unit scale of the projected map, relative to meters. For example, a scale of one means one unit in the projected map equals one meter; a scale of two means two units in the projected map equal one meter; etc.

False Easting	Specifies the false easting, or horizontal offset, of the projected coordinates, in meters. *False Eastings* and *False Northings* are added to the underlying "projected" coordinates as a way to arbitrarily offset their internal XY coordinates after the projection. Unless you have a reason for using these offset values, do not use them. These values do not affect the latitude/longitude coordinates for the map, only the internal coordinates used to plot the map on the screen. If you use *False Easting* and *False Northing* offsets for a map, any subsequent boundaries you append to the map must also use these same offsets if you want the imported boundaries to be drawn in the correct relative position to the existing boundaries.
False Northing	Specifies the false northing, or vertical offset, of the projected coordinates, in meters. See above.
Central Longitude	Specifies the central longitude of the projection in degrees. This value should be defined as the longitudinal center of the map you are going to produce. This value is only significant when you define *False Easting* and *False Northing* values and it has no apparent effect on the map.

Mollweide Projection



World Map
Mollweide Projection
Central Longitude: 0

Projection Characteristics

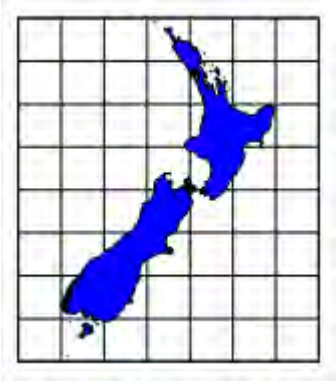
The Mollweide projection is a pseudo-cylindrical, equal area projection. Scale is constant along any given parallel, and true along 40°44' north and south. The central longitude is half the length of the equator. This projection was designed to produce aesthetically pleasing world maps.

Projection Parameters

Parameter	Description
Name	Specifies the units used in the map.
Scale	Specifies the unit scale of the projected map, relative to meters. For example, a scale of one means one unit in the projected map equals one meter; a scale of two means two units in the projected map equal one meter; etc.

False Easting	Specifies the false easting, or horizontal offset, of the projected coordinates, in meters. *False Eastings* and *False Northings* are added to the underlying "projected" coordinates as a way to arbitrarily offset their internal XY coordinates after the projection. Unless you have a reason for using these offset values, do not use them. These values do not affect the latitude/longitude coordinates for the map, only the internal coordinates used to plot the map on the screen. If you use *False Easting* and *False Northing* offsets for a map, any subsequent boundaries you append to the map must also use these same offsets if you want the imported boundaries to be drawn in the correct relative position to the existing boundaries.
False Northing	Specifies the false northing, or vertical offset, of the projected coordinates, in meters. See above.
Central Longitude	Specifies the central longitude of the projection in degrees. The *Central Longitude* value typically should be defined as the longitudinal center of the map you are going to produce.

New Zealand Map Grid



New Zealand
New Zealand Map Grid Projection
False Easting: 2510000
False Northing: 6023150
Scale = 1

Projection Characteristics

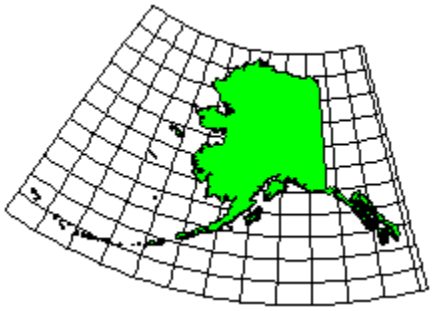
The New Zealand Map Grid projection is a modified cylindrical projection and is conformal. It is a sixth-order conformal modification of the Mercator projection using the International spheroid. Scale is constant along any given parallel and is highly accurate for New Zealand. Scale is the same in all directions near any given point. Distortion is minimal near 173° East, 41° South, and becomes more distorted the further from this location. This projection should only be used for large-scale maps of New Zealand and is not useful outside of New Zealand.

Projection Parameters

Parameter	Description
Name	Specifies the units used in the map.
Scale	Specifies the unit scale of the projected map, relative to meters. For example, a scale of one means one unit in the projected map equals one meter; a scale of two means two units in the projected map equal one meter; etc.

False Easting	Specifies the false easting, or horizontal offset, of the projected coordinates, in meters. *False Eastings* and *False Northings* are added to the underlying "projected" coordinates as a way to arbitrarily offset their internal XY coordinates after the projection. Unless you have a reason for using these offset values, do not change them. These values do not affect the latitude/longitude coordinates for the map, only the internal coordinates used to plot the map on the screen. If you use *False Easting* and *False Northing* offsets for a map, any subsequent boundaries you append to the map must also use these same offsets if you want the imported boundaries to be drawn in the correct relative position to the existing boundaries.
False Northing	Specifies the false northing, or vertical offset, of the projected coordinates, in meters. See above.

Oblique Mercator Projection



State of Alaska
Oblique Mercator Projection
Scale = 1
Central Scale Factor (KO) = 1
1st Meridian = -132.27
Standard Parallel = 55.95
2nd Meridian = -177.22
2nd Standard Parallel = 52.45

Projection Characteristics

The Oblique Mercator projection is a cylindrical, conformal projection. The scale is accurate along the chosen central line by the longitude and latitude settings below (*1st Point* and *2nd Point*). This projection is used for oblique areas which are areas that do not follow lines of latitude and longitude, such as the Alaska panhandle. This projection is usually used with areas that are state or province sized. This projection is not suitable for maps of the world.

Projection Parameters

Parameter	Description
Name	Specifies the units used in the map.
Scale	Specifies the unit scale of the projected map, relative to meters. For example, a scale of one means one unit in the projected map equals one meter; a scale of two means two units in the projected map equal one meter; etc.

False Easting	Specifies the false easting, or horizontal offset, of the projected coordinates, in meters. *False Eastings* and *False Northings* are added to the underlying "projected" coordinates as a way to arbitrarily offset their internal XY coordinates after the projection. Unless you have a reason for using these offset values, do not use them. These values do not affect the latitude/longitude coordinates for the map, only the internal coordinates used to plot the map on the screen. If you use *False Easting* and *False Northing* offsets for a map, any subsequent boundaries you append to the map must also use these same offsets if you want the imported boundaries to be drawn in the correct relative position to the existing boundaries.
False Northing	Specifies the false northing, or vertical offset, of the projected coordinates, in meters. See above.
1st Meridian	Defines one end of the X extent for the central line.
Standard Parallel	Defines one end of the Y extent for the central line.
2nd Meridian	Defines the other end of the X extent for the central line.
2nd Standard Parallel	Defines the other end of the Y extent for the central line.
Central Scale Factor (KO)	Specifies the central scaling factor for the projection. This value is often set to 1.0, but may be set to another value for specific applications.

Orthographic Projection



World Map
Orthographic Projection
Central Longitude: -90
Central Latitude: 45

Projection Characteristics

The Orthographic projection is an azimuthal projection. In an Orthographic projection, scale is accurate at the center and along any circle circumscribed around the center. Distortion is nil at the center, and increasingly extreme with increasing distance from the center. This projection is useful for "view of globe" or "view from space" pictures of the Earth. A limitation of this projection is that the hemisphere facing away from the center of the projection is not visible. Objects near the edge of the visible hemisphere may be clipped.

Projection Parameters

Parameter	Description

Name	Specifies the units used in the map.
Scale	Specifies the unit scale of the projected map, relative to meters. For example, a scale of one means one unit in the projected map equals one meter; a scale of two means two units in the projected map equal one meter; etc.
False Easting	Specifies the false easting, or horizontal offset, of the projected coordinates, in meters. *False Eastings* and *False Northings* are added to the underlying "projected" coordinates as a way to arbitrarily offset their internal XY coordinates after the projection. Unless you have a reason for using these offset values, do not use them. These values do not affect the latitude/longitude coordinates for the map, only the internal coordinates used to plot the map on the screen. If you use *False Easting* and *False Northing* offsets for a map, any subsequent boundaries you append to the map must also use these same offsets if you want the imported boundaries to be drawn in the correct relative position to the existing boundaries.
False Northing	Specifies the false northing, or vertical offset, of the projected coordinates, in meters. See above.
Central Longitude	Specifies the central longitude of the projection in degrees. The *Central Longitude* value typically should be defined as the longitudinal center of the map you are going to produce.
Central Latitude	Specifies the central latitude of the projection in degrees. The *Central Latitude* value typically should be defined as the latitudinal center of the map you are going to produce.

Polyconic Projection



World Map
Polyconic Projection
Central Longitude: 0
Central Latitude: 0

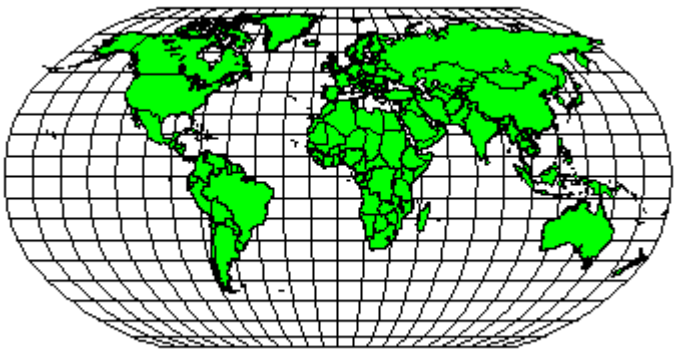
Projection Characteristics

The Polyconic projection is useful for maps of continental or smaller regions. Generally, this projection is not used for world maps due to extreme distortion at any significant distance from the center of the projection. Only the central meridian is distortion-free. Notice in the sample map shown above that Africa is relatively undistorted, but the rest of the world is barely recognizable. Scale is true along the central longitude and along each parallel.

Projection Parameters

Parameter	Description
Name	Specifies the units used in the map.
Scale	Specifies the unit scale of the projected map, relative to meters. For example, a scale of one means one unit in the projected map equals one meter; a scale of two means two units in the projected map equal one meter; etc.
False Easting	Specifies the false easting, or horizontal offset, of the projected coordinates, in meters. *False Eastings* and *False Northings* are added to the underlying "projected" coordinates as a way to arbitrarily offset their internal XY coordinates after the projection. Unless you have a reason for using these offset values, do not use them. These values do not affect the latitude/longitude coordinates for the map, only the internal coordinates used to plot the map on the screen. If you use *False Easting* and *False Northing* offsets for a map, any subsequent boundaries you append to the map must also use these same offsets if you want the imported boundaries to be drawn in the correct relative position to the existing boundaries.
False Northing	Specifies the false northing, or vertical offset, of the projected coordinates, in meters. See above.
Central Longitude	Specifies the central longitude of the projection in degrees. The *Central Longitude* value typically should be defined as the longitudinal center of the map you are going to produce.
Central Latitude	Specifies the central latitude of the projection in degrees. The *Central Latitude* value typically should be defined as the latitudinal center of the map you are going to produce.

Robinson and Robinson-Sterling Projections



World Map
Robinson-Sterling Projection
Central Longitude: 0

Projection Characteristics

These projections are pseudo-cylindrical. Scale and area are always distorted by the Robinson and Robinson-Sterling projections. These projections are designed to produce aesthetically pleasing world maps.

The Robinson and Robinson-Sterling projections produce visually similar results, but use two entirely different mathematical processes. As a result, the numeric values of the projected coordinates produced by these two methods are slightly different. For most applications, these differences are not significant.

Projection Parameters

Parameter	Description
Name	Specifies the units used in the map.
Scale	Specifies the unit scale of the projected map, relative to meters. For example, a scale of one means one unit in the projected map equals one meter; a scale of two means two units in the projected map equal one meter; etc.
False Easting	Specifies the false easting, or horizontal offset, of the projected coordinates, in meters. *False Eastings* and *False Northings* are added to the underlying "projected" coordinates as a way to arbitrarily offset their internal XY coordinates after the projection. Unless you have a reason for using these offset values, do not use them. These values do not affect the latitude/longitude coordinates for the map, only the internal coordinates used to plot the map on the screen. If you use *False Easting* and *False Northing* offsets for a map, any subsequent boundaries you append to the map must also use these same offsets if you want the imported boundaries to be drawn in the correct relative position to the existing boundaries.
False Northing	Specifies the false northing, or vertical offset, of the projected coordinates, in meters. See above.
Central Longitude	Specifies the central longitude of the projection in degrees. The *Central Longitude* value typically should be defined as the longitudinal center of the map you are going to produce. This option is only available for the Robinson-Sterling projection.

Sinusoidal Projection



World Map
Sinusoidal Projection
Central Longitude: 0

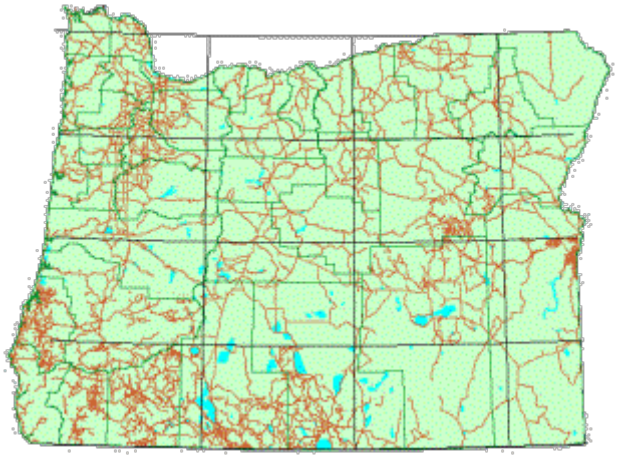
Projection Characteristics

The Sinusoidal projection is a pseudo-cylindrical, equal area projection. Scale is accurate along any given parallel and along the specified central longitude. This projection is useful for continental or world maps, particularly for high-aspect regions (regions taller than they are wide). To get good results with the Sinusoidal projection, the map must have coordinates between +/-180 degrees longitude and +/- 90 degrees latitude.

Projection Parameters

Parameter	Description
Name	Specifies the units used in the map.
Scale	Specifies the unit scale of the projected map, relative to meters. For example, a scale of one means one unit in the projected map equals one meter; a scale of two means two units in the projected map equal one meter; etc.
False Easting	Specifies the false easting, or horizontal offset, of the projected coordinates, in meters. *False Eastings* and *False Northings* are added to the underlying "projected" coordinates as a way to arbitrarily offset their internal XY coordinates after the projection. Unless you have a reason for using these offset values, do not use them. These values do not affect the latitude/longitude coordinates for the map, only the internal coordinates used to plot the map on the screen. If you use *False Easting* and *False Northing* offsets for a map, any subsequent boundaries you append to the map must also use these same offsets if you want the imported boundaries to be drawn in the correct relative position to the existing boundaries.
False Northing	Specifies the false northing, or vertical offset, of the projected coordinates, in meters. See above.
Central Longitude	Specifies the central longitude of the projection in degrees. The *Central Longitude* value typically should be defined as the longitudinal center of the map you are going to produce.

State Plane Coordinate System Projections



State of Oregon
State Plane Coordinate System of 1983
Projection for Oregon North Zone

Projection Characteristics

The State Plane Coordinate System (SPCS) divides the United States into a number of zones, and defines a different projection for each zone such that a suitable map of any given zone is plotted. SPCS is used mainly for intrastate views such as county or parish maps.

Unlike most forms of projection where the datum may be specified separately, the SPCS is tied to a specific datum. There are two State Plane Coordinate Systems commonly used. The State Plane Coordinate System of 1927 uses the North American Datum 1927 (NAD27), while the State Plane Coordinate System of 1983 uses the North American Datum 1983 (NAD83).

Projection Parameters

Parameter	Description
Zone	Specifies which one of the predefined zone projections to use for this coordinate system.
Feet or Meters	Most SPCS have both a meters and feet option available in the predefined list.

Stereographic Projection



North America
(Other continents included to show distortion of objects far away from projection center)
Stereographic Projection
Central Longitude: -90
Central Latitude: 45

Projection Characteristics

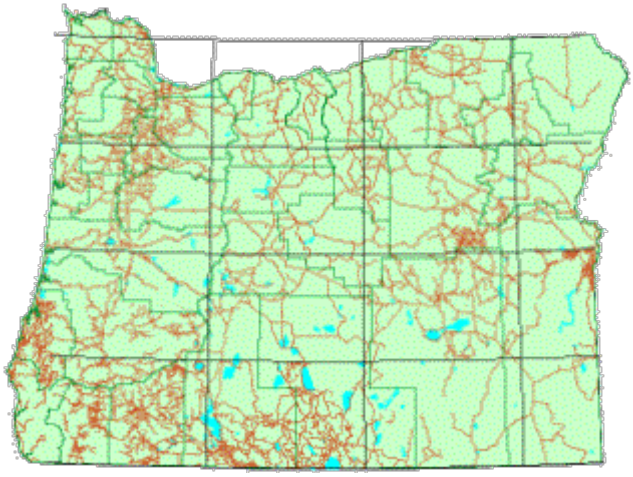
The Stereographic projection is an azimuthal, conformal projection. In a Stereographic projection, scale is constant along any circle circumscribed around the center of the projection. Distortion is minimal at the center and becomes extreme with distance from the center. Generally, this projection is not used for regions larger than a continent or a hemisphere due to distortion effects. The Stereographic projection is often used for maps of the poles.

Projection Parameters

Parameter	Description
Name	Specifies the units used in the map.
Scale	Specifies the unit scale of the projected map, relative to meters. For example, a scale of one means one unit in the projected map equals one meter; a scale of two means two units in the projected map equal one meter; etc.
False Easting	Specifies the false easting, or horizontal offset, of the projected coordinates, in meters. *False Eastings* and *False Northings* are added to the underlying "projected" coordinates as a way to arbitrarily offset their internal XY coordinates after the projection. Unless you have a reason for using these offset values, do not use them. These values do not affect the latitude/longitude coordinates for the map, only the internal coordinates used to plot the map on the screen. If you use *False Easting* and *False Northing* offsets for a map, any subsequent boundaries you append to the map must also use these same offsets if you want the imported boundaries to be drawn in the correct relative position to the existing boundaries.
False Northing	Specifies the false northing, or vertical offset, of the projected coordinates, in meters. See above.
Central Scale Factor (KO)	Specifies the central scaling factor for the projection. This value is often set to 1.0, but may be set to another value for specific applications.

Central Longitude	Specifies the central longitude of the projection in degrees. The *Central Longitude* value typically should be defined as the longitudinal center of the map you are going to produce.
Central Latitude	Specifies the central latitude of the projection in degrees. The *Central Latitude* value typically should be defined as the latitudinal center of the map you are going to produce.

Transverse Mercator Projection



State of Oregon
Transverse Mercator Projection
Central Latitude: 41.75
Central Longitude: -120.5

Projection Characteristics

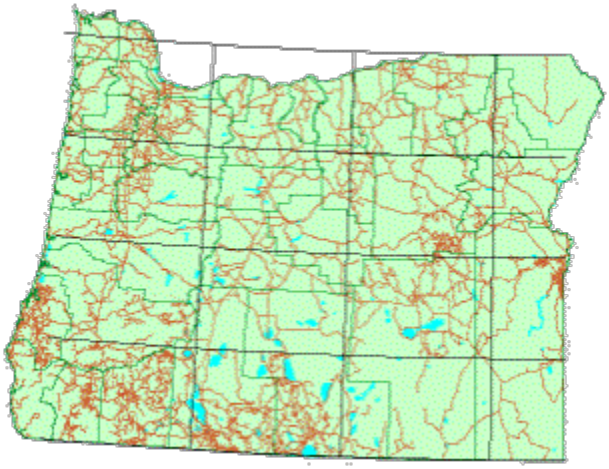
The Transverse Mercator projection is also known as the Gauss-Kruger projection or the Gauss Conformal projection. This projection is cylindrical and conformal. In this projection, scale is constant along any straight line that is parallel to the specified central meridian. Scale increases with distance from the central meridian. Distortion is minimal near the center of the projection and increases dramatically with distance from the center. Distortion is considerable when projecting coordinates that fall within a few degrees of the poles. This projection is useful primarily for mapping small regions no more than a few degrees across, particularly high-aspect regions (regions taller than they are wide). A limitation of this projection is that coordinates at or near the poles cannot be projected. An additional limitation is that regions larger than a quadrant (e.g. having greater than 90 degrees extent either vertically or horizontally) result in portions of the projected image folding over each other due to constraints of the mathematical formulas used. The projection is best used with areas that are no greater than 30 degrees wide or tall, and preferably with areas that are much smaller.

Projection Parameters

Parameter	Description
Name	Specifies the units used in the map.
Scale	Specifies the unit scale of the projected map, relative to meters. For example, a scale of one means one unit in the projected map equals one meter; a scale of two means two units in the projected map equal one meter; etc.

False Easting	Specifies the false easting, or horizontal offset, of the projected coordinates, in meters. *False Eastings* and *False Northings* are added to the underlying "projected" coordinates as a way to arbitrarily offset their internal XY coordinates after the projection. Unless you have a reason for using these offset values, do not use them. These values do not affect the latitude/longitude coordinates for the map, only the internal coordinates used to plot the map on the screen. If you use *False Easting* and *False Northing* offsets for a map, any subsequent boundaries you append to the map must also use these same offsets if you want the imported boundaries to be drawn in the correct relative position to the existing boundaries.
False Northing	Specifies the false northing, or vertical offset, of the projected coordinates, in meters. See above.
Central Scale Factor (KO)	Specifies the central scaling factor for the projection. This value is often set to 1.0, but may be set to another value for specific applications.
Central Longitude	Specifies the central longitude of the projection in degrees. The *Central Longitude* value typically should be defined as the longitudinal center of the map you are going to produce.
Central Latitude	Specifies the central latitude of the projection in degrees. The *Central Latitude* value typically should be defined as the latitudinal center of the map you are going to produce.

Universal Transverse Mercator (UTM) Projections



State of Oregon
Universal Transverse Mercator Projection
Central Latitude: 41.75
Central Longitude: -120.5

Projection Characteristics

The Universal Transverse Mercator system, commonly known as UTM, divides the Earth into sixty discrete zones, each representing a vertical slice of the globe spanning six degrees of longitude. A Transverse Mercator projection is applied to each zone with the central meridian of the projection at the center of the given zone and the central latitude of the projection at the equator. This coordinate system is the basis for many standardized regional maps, such as tract or neighborhood maps by the US Census Bureau and topographic quadrangles by the US Geological Survey. UTM is not generally used for coordinates outside the range of -80 to +84 degrees latitude due to the distortion inherent in Transverse Mercator projections near the poles.

Projection Parameters

Parameter	Description
Name	Specifies the units used in the map.
Scale	Specifies the unit scale of the projected map, relative to meters. For example, a scale of one means one unit in the projected map equals one meter; a scale of two means two units in the projected map equal one meter; etc.
Central Scale Factor (KO)	Specifies the central scaling factor for the projection. This value is often set to 1.0, but may be set to another value for specific applications.
Central Longitude	Specifies the central longitude of the projection in degrees. The *Central Longitude* value typically should be defined as the longitudinal center of the map you are going to produce. For example, the value -95.5 represents the geographic center of the United States, so the map of the U.S. is drawn upright.
Central Latitude	Specifies the central latitude of the projection in degrees. The *Central Latitude* value should be defined as the latitudinal center of the map you are going to produce.
Projection in Southern Hemisphere	Choose True if your map is in the southern hemisphere. Choose False if your map is in the northern hemisphere.

Van der Grinten Projection



World Map
Van der Grinten Projection
Central Longitude: 0

Projection Characteristics

This projection is typically used for maps of the world and the scale is accurate along the equator. This projection was used by the National Geographic for world maps. The central longitude and the equator are straight lines and the poles are greatly distorted.

To get good results with the Van der Grinten projection, the map must have coordinates between +/-180 degrees longitude and +/- 90 degrees latitude. **Surfer** does not wrap around +/-180 degrees longitude. The example map is using world-proj.gsb as the base map. If you are using world-scale maps, the central longitude generally cannot be set far from 0 degrees since there is no wrap around in **Surfer**.

Projection Parameters

Parameter	Description
Name	Specifies the units used in the map.
Scale	Specifies the unit scale of the projected map, relative to meters. For example, a scale of one means one unit in the projected map equals one meter; a scale of two means two units in the projected map equal one meter; etc.
False Easting	Specifies the false easting, or horizontal offset, of the projected coordinates, in meters. *False Eastings* and *False Northings* are added to the underlying "projected" coordinates as a way to arbitrarily offset their internal XY coordinates after the projection. Unless you have a reason for using these offset values, do not use them. These values do not affect the latitude/longitude coordinates for the map, only the internal coordinates used to plot the map on the screen. If you use *False Easting* and *False Northing* offsets for a map, any subsequent boundaries you append to the map must also use these same offsets if you want the imported boundaries to be drawn in the correct relative position to the existing boundaries.
False Northing	Specifies the false northing, or vertical offset, of the projected coordinates, in meters. See above.
Central Longitude	Specifies the central longitude of the projection in degrees. The *Central Longitude* value typically should be defined as the longitudinal center of the map you are going to produce.

Projection References

Dent, Borden D., *Cartography, Thematic Map Design*, Wm. C. Brown Publishers, Dubuque, 1990.

Greenhood, David, *Mapping*, The University of Chicago Press, Chicago, 1964, pp. 113-171.

Robinson, A.H., et al., *Elements of Cartography*, Fifth Edition, John Wiley & Sons, New York, 1984, pp. 75-105.

Snyder, John P., *Map Projections - A Working Manual*, U.S. Geological Survey Professional Paper 1395, Washington D.C., Department of the Interior, 1987.

Verhoogan, John, Francis., J. Turner, Lionel E. Weiss, Clyde Wahrhaftig, William S. Fyfe, *The Earth: An Introduction To Physical Geology,* Holt, Rinehart and Winston, Inc., New York, 1970.

Coordinate Systems Overview,
http://www.colorado.edu/geography/gcraft/notes/coordsys/coordsys_f.html, July 2, 2001.

Map Projection Overview,
http://www.colorado.edu/geography/gcraft/notes/mapproj/mapproj_f.html, July 2, 2001.

Map Projection Home Page, http://www.geography.hunter.cuny.edu/mp/, June 16, 2004.

Datums, Ellipsoids, Grids, and Grid Reference Systems, Defense Mapping Agency Technical Manual 8358.1, August 2, 2004.

Matching the Map Projection to the Need, http://www.gis.psu.edu/projection/, June 16, 2004.

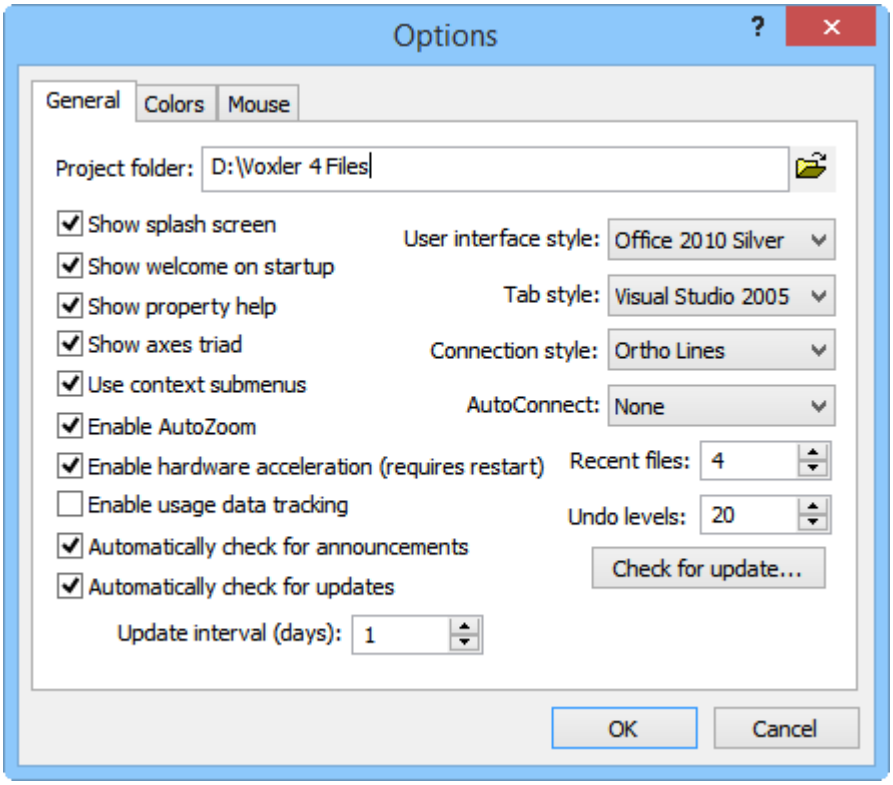
Chapter 19 - Program Options

Control **Voxler's** default settings by clickig the **Tools | Options** command. The **Options** dialog contains three tabs.

General	The *General* items are used to set basic window features such as file open/save paths and undo levels
Colors	The *Colors* items are used to set the background, axis, and text colors
Mouse	The *Mouse* items are used to set functions for the mouse buttons

General

Click the **Tools | Options** command to open the **Options** dialog. Use the *General* page of the **Options** dialog to set general preferences.



*Use the General page of the **Options** dialog to change general user settings.*

Project Folder

Enter a path name for the *Project folder*. The *Project folder* is the default path for opening and saving files. The *Project folder* is used as the starting path for the **Open**, **Save**, and **Save As** dialogs and the *Project files* in the **Welcome to Voxler** dialog.

Show Splash Screen

Check the box next to *Show splash screen* to display the **Voxler** splash screen when starting the program. Uncheck the box next to *Show splash screen* to hide the splash screen when starting the program.

Show Welcome on Startup

Check the box next to *Show welcome on startup* to display the **Welcome to Voxler** dialog immediately after starting **Voxler**.

Show Property Help

Check the box next to *Show property help* to display the help area at the bottom of the Property Manager that shows information about the selected property. Drag the horizontal border at the top of the help area to resize the help area if the text exceeds the current limit of the area.

Show Axes Triad

Check the box next to *Show axes triad* to turn on the world axis triad in the lower right corner of the **Viewer** window. Uncheck the box next to *Show axes triad* to hide the world axis triad. Axes triad options can be changed on the **Colors** page of the **Options** dialog.

Use Context Submenus

When a module in the **Network Manager** is right clicked, a context menu appears that contains all of the other modules that may be connected to the clicked module. Check the box next to *Use context submenus* to view these connectable modules in a hierarchy of submenus by type. Uncheck the box next to *Use context submenus* to view the connectable modules as a single list.

Enable AutoZoom

Check the box next to *Enable AutoZoom* to allow **Voxler** to automatically zoom the camera when the scene extents change by more than a fixed threshold. This is useful for keeping the camera adjusted properly so the geometry is always in view. It can sometimes result in unwanted zooming, e.g., when manipulating draggers. Uncheck the box next to *Enable AutoZoom* to disable the *AutoZoom* feature. With *AutoZoom* disabled, it may be necessary to make use of the **View | Fit to Window** option to reorient the camera.

Enable Hardware Acceleration

Check the box next to *Enable hardware acceleration (requires restart)* to improve display performance. Uncheck this box in order to help diagnose OpenGL driver issues. In general, this option should remain checked. **Voxler** must be closed and restarted anytime this option is changed.

Enable Usage Data Tracking

Check the box next to *Enable usage data tracking* to allow **Voxler** to send anonymous usage data to **Golden Software**. Sending usage data will help improve **Voxler** to meet our users' needs. This property does not have a default setting; it's initial setting is determined based on your selection to opt-in or opt-out of the Customer Experience Improvement Program during the **Voxler** install process.

Automatically Check for Announcements

An announcement is a message from Golden Software regarding changes made to **Voxler**. It is highly recommended that you have the box next to *Automatically check for announcements* checked. No information is transferred to Golden Software with this option.

Automatically Check for Updates

An update is a free newer version of the program. Updates include corrections to problems that have been found. Check the box next to *Automatically check for updates* to allow Golden Software to check for program updates according to the user defined *Update interval (days)*. Additionally, if a crash occurs while the program is running, a check for update will occur the next time the program is started.

When this box is checked, the program will check for an update immediately, and every two weeks thereafter. Adjust the update preferences at any time. Allowing automatic checks for updates allows you to have the most recent version of the program. It is highly advised that you have this option checked. No information is transferred to Golden Software with this option.

Update Interval (Days)

Specify the update time interval in days by entering a number in the *Update interval (days)* box. To change the update check interval, highlight the existing value and type a new value or click the  to increase or decrease the number of days. **Voxler** will automatically check for updates after the specified interval of time has passed.

Check for Update

Click the *Check for update* button to manually check for program updates (e.g. **Voxler** 4.0 to 4.1). Before using this command, make sure your computer is connected to the Internet. Follow the directions in the **Voxler Update** dialog to complete the update if an update is available. In addition to this location, program updates can be checked at any time with the **Help | Check For Update** command.

User Interface Style

Change the appearance of the user interface by selecting Office 2000, Office 2003, Office XP, Visual Studio 2005, Visual Studio 2008, Windows XP, Office 2010 Black, Office 2010 Blue, or Office 2010 Silver from the User interface style drop down menu. The title bar of each window, the toolbars, and other user interface elements varies in appearance depending on the selected user interface style. All User Interface selections provide a helpful docking mechanism feature that allows for easy docking of managers. The default option is Office 2010 Black.

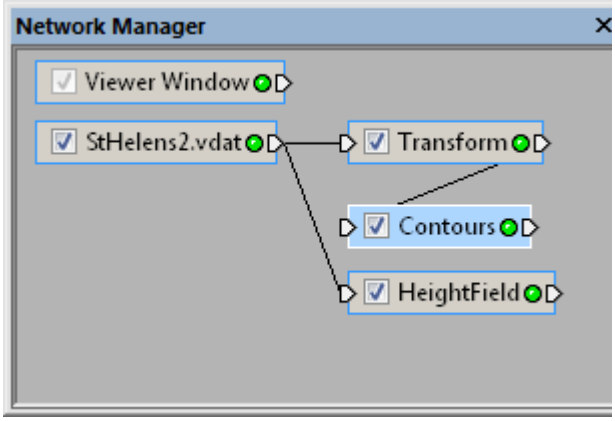
Tab Style

Specify the tabbed document view style by selecting the *Tab style.* Available options are *None, 3D, One Note,* and *Visual Studio 2005*. All visible **Viewer** windows have a tab appearance when this option is set to anything other than *None*.

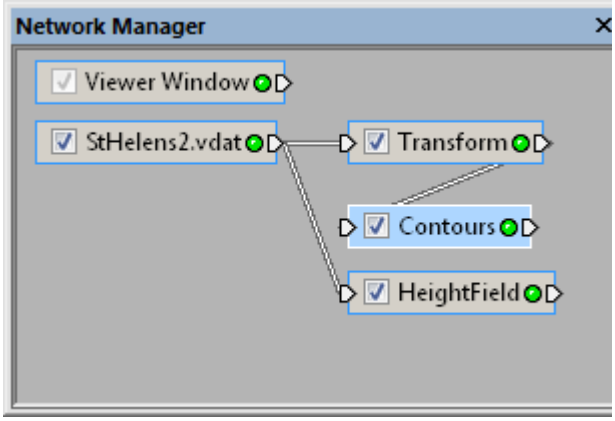
Connection Style

The *Connection style* is the style of the connecting lines between modules in the Network Manager. Select *Straight Lines, Straight Pipes, Ortho Lines,* or *Ortho Pipes* from the *Connection style* drop down list. *Straight Lines* and *Straight Pipes* are drawn as direct lines. These can go under modules,

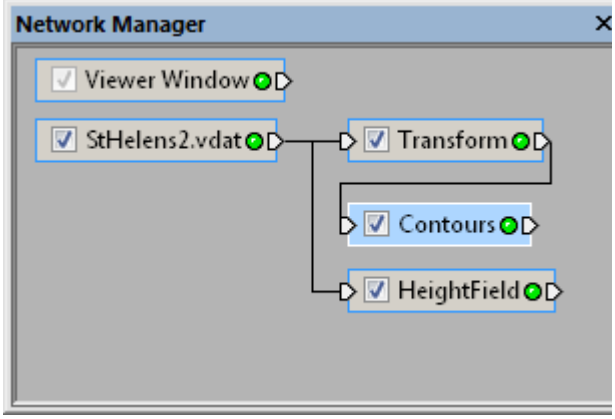
causing difficulty determining which modules are connected. *Ortho Lines* and *Ortho Pipes* are drawn as horizontal and vertical segments. Pipes are drawn as hollow "tubes."



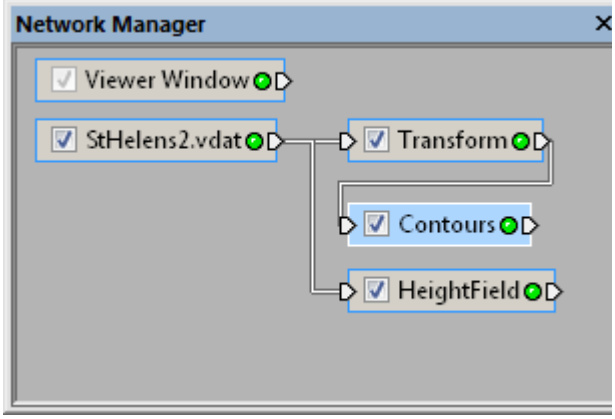
The Connection Style is set to Straight Lines in this example.



The Connection Style is set to Straight Pipes.



The Connection Style is set to Ortho Lines in this example.



The Connection Style is set to Ortho Pipes in this example.

AutoConnect

Choose *Bounding Box* or *From Input* from the *AutoConnect* drop down list to enable automatic connecting of items in the **Network Manager**. Choose *None* to disable automatic connecting. Choosing *From Input* will automatically connect an appropriate module when you import new data. Choosing *Bounding Box* will automatically add a bounding box to any newly imported data.

Recent Files

Choose the number of *Recent files* to display under the **File** menu and in the **Welcome to Voxler** dialog *Recent files* list. Select a value between *0* and *16*. Recent files are displayed in the **File** menu of the menu bar and can provide a quick way to access recently opened files. To change the number of recent files, highlight the existing value and type a new value or click the  to increase or decrease the number of files.

Undo Levels

Set the number of commands to undo in the *Undo levels* box. The maximum number of *Undo levels* is 100. Once the maximum number of actions has been performed, the oldest action is dropped off

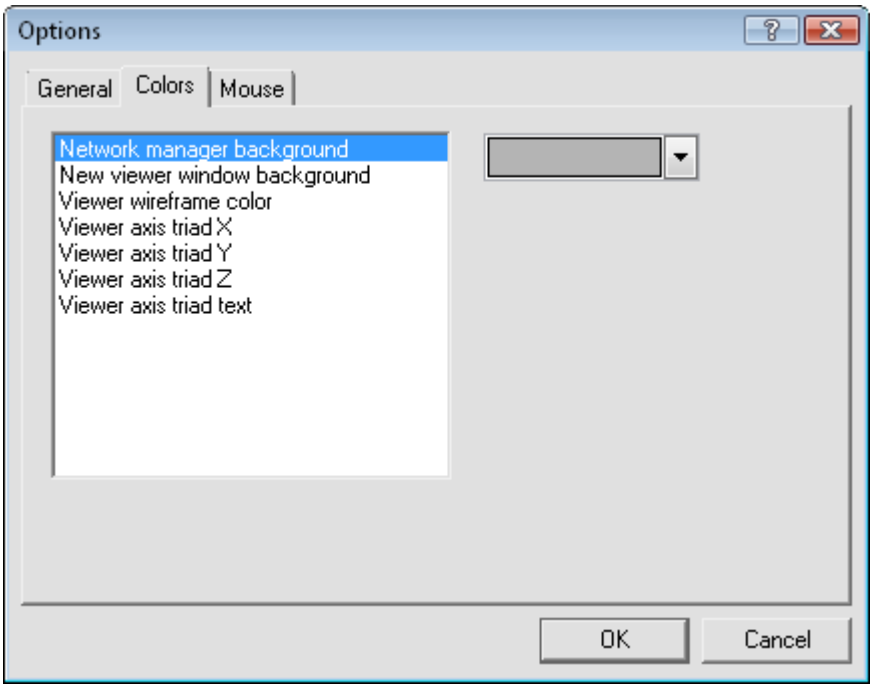
the list as new actions are added. Undo can consume significant amounts of memory, so this option should be left between 3 and 10 if memory is at a premium. Set the undo levels to 0 to disable *Undo levels* completely. To change the number of undo levels, highlight the existing value and type a new value or click the  to increase or decrease the number of undo levels.

OK or Cancel

Click *OK* to close the **Options** dialog and save your changes. Click *Cancel* to close the dialog without saving your changes.

Colors

Click the **Tools | Options** command to open the **Options** dialog. Use the *Colors* page of the **Options** dialog to specify the default colors used for various windows and user interface elements within **Voxler**.



*Use the **Colors** page of the **Options** dialog to change **Voxler** program colors.*

Program Area Selection

On the left side of the dialog, choose an area of the program to change. The options are Network manager background, New viewer window background, Viewer wireframe color, Viewer axis triad X, Viewer axis triad Y, Viewer axis triad Z, and Viewer axis triad text.

The *Network manager background* controls the background color for the **Network Manager**.

The *New viewer window background* controls the background color for a new **Viewer** window. The existing **Viewer** window does not change color. To change the color of the existing **Viewer** window, click on the *Viewer Window* module in the **Network Manager**.

The *Viewer axis triad* options control the world axis triad displayed in the **Viewer** window.

Color Selection

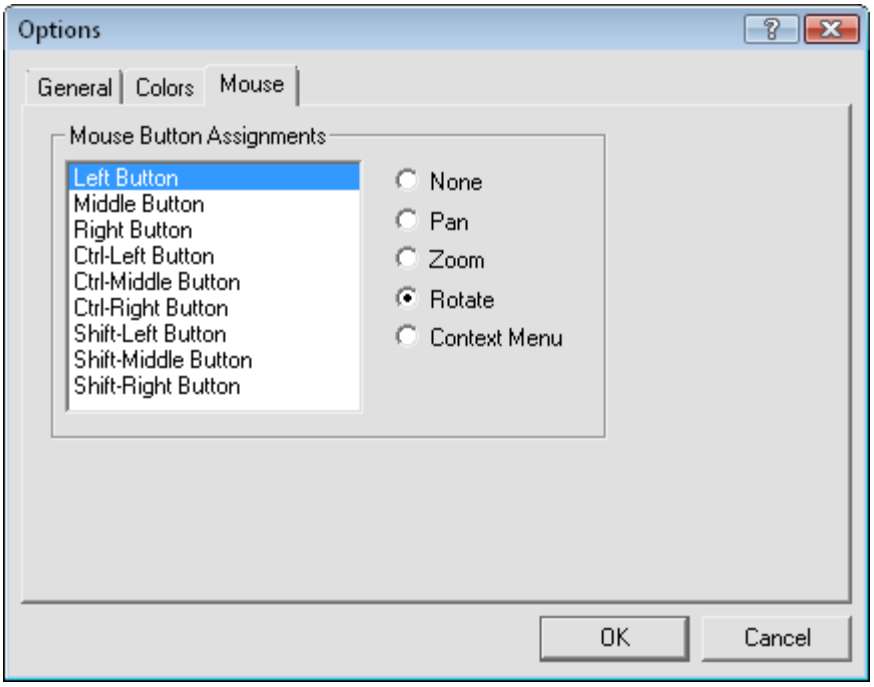
On the right side of the dialog, the current color appears in a drop down list. Click the color to open the color palette. Select the desired color for the option.

OK or Cancel

Click *OK* to close the **Options** dialog and save your changes. Click *Cancel* to close the dialog without saving your changes.

Mouse

Click the **Tools | Options** command to open the **Options** dialog. Use the *Mouse* page of the **Options** dialog to control which functions in the **Viewer** window are assigned to the mouse buttons.



*Use the **Mouse** page of the **Options** dialog to change the mouse control settings.*

Mouse Button Assignments

The buttons that can be used appear in the list at the left. As each button/key combination is selected, the assigned function displays on the right. The assigned function options are *None*, *Pan*, *Zoom*, *Rotate*, and *Context Menu.* Change the assigned function if you wish to customize the use of the mouse buttons.

OK or Cancel

Click *OK* to close the **Options** dialog and save your changes. Click *Cancel* to close the dialog without saving your changes.

Log Path

Voxler stores a log file that contains information about the process run during the last instance of **Voxler**. The **Voxler** log path is stored in the following locations:

Windows 8

C:\Users\<user>\AppData\Roaming\Golden Software\Voxler 4\

Windows 7

C:\Users\<user>\AppData\Roaming\Golden Software\Voxler 4\

Vista

C:\Users\<user>\AppData\Roaming\Golden Software\Voxler 4\

Windows XP

C:\Documents and Settings\<user>\Application Data\Golden Software\Voxler 4

Chapter 20 - Importing, Exporting, and Printing

The following file formats are supported for import and export from **Voxler**.

Abbreviations are listed below the chart. Click the green links to display the column header and abbreviation information on this page. Click the blue links to navigate to a new page with detailed information on the selected file format.

File Extension	File Format	Import	Import Data Type	Export Method	Export Data Type
?10g	GLOBE DEM	2	U	-	-
ACCDB	Access Database	2,6	P	-	-
ADF	Esri Arc/Info Binary Grid	2	U	3b 3d	U
AM, COL	Amira Mesh	2	U3, R3, C3, P	3a 3b 3d	U3, R3, C3
AN1, AN2	ACR-NEMA Medical Image	2	U2, U3, B, I	-	-
ASC AIG, AGR, GRD	Esri ArcInfo ASCII Grid	2	U	3b 3d	U
ASC, DAT, GRD, CPS, CPS3	CPS-3 Grid	2	U	3b 3d	U
ASC, DAT, GRD, XYZ, ZMAP, ZYC, ZYCOR	Z-MAP Plus Grid	2	U	3b 3d	U
ASI	Amira Stacked Image	2	U3, R3	-	-
BIL, BIP, BSQ	Banded Lattice	2	U	3b 3d	U

BLN	Golden Software Blanking	2, 6	V2, V3 S	7	P
BMP	Windows Bitmap	2	B, I	3d 4	B, I
BNA	Atlas Boundary	2, 6	P	3a, 3b, 3d, 7	P
CSV	Data	2, 6	P	3a, 7	P
DAT	Golden Software Data	2, 6	P	3a, 3b, 7	P
DAT	XYZ Grid	-	-	3d	U
DBF	DBF Database	2, 6	P	-	-
DDF, TAR, TAR.GZ, ZIP, TGZ	SDTS Raster (DEM)	2	U2	-	-
DDF, TAR, TAR.GZ, ZIP, TGZ	SDTS Vector (TVP)	2	V2, S	-	-
DEM	USGS Digital Elevation Model	2	U2	3b 3d	N
DIC, DCM, AN?	DICOM3 Medical Image	2	U2, U3, B, I	-	-
DLG, LGO, LGS	USGS Digital Line Graph	2	V2, S	-	-
DOS, DAT	USGS ETOPO5	2	U	-	-
DT0, DT1, DT2, DT?	DTED	2	U2	-	-
DXF	AutoCAD DXF Drawing	2	V3, S	3c, 3d	V3, S
E00	Esri ArcInfo Export	2	V2, S	-	-
E00	Esri Grid File	2	U2	-	-
ECW	ER Mapper	2	B, I	-	-

EPS	Encapsulated Postscript	-	-	3d 4	B, I
ERS	ER Mapper Grid	2	U	3b 3d	U
FLD	AVS Field	2	U3, R3, C3	3a 3b 3d	U3, R3, C3
FLT	Esri Float Grid	2	U	3b 3d	U
GIF	GIF Image	2	B, I	3d 4	B, I
GRB, GRIB	GRIB Global Weather Data Grid	2	U2	-	-
GRD	Surfer Grid	2	U2	3b 3d	N
GRD, GGF	Geosoft Binary Grid	2	U	3b 3d	U
GSB	Golden Software Boundary	2	V2, S	-	-
GSI	Golden Software Interchange	2	V2, S	-	-
GXF	Grid eXchange	2	U	3b 3d	U
HDF	Hierarchical Data Format	2	U3, R3, C3	3a 3b 3d	U3, R3, C3
HDR	GTOPO-30	2	U2	-	-
HDR (data set), IMG (image)	Analyze 7.5 Medical Image	2	U3	3b 3d	U3
HGT, ZIP	NASA SRTM Grid Data	2	U2	-	-
IMG	ERDAS Imagine Image	2	U2, I	-	-
INFO	Leica Confocal Raw Slices	2	U3	-	-
IV	Inventor, SGI Open Inventor	2	G	3c 4	G

JPG, JPEG	JPEG Compressed Bitmap	2	B, I	3d 4	B, I
JP2, J2K, JPC, JPT, JPEG2000, J2000	JPEG-2000 File Interchange	2	B, I	3d 4	B, I
KML, KMZ	Google Earth Keyhole Markup	2	V2, P	-	-
LAS	LAS LiDAR Binary File	2, 6	P	-	-
LAS	LAS Well Data	2	P	-	-
LAT	Iris Explorer	2	U3, R3, C3	3a 3b 3d	U3, R3, C3
MDB	Microsoft Access	2, 6	P	-	-
MIF	MapInfo Interchange	2	V2, S	-	-
NC	NetCDF Lattice	2	U	3b 3d	U
P3D, XYZ	PLOT-3D	2	C3, P	3a 3b 3d	C3
PCX	ZSoft Paintbrush Bitmap	2	B, I	3d 4	B, I
PDF	PDF (Vector)	2	G, I	-	-
PDF	PDF (Raster)	2	G, I	3d 4	G, I
PLT	Golden Software PlotCall	2	V2	-	-
PLY	Stanford Polygon	2	V3, G	-	-
PNG	Portable Network Graphic	2	B, I	3d 4	B

PNM, PPM, PGM, PBM	Portable Bitmap	2	B, I	3d 4	B, I
RAS, SUN	Sun Raster	2	B, I	3d 4	B, I
RAW, BIN	Raw Binary Lattice	2	U3	3b 3d	U3
RGB, RGBA, BW	Silicon Graphics (SGI) RGB Image	2	B, I	3d 4	B, I
RST, IMG	Idrisi Raster	2	U	-	-
SEG, SP1	SEG-SP1 Data File Format	2, 6	P	-	-
SGY, SEGY	SEG-Y Seismic Data	2	U2, V2	-	-
SHP, TAR.GZ, TAR, ZIP, TGZ	Esri Shapefile	2	V3, S	-	-
SID	LizardTech MrSID Image	2	B, I	-	-
SLK	Sylk Spreadsheet	2, 6	P	3a, 7	P
STK	Metamorph STK	2	U2 U3	-	-
TAR	SDTS Raster (DEM)	2	U2	-	-
TAR	SDTS Vector (TVP)	2	V2, S	-	-
TAR.GZ	SDTS Raster (DEM)	2	U2	-	-
TAR.GZ	SDTS Vector (TVP)	2	V2, S	-	-
TGA	TrueVision Targa	2	B, I	3d 4	B, I
TGZ	SDTS Raster (DEM)	2	U2	-	-
TGZ	SDTS Vector (TVP)	2	V2, S	-	-

TIF, TIFF	Tagged Image File Format	2	U2, U3, B, I	3d 4	B, I
TXT	Text Data	2, 6	P	3a, 7	P
VDAT	Voxler Data	2	U2, R2, C2, U3, R3, C3, P	3a 3b 3d	U2, R2, C2, U3, R3, C3, P
VOXB	Voxler Network File	6	-	5	-
VTK	Visualization Toolkit	2	U3, R3, C3, I	3a 3b 3d	U3, R3, C3, I
X, XIMG	AVS X-Image	2	B, I	3d 4	B, I
XLS	Excel Spreadsheet	2, 6	P	3a, 7	P
XLSM	Excel 2007 Macro-Enabled Spreadsheet	2, 6	P	-	-
XLSX	Excel 2007 Spreadsheet	2, 6	P	3a, 7	P
ZIP	SDTS Raster (DEM)	2	U2	-	-
ZIP	SDTS Vector (TVP)	2	V2, S	-	-
File Extension	File Format	Import Method	Formats Supported for Import[1]	Export Method	Formats Supported for Export[1]

1 The table uses the following format abbreviations:

U = uniform lattice

U2 = 2D uniform lattice

U3 = 3D uniform lattice

R = rectilinear lattice

R2 = 2D rectilinear lattice

R3 = 3D rectilinear lattice

C = curvilinear lattice

C2 = 2D curvilinear lattice

C3 = 3D curvilinear lattice

N = Lattice has one component per node and only one slice

I = images (planar uniform lattice, 1 to 4 color components)

V2 = 2D Vector Graphics

V3 = 3D Vector Graphics
P = point data
G = geometry or scene data (iv)
W = well data or log data
- = not supported

2 Choose the **File | Import** command.
3a Select a data source module and choose the **File | Save Data** command.
3b Select a lattice source module (i.e. *Gridder*) and choose the **File | Save Data** command
3c Select an IV data module and choose the **File | Save Data** command
3d Select an image module and choose the **File | Save Data** command
4 Choose the **File | Export** command.
5 Choose the **File | Save** command.
6 Choose the **File | Open** command.
7 Choose the **File | Save As** command in the worksheet.

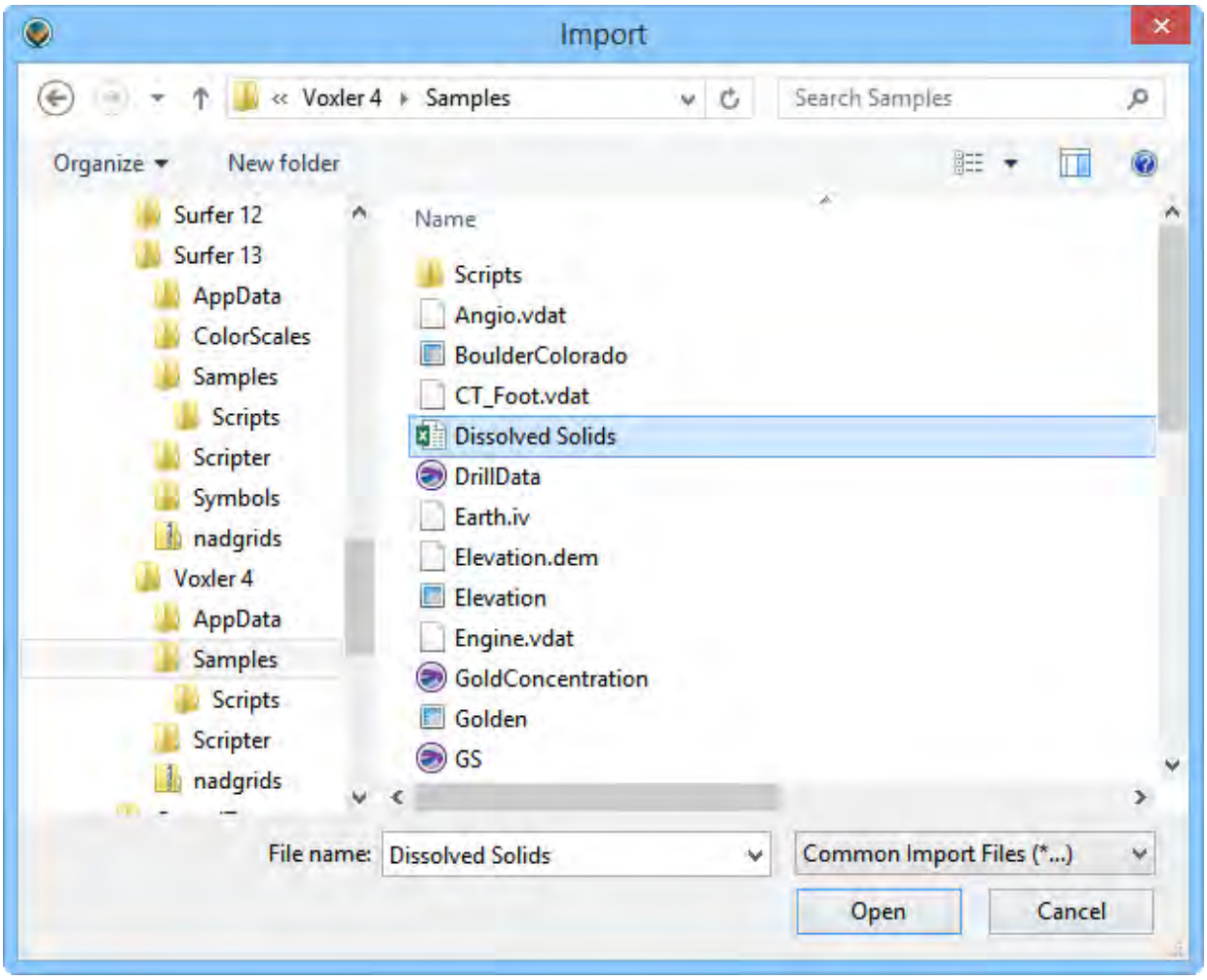
Import

Click the **File | Import** command, click the  button, right-click in the **Network Manager** and choose **Import**, or double-click *Import* in the **Module Manager** to load a data file into a data source module. Depending on the format of the data, one or more format-specific dialogs may appear. The loaded data displays as a new icon in the **Network Manager**. It can then be connected to other modules in the network.

When this command is selected, the **Import** dialog displays, allowing one or more files to be selected and loaded. Each data set from each file is loaded into its own source module. Multiple data files can be loaded at once by pressing the SHIFT or CTRL keys while selecting files in the dialog.

The Import Dialog

The **File | Import** command opens the **Import** dialog.



*Specify files to import using the **Import** dialog.*

The **Import** dialog has the following options available:

Look In

The *Look in* field shows the current directory. Click the down arrow to see the directory structure; click the folders to change directories.

Creating New Folders and Changing the View

The buttons to the right of the *Look in* field allow you to create new folders and change the view of the file list.

File List

The *File list* displays files in the current directory, which is listed in the *Look in* field. The *File of type* field controls the display of the file list. For example, if *Voxler Project Files (*.voxb)* is listed in the *Files of type* field, only *.VOXB files appear in the files list.

File Name

The *File name* field shows the name of the selected file. Type a path and file name into the box to open a specific file.

Specify a File Type

The *Files of type* field controls the display of the file list. For example, if *Voxler Project Files (*.VOXB)* is listed in the *Files of type* field, only .VOXB files appear in the file list. To see all files in the directory, choose *All Files (*.*)* from the *Files of type* list. Double-click a file to open it, or click the file once and then click the *Open* button.

The *Files of type* field shows the file format to be imported. To change the file format, click the down arrow and select the file type from the list. *All Files (*.*)* displays all files in a directory.

Open or Cancel

Click *Open* to close the **Open** dialog and import the specified file. Click *Cancel* to close the dialog without opening a new file.

Data Import Options Dialog

The **Data Import Options** dialog appears only once, even if more than one file is selected. Any changes made in this dialog are automatically applied to all files. If you need to specify different import options for one or more files, you must load them individually.

File Formats

Voxler automatically attempts to determine the selected file formats. The format of the imported file(s) is determined as follows:

1. If a particular format is specified in the *Files of type* list of the **Import** dialog, that format is used.
2. If that fails, then a match is made by using the file extension.
3. If the format still cannot be determined, the **Select Format** dialog displays and you are prompted to select the format from a list of supported formats.

Data Types

Voxler supports several different types of data:

- Tabular data— such as worksheets, comma-separated variable (CSV), and data files— are loaded into *Data Source* modules. The data must be organized by columns. For example, one column corresponds to the X coordinate, another to the Y coordinate, etc. You can then specify if the data file contains point or well data in the **Property Manager**. The columns that contain the X, Y, and Z coordinates along with the columns containing the data at those coordinates are all also specified in the **Property Manager**.
- Log ASCII Standard .LAS files are loaded into a *WellData* module. Each of the variables in the .LAS file are imported into **Voxler** as a log component.
- Data files that are not in a tabular file format are loaded into a *point source* module.
- Images and bitmaps are loaded as two-dimensional lattices. Grayscale images are converted to 1-byte scalar lattices. Color images are converted to four- component RGBA lattices.
- Native three-dimensional lattice data in various proprietary and public formats are supported, including .RAW format.
- One-dimensional curvilinear lattices are automatically converted to point sets during import.

- Vector formats.

Loading Multiple Images

Image stacks are loaded as a three-dimensional lattice. These are usually a series of planar image slices that are meant to be stacked one on top of another to form a three-dimensional volume. The individual slices are often in a standard two-dimensional format like .TIFF, .JPEG, or .DICOM. This type of data is commonly created by medical imaging equipment.

If all of the images are selected at once and the sizes are identical, **Voxler** attempts to combine them into a single three-dimensional lattice. The slices are combined in alphabetical order according to the file names. The stacking starts at the bottom (Z=1) and proceeds upward. Simultaneous selection is most easily achieved by first clicking the first slice and then holding down the SHIFT key and clicking the last slice. The resulting source module takes the name of the first loaded image. The name of the last file is listed next to *File Path* in the **Property Manager**.

Image Coordinates

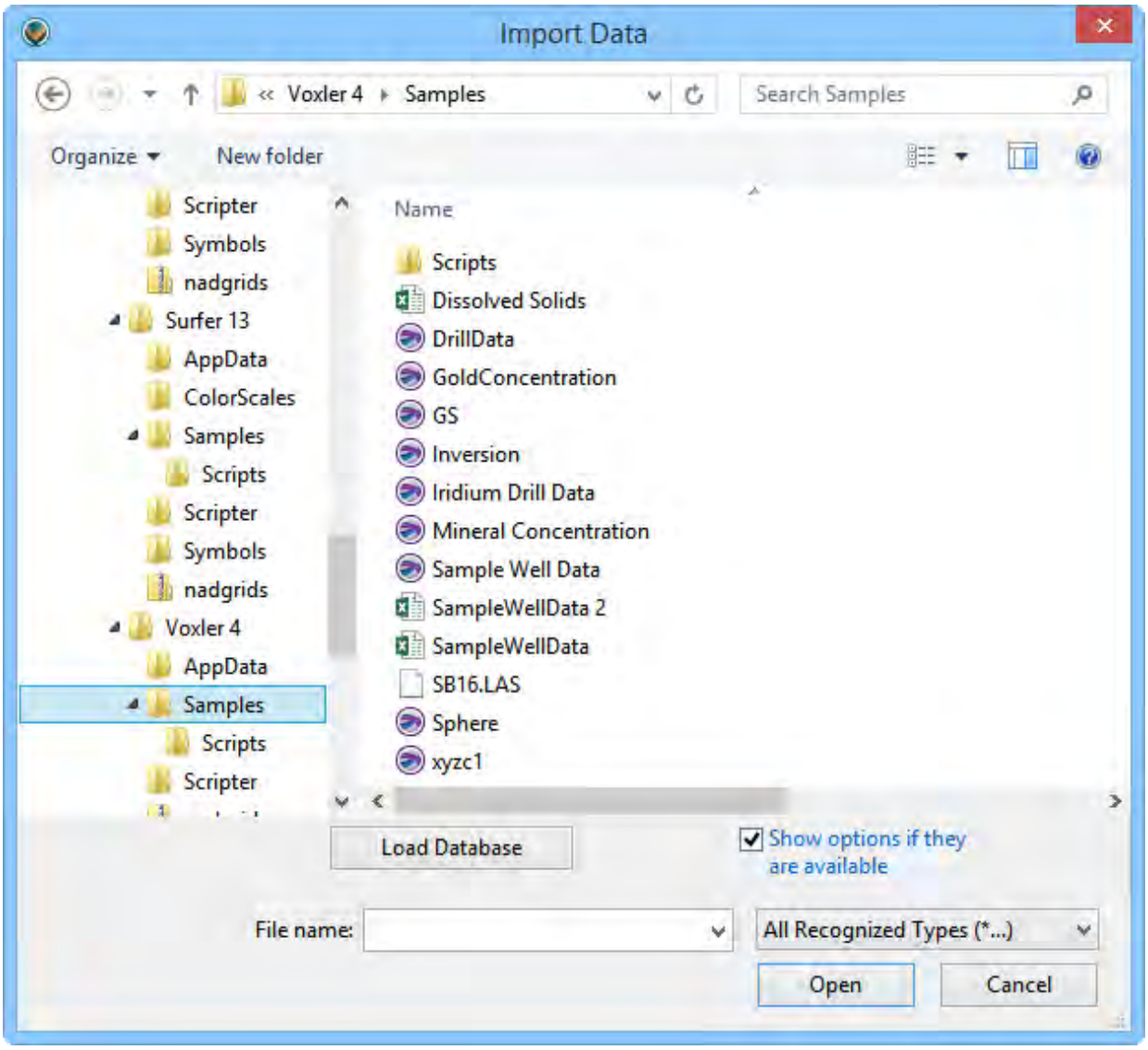
When loading image bitmap files with coordinates, the coordinate information comes from a geotiff (internal information) or external files.

Import - Worksheet Document

Clicking the **File | Import** command in the worksheet loads the contents of a data file into the existing worksheet. Select a file to import into the existing file in the **Import Data** dialog. The contents of the new file are merged into the worksheet at the active cell so be sure to position the cell at the edge of the existing data. Any cells in the existing worksheet that lie to the right of and below the active cell are overwritten with the contents of the merging file.

Import Data Dialog

The **File | Import** command in the worksheet opens the **Import Data** dialog.



*Specify files to import into the **Surfer** worksheet using the **Import Data** dialog.*

Look In

The *Look in* field shows the current directory. Click the down arrow to see the directory structure and click on the folders to change directories.

Creating New Folders and Changing the View

The buttons to the right of the *Look in* field allow you to create new folders and change the view of the file list.

File List

The *File list* displays files in the current directory. The current directory is listed in the *Look in* field. The *Files of type* field controls the display of the file list. For example, if *DAT Data (*.dat)* is listed in the *Files of type* field only *.DAT files appear in the files list.

Specify a File Name

The *File name* field shows the name of the selected file. Alternatively, type a path and file name into the box to open a file.

Files of Types

The *Files of type* field controls the display of the file list. For example, if *DAT Data (*.dat)* is listed in the *Files of type* field only *.DAT files appear in the files list.

The *All Recognized Types (*...)* format type is selected by default. This displays all the common file formats in the navigation pane. If a different format type is selected, **Voxler** will remember the setting until the end of the current session. When **Voxler** is restarted, the default format type will be used.

To see all files in the directory, choose *All Files (*.*)* from the *Files of type* list. Double-click on a file to open it or single-click the file and then click the *Open* button. The *All Files (*.*)* option shows all of the file formats in the current directory, even if the file type is not appropriate for the action chosen. For example, a .GRD file may be displayed, even though a .GRD file cannot be imported into the worksheet.

Select a file type from the *Files of type* list. The following formats are supported:

- ACCDB Microsoft Access .ACCDB
- CSV Comma Separated Variables .CSV
- DAT Data .DAT
- DBF Database .DBF
- LAS LiDAR Data .LAS
- MDB Microsoft Access 97-2003 Database .MDB
- P1 Data Exchange Format .SP1
- PLOT-3D .P3D .XYZ
- SEG-P1 Data Exchange Format .SEG
- SLK Sylk Spreadsheet .SLK
- TXT Text Data .TXT
- XLS Excel Spreadsheet .XLS
- XLSX Excel 2007 Spreadsheet .XLSX
- XLSM Excel 2007 Spreadsheet .XLSM

Load Database

Click the *Load Database* button in the **Import Data** dialog to open the **Data Link Properties** dialog and import a database.

Show Options If They Are Available

If *Show options if they are available* is checked, then opening .TXT files or ASCII text files with unsupported file extensions will bring up the **Data Import Options** dialog where you can specify the import options.

Data Link Properties

You can open database files in **Voxler** by clicking *Load Database* in the **Import Data** dialog. The **Data Link Properties** dialog will open. Imported databases appear in a new worksheet window. Once the worksheet is saved, the link to the database is removed.

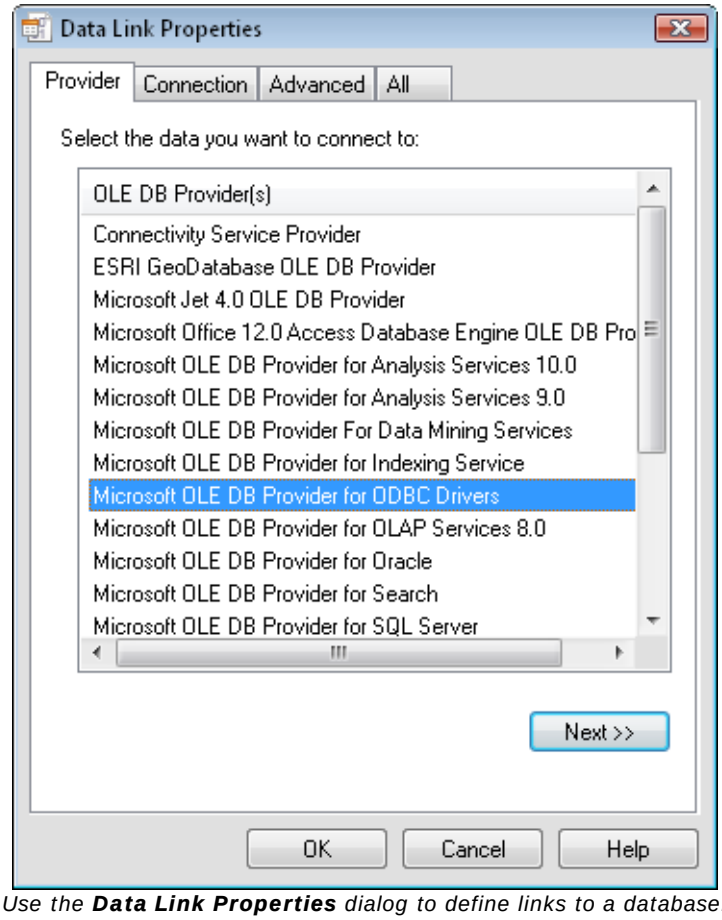
The data linking provides a method to link virtually any database supported by Microsoft via an OLE DB Provider, ODBC, or some other supported format. Because the data link provides access to many types of databases that vary by computer, and since this link is provided by Microsoft, only general information is provided here. Click *Help* in the **Data Link Properties** dialog to access Microsoft's help file for this dialog. If you need additional information on specific connection information, contact your network administrator.

ODBC Drivers

Note that ODBC drivers are installed by the database program, not by **Voxler**. If a database program is not installed on the computer, a driver may still be able to be located to import the data from the database. For example, the Microsoft Access database engine can be downloaded from Microsoft's website.

The Data Link Properties Dialog

Data linking is used to define links to many types of databases. The **Data Link Properties** dialog is accessed by clicking the *Load Database* button in the **Import Data** dialog. The **Import Data** dialog appears when you use the worksheet **File | Import** command.



*Use the **Data Link Properties** dialog to define links to a database.*

Help Button

Click the *Help* button in the **Data Link Properties** dialog to access Microsoft's help file for this dialog. If you need additional information on specific connection information, contact your network administrator.

Data Linking

To use data linking,

1. Click the **File | Import** command from the worksheet window.
2. Click *Load Database* in the **Import Data** dialog.
3. The **Data Link Properties** dialog opens, with the *Provider* tab active.
4. Step through the dialogs to import the database data.

Provider Tab

Use the **Provider** tab to select the appropriate OLE DB provider for the type of data you want to access. Not all applications allow you to specify a provider or modify the current selection; this tab is displayed only if your application allows the OLE DB provider selection to be edited. You can save a data link with the application itself or as a separate file. For more information about creating a data link, refer to the Microsoft help.

Option	Description
OLE DB Provider(s)	Lists all OLE DB providers detected on your computer. For more information about providers, see "Microsoft OLE DB Providers Overview" in the MDAC SDK.
Next	Opens the Connection tab for the selected OLE DB provider.

Note: You can navigate directly to the **Connection** tab by double-clicking the desired provider. For more information about Data Links, see the *Data Link API Reference*.

Click **Next** or click the **Connection** tab to specify how to connect to ODBC data.

Connection Tab

This **Connection** tab is provider-specific and displays only the connection properties that are required by the Microsoft OLE DB Provider for ODBC.

Option	Description
Use data source name	Select from the list, or type the ODBC data source name (DSN) you want to access. You can add more sources through the ODBC Data Source Administrator. Refresh the list by clicking Refresh.
Use connection string	Allows you to type or build an ODBC connection string instead of using an existing DSN.
Build	Opens the Select Data Source dialog box. Once you select a data source the connection string in that data source will appear in the Connection tab of the Data Link Properties dialog.

	If you select a File DSN, the resulting ODBC connection string is not based on a DSN. The ODBC connection string is persisted in the data link (.udl) file and does not rely on the selected file DSN. If you select a Machine DSN, the resulting ODBC connection string is based on a DSN. The ODBC connection string references the selected machine DSN. If a user on a different system attempts to access the data link (.udl) file, the user must also have the machine DSN installed.
User name	Type the User ID to use for authentication when you log on to the data source.
Password	Type the password to use for authentication when you log on to the data source.
Blank password	Enables the specified provider to return a blank password in the connection string.
Allow saving password	Allows the password to be saved with the connection string. Whether the password is included in the connection string depends on the functionality of the calling application. Note If saved, the password is returned and saved unmasked and unencrypted.
Enter the initial catalog to use	Type in the name of the catalog (or database), or select from the drop-down list.
Test Connection	Click to attempt a connection to the specified data source. If the connection fails, ensure that the settings are correct. For example, spelling errors and case sensitivity can cause failed connections.

Note For more Data Link connection information, see the *Data Link API Reference*.

Advanced Tab

Use the **Advanced** tab to view and set other initialization properties for your data.

The **Advanced** tab of the **Data Link Properties** dialog box is provider-specific and displays only the initialization properties required by the selected OLE DB provider. For more information about advanced initialization properties, see the documentation provided with each specific OLE DB provider.

The following table describes most initialization options.

Option	Description
Impersonation level	The level of impersonation that the server is allowed to use when impersonating the client. This property applies only to network connections other than Remote Procedure Call (RPC) connections; these impersonation levels are similar to those provided by RPCs. The values of this property correspond directly to the levels of impersonation that can be specified for authenticated RPC connections, but can be applied to connections other than authenticated RPCs. Select from the following levels: • Anonymous—The client is anonymous to the server. The server process cannot obtain identification information about the client and cannot impersonate the client. • Delegate—The process can impersonate the client's security context while acting on behalf of the client. The server process can also make outgoing calls to other servers while acting on behalf of the client. • Identify—The server can obtain the client's identity. The server can impersonate the client for Access Control List (ACL) checking but cannot access system objects as the client. • Impersonate—The server process can impersonate the client's security context while acting on behalf of the client. This information is obtained when the connection is established, not on every call.

Protection level	The level of data protection sent between client and server. This property applies only to network connections other than RPC connections; these protection levels are similar to those provided by RPCs. The values of this property correspond directly to the levels of protection that can be specified for authenticated RPC connections, but can be applied to connections other than authenticated RPCs. Select from the following levels: • Call—Authenticates the source of the data at the beginning of each request from the client to the server. • Connect—Authenticates only when the client establishes the connection with the server. • None—Performs no authentication of data sent to the server. • Pkt—Authenticates that all data received is from the client. • Pkt Integrity—Authenticates that all data received is from the client and that it has not been changed in transit. • Pkt Privacy—Authenticates that all data received is from the client, that it has not been changed in transit, and protects the privacy of the data by encrypting it.
Connect timeout	Specifies the amount of time (in seconds) that the OLE DB provider waits for initialization to complete. If initialization times out, an error is returned and the connection is not created.
Access permissions	Select one or more of the following permissions: • **Read** - Read only. • **ReadWrite** - Read and write. • **Share Deny None** - Neither read nor write access can be denied to others. • **Share Deny Read** - Prevents others from opening in read mode. • **Share Deny Write** - Prevents others from opening in write mode. • **Share Exclusive** - Prevents others from opening in read/write mode. • **Write** - Write only.

Note For more Data Link initialization information, see the *Data Link API Reference*.

All Tab

Use the **All** tab to view and edit all OLE DB initialization properties available for your OLE DB provider. Properties can vary depending on the OLE DB provider you are using. For more information about the initialization properties, refer to the documentation provided with each specific OLE DB provider.

Option	Description
Initialization properties list	Lists all properties and their current values.
Edit Value	Opens the Edit Property Value dialog box for the selected property.

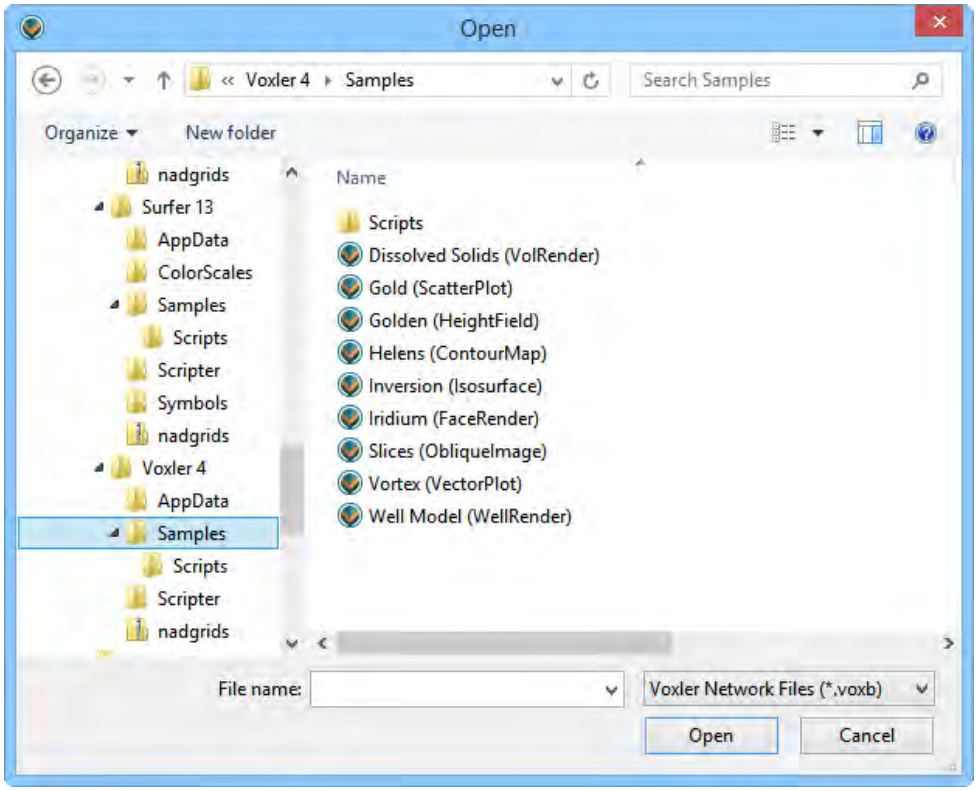
Note: You can open the **Edit Property Value** dialog box by double-clicking the desired property. For more Data Link initialization information, see the *Data Link API Reference*.

Open

Click the **File | Open** command or the  button to open a previously saved **Voxler** project or data file. You are prompted for a file name. Project files use the .VOXB extension.

The Open Dialog

Click the **File | Open** command to open the **Open** dialog.



*The **Open** dialog is displayed, allowing you to select the .VOXB file to open.*

The following options are available in the **Open** dialog:

Look In

The *Look in* field shows the current directory. Click the down arrow to see the directory structure; click the folders to change directories.

Creating New Folders and Changing the View

The buttons to the right of the *Look in* field allow you to create new folders and change the view of the file list.

File List

The *File list* displays files in the current directory, which is listed in the *Look in* field. The *File of type* field controls the display of the file list. For example, if *Voxler Network Files (*.voxb)* is listed in the *Files of type* field, only *.VOXB files appear in the files list.

File Name

The *File name* field shows the name of the selected file. Type a path and file name into the box to open a specific file.

Specify a File Type

The *Files of type* field controls the display of the file list. For example, if *Voxler Network Files (*.VOXB)* is listed in the *Files of type* field, only .VOXB files appear in the file list. To see all files in the directory, choose *All Files (*.*)* from the *Files of type* list. Double-click a file to open it, or click the file once and then click the *Open* button.

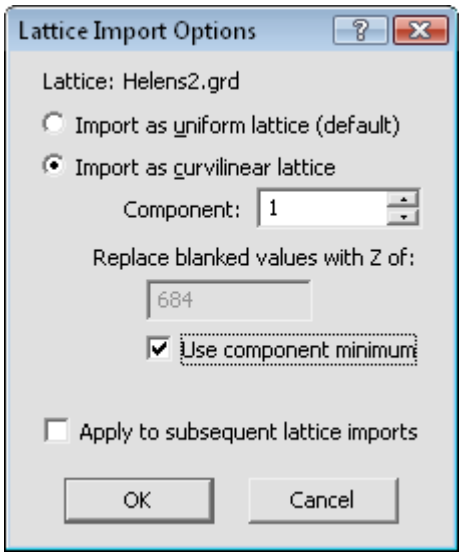
The *Files of type* field shows the file format to be opened. To change the file format, click the down arrow and select the file type from the list. *All Files (*.*)* displays all files in a directory. The *Common Document Files (*...)* selection shows all files in the directory that match one of the file types in the *Files of type* list.

Open or Cancel

Click *Open* to close the **Open** dialog and open the specified **Voxler** .VOXB or data file. Click *Cancel* to close the dialog without opening a new file.

Lattice Import Options Dialog

Load a lattice into **Voxler** using the **File | Import** command and the **Lattice Import Options** dialog is displayed.



Specify import options in the ***Lattice Import Options*** *dialog.*

Lattice Name

The lattice file being imported into **Voxler** is displayed next to *Lattice*. The displayed lattice name is read-only. The supported file formats that display this dialog include:

- ADF Arc/Info Binary Grid
- BIL Band Interleaved .BIL, .BIP, .BSQ
- CPS-3 Grid .ADX, .DAT, .GRD, .CPS
- DDF SDTS DEM .DDF
- DEM USGS Digital Elevation Model .DEM
- DOS USGS ETOPO5 .DOS, .DAT
- ERS ER Mapper Grid .ERS
- FLT Esri Binary Float Grid .FLT, .HDR
- GRD Surfer Grid .GRD
- GXF Grid Exchange .GXF
- HDF Hierarchical Data Format .HDF
- HGT NASA SRTM Data Format .HGT
- IMG ERDAS Imagine File Format .IMG
- RAW Binary Lattice .RAW, .BIN
- RST Idrisi Raster Image .RST, .IMG
- SEG-Y Seismic Data Log .SGY, .SEGY
- Z-MAP Plus Grid .ASC, .DAT, .GRD, .XYZ, .ZMAP, .ZYC, .ZYCOR

Import as Uniform Lattice

Select *Import as uniform lattice (default)* to import the grid file with the Z values as zeros for the entire grid. The grid values are imported as component information. This method imports all Z values data as zero. This means that some graphical output modules cannot be attached to the grid.

Import as Curvilinear Lattice

Select *Import as curvilinear lattice* to import the grid with a Z value for each grid node. Specify the Z value using the *Component* option. This method imports the component information as the Z data. For blanked values, a value can be specified or the component minimum value can be used.

Component

When the *Import as curvilinear lattice* is selected, specify the Z value for each grid node in the *Component* box. To change the component, highlight the existing value and type a new value in the box or click the  button to increase or decrease the *Component* value. The number is controlled by the number of components in the grid file.

Blanked Z Values

When the original grid has a blank value, the program must use a new number in the curvilinear lattice. The *Replace blanked values with Z of* can be specified for the blanking value. To set the blank value to any numeric value, highlight the existing number and type a new value.

Use Component Minimum

Check the box next to the *Use component minimum* option to use the minimum component value when a blank value in the grid needs to be filled. This option is useful when using the *Transform* module on a *HeightField*. For example, when a grid file has X and Y scales that differ from the Z scale and a *Transform* module is added between the data and the *HeightField*.

Apply to Subsequent Uniform Lattices

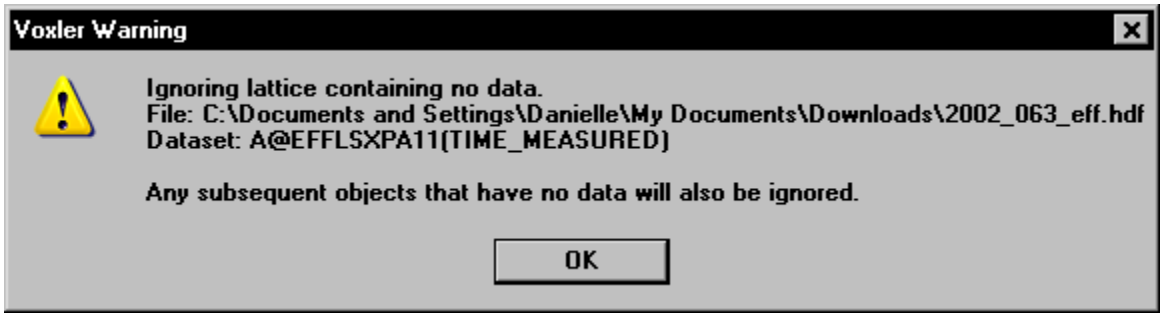
Check the *Apply to subsequent lattice imports* box to apply the selections to future imported lattices.

OK or Cancel

Click the *OK* button to close the dialog and import the lattice file using the specified options. Click the *Cancel* button to close the dialog without saving changes.

Voxler Warning

If you load a file with multiple 2D slices, choose to import as a uniform lattice, and the *Apply to subsequent lattice imports* box is checked, a warning is issued. The warning is a reminder that **Voxler** is ignoring a lattice that contains no data and any subsequent objects that have no data will also be ignored. Click *OK* to accept the warning and continue.

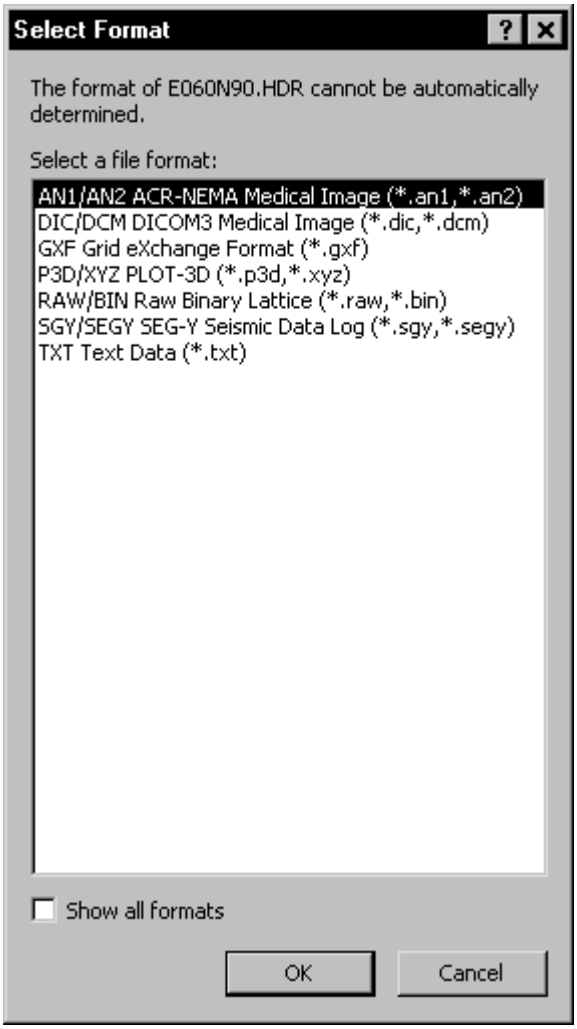


A warning message is issued to confirm that subsequent objects with no data will also be ignored.

Select Format Dialog

The **Select Format** dialog displays when **Voxler** cannot determine the format of a file the user is attempting to import.

This dialog appears when the **File | Import** command is chosen and a non recognizable file format is selected.



*Select a file format from the list in the **Select Format** dialog to specify a file type for **Voxler** to open.*

Show All Formats

Check the *Show all formats* box to show all possible file format choices.

OK

Select the appropriate file format and click the *OK* button to save your changes and the dialog closes.

Cancel

Click the *Cancel* button to close the dialog without proceeding with the import process.

Save

Click the **File | Save** command or click the  button to save the current project as a Voxler .VOXB file or data file in one of the data file formats. Project files use the .VOXB extension. If the project or data file has not previously been saved, the **Save As** dialog is displayed. The saved project contains all modules, connections, and source data. Intermediate data are recalculated when the project is reopened.

If you would like to save an existing file to a new file name or change the directory, click the **File | Save As** command instead of **File | Save**.

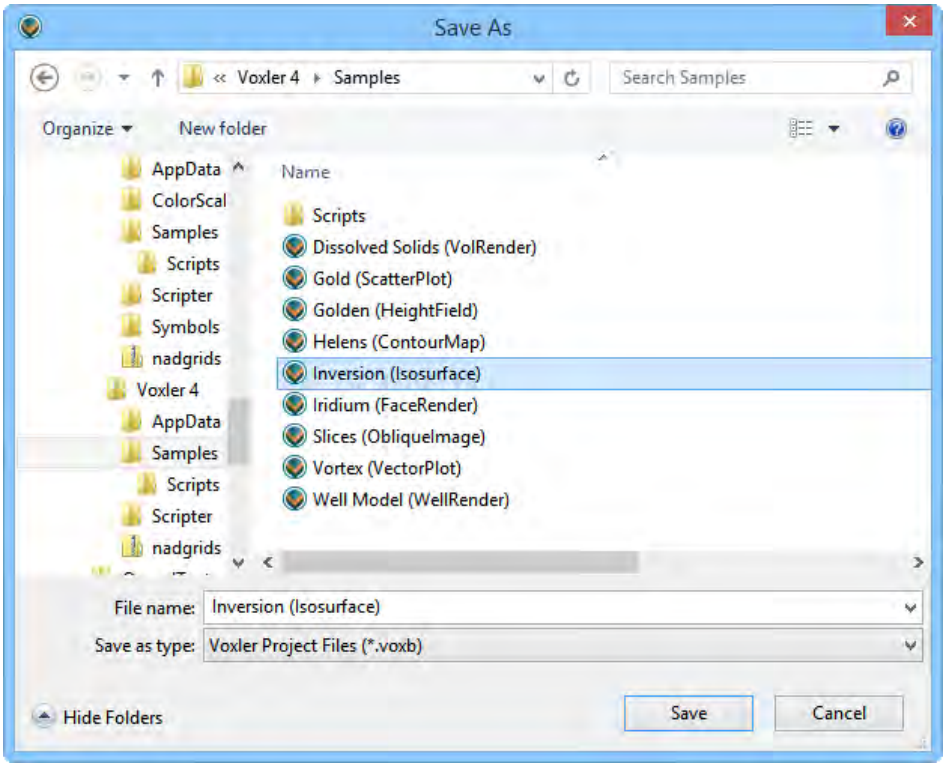
If you would like to save data from a module, click the **File | Save Data** command from the viewer window or the **File | Save** command in the worksheet window.

Save As

Click the **File | Save As** command to save a Voxler project .VOXB file or data file for the first time or to a new name. The **Save As** dialog is displayed. Set the directory and file name in the **Save As** dialog. **Voxler** allows .VOXB files to be saved with the **File | Save As** command. Data files can be saved in a number of data file formats. To create other file types, use the **File | Export** commands.

The Save As Dialog

Click the **File | Save As** command to open the **Save As** dialog.



*The **Save As** dialog is displayed, allowing you to set the directory and file name.*

The following options are available in the **Save As** dialog:

Save In

The *Save in* field shows the current directory. Click the down arrow to see the directory structure; click the folders to change directories.

Creating New Folders and Changing the View

The buttons to the right of the *Save in* field allow you to create new folders and change the view of the file list.

File List

The *File list* displays files in the current directory, which is listed in the *Look in* field. The *File of type* field controls the display of the file list. For example, if *Voxler Network Files (*.voxb)* is listed in the *Files of type* field, only *.VOXB files appear in the files list.

File Name

The *File name* field shows the name of the selected file. Type a path and file name into the box to save the file with a a specific name.

Specify a File Type

The *Save as type* field controls the file format for the saved file and the display of the file list. For example, if *Voxler Network Files (*.VOXB)* is listed in the *Save as type* field, only .VOXB files appear in the file list. To see all files in the directory, choose *All Files (*.*)* from the *Files of type* list. Click a file to place the file name in the *File name* field.

The *Save as type* field shows the file format to be saved. To change the file format, click the down arrow and select the file type from the list. *All Files (*.*)* displays all files in a directory. **Voxler** allows .VOXB files to be saved with the **File | Save As** command. To create other file types from a network file, use the **File | Export** or **File | Save Data** commands.

When the **Save As** command is clicked from the worksheet window, the data in the worksheet can be saved as a Comma Separated Variables (*.csv), Golden Software Data (*.dat), Sylk Spreadsheet (*.slk), Text Data (*.txt), Excel Spreadsheet (*.xls), or Excel 2007 Spreadsheet (*.xlsx) file type. Select the desired output file type in the Save as type list.

Save or Cancel

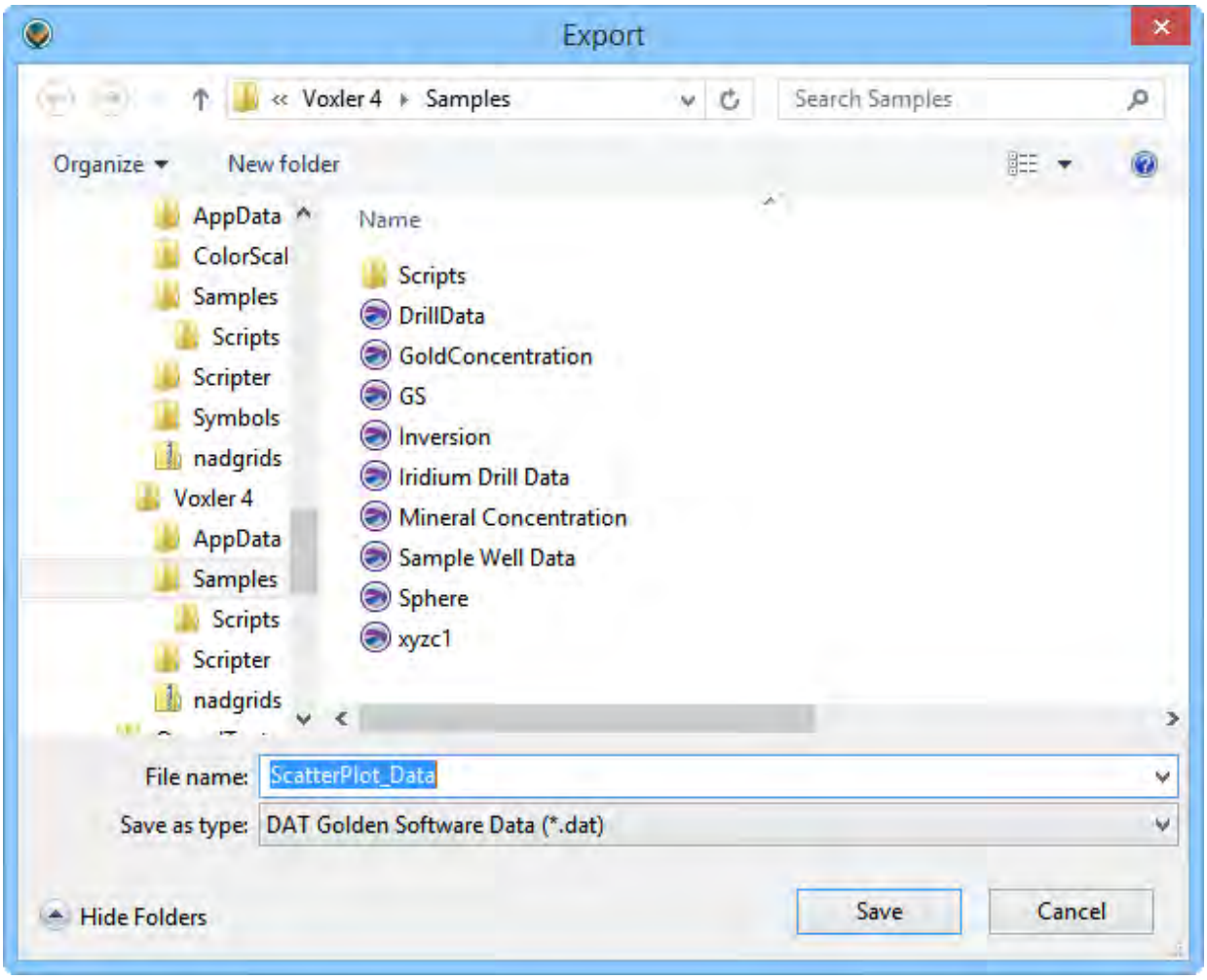
Click *Save* to close the **Save As** dialog and save the specified **Voxler** .VOXB file. Click *Cancel* to close the dialog without saving the new file.

Save Data

Click the **File | Save Data** command to save the output data from the currently selected module. You may also select the module whose output you wish to save, right-click the module and select **Save Data**. The **Export** dialog is displayed.

The Export Dialog

The File | Save Data command opens the Export dialog.



*Set the directory and file name in the **Export** dialog.*

The **Export** dialog contains the following options:

Save In

The *Save in* field shows the current directory. Click the down arrow to see the directory structure; click the folders to change directories.

Creating New Folders and Changing the View

The buttons to the right of the *Save in* field allow you to create new folders and change the view of the file list.

File List

The *File list* displays files in the current directory, which is listed in the *Look in* field. The *File of type* field controls the display of the file list. For example, if *DAT Golden Software Data (*.dat)* is listed in the *Files of type* field, only DAT files appear in the files list.

File Name

The *File name* field shows the name of the selected file. Type a path and file name into the box to save the file with a a specific name.

Specify a File Type

The *Files of type* field controls the display of the file list. For example, if *Voxler Network Files (*.VOXB)* is listed in the *Files of type* field, only .VOXB files appear in the file list. To see all files in the directory, choose *All Files (*.*)* from the *Files of type* list. Click a file to place the file name in the *File name* field.

The *Files of type* field shows the file format to be exported. To change the file format, click the down arrow and select the file type from the list. *All Files (*.*)* displays all files in a directory. **Voxler** allows .VOXB files to be saved with the **File | Save As** command. To create other file types, use the **File | Export** or **File | Save Data** commands.

Export files by typing a name into the *File name* field and selecting a file type from the *Save as type* field. For example, typing MYIMAGE in the *File name* field and choosing *BMP Windows Bitmap (*.bmp)* from the *Save as type* drop down menu results in MYIMAGE.BMP. The extension is automatically added. If a file extension is typed in the box along with the file name, the file type is determined by the typed extension. For example, if MYIMAGE.TIF is typed in the *File name* box, the file saves as a .TIF regardless of what is set in the *Save as type* field.

Save or Cancel

Click *Save* to close the **Save As** dialog and export the specified file. An **Export Options** dialog will likely appear. Click *Cancel* to close the dialog without exporting the file.

Specify the desired format in the *Save as type* list. The formats displayed in the list are dependent on the type of data being exported.

Tips on Saving Data

A few considerations:

- Most image formats are capable of storing byte data only. If a lattice consists of other data types, it must first be converted using the *ChangeType* module.
- Many image formats do not support the alpha channel, which controls transparency. In this case, it is ignored when RGBA images are saved.
- A lattice may be saved as tabular data (spreadsheet formats). Tabular data contains a row for every X, Y, Z, C point in the lattice. This can result in very large data files.
- A uniform lattice may be saved as one or more heightfield data files, e.g., .GRD, if the number of data components equals 1. If the lattice does not meet this requirement, then the heightfield data formats are removed from the list.
- A uniform lattice may be saved as one or more image (bitmap) slices if the data type is BYTE and there are one, three, or four data components. Single-component lattices are saved as grayscale images; three-component lattices are saved as RGB; and four-component lattices

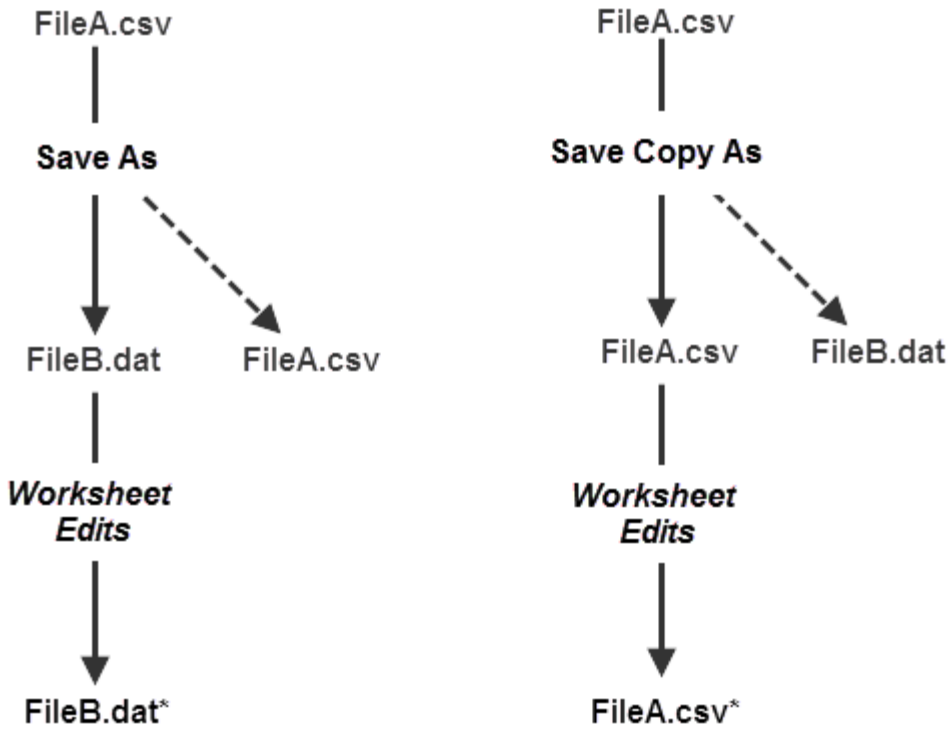
are saved as RGBA. If the lattice does not meet these requirements, then the image formats are removed from the list.

- Point sets may be output as tabular data (spreadsheet formats) or as one-dimensional curvilinear lattices, depending on the format chosen. In general, the lattice formats import and export quite a bit faster.
- If a multi-slice lattice is output to an image or heightfield data format, it is output as a series of slices in the Z direction. The **Select Slices** dialog appears.
- If rectilinear lattices are not supported by the chosen output format, they are automatically converted to curvilinear lattices.
- If an entire slice contains only null values the file will not be created for that slice. All slices with at least one non-null value will be exported as expected.

Save Copy As

The **File | Save Copy As** command saves a copy of the active worksheet as a new file. Unlike with the **Save As** command, the newly created file is not opened in the current worksheet window. The original file remains in the worksheet window. Subsequent changes are performed on the original file that existed in the worksheet before the **Save Copy As** command was clicked.

When the **File | Save As** command is used, the original file is closed in the state in which it exists when the **Save As** command is clicked. The newly created file, with the new file name, type, and/or location is opened in the original file's place. Subsequent changes are performed on the newly created file.



*In the example workflow on the left, the **Save As** command is used to save the active worksheet with a new file name and type. The new file is then edited in the worksheet window. In the example workflow on the right, the **Save Copy As** command is used to save a copy of the active worksheet with a new file name and type. The original file is then edited in the worksheet window.*

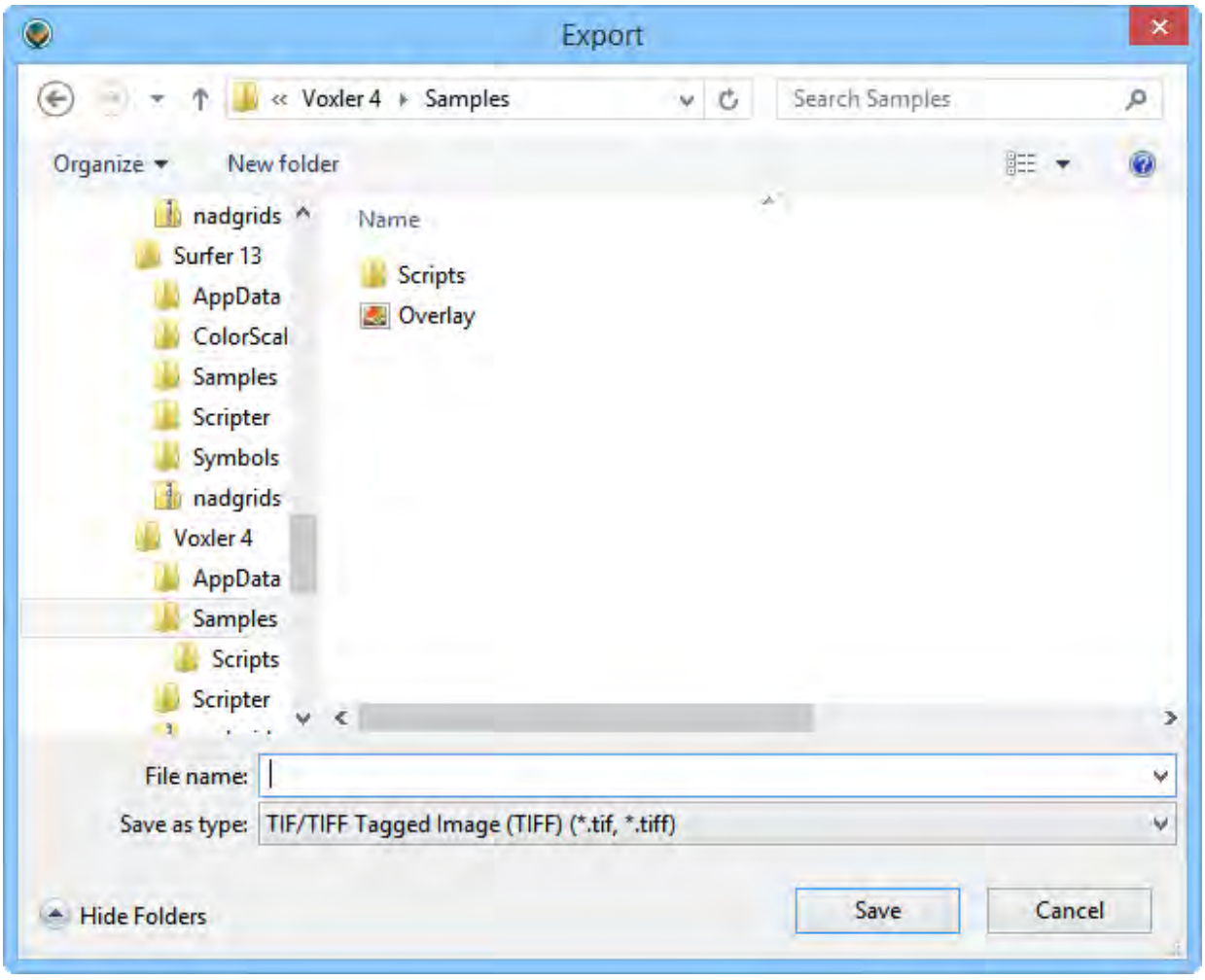
Export

Click the **File | Export** command or the  button to export the entire scene as an inventor scene graph .IV file or one of several different image file formats. This is a graphics-only export, i.e., no data are saved for this operation. **Voxler** is designed to export the visible portion of the plot when using the **File | Export** command. It uses the monitor to define the limits of the exported image. To export a larger or smaller portion of the display, zoom in or out prior to export.

When a scene is exported as an image file format, the **Export Options** dialog appears. This dialog allows you to specify various options for the exported file. Each file type has slightly different options. Refer to the specific file format for those options.

The Export Dialog

The **File | Export** command opens the Export dialog.



*Set the directory and file name in the **Export** dialog.*

The **Export** dialog contains the following options:

Save In

The *Save in* field shows the current directory. Click the down arrow to see the directory structure; click the folders to change directories.

Creating New Folders and Changing the View

The buttons to the right of the *Save in* field allow you to create new folders and change the view of the file list.

File List

The *File list* displays files in the current directory, which is listed in the *Look in* field. The *Save as type* field controls the display of the file list. For example, if *BMP Windows Bitmap (*.bmp)* is listed in the *Save as type* field, only BMP files appear in the files list.

File Name

The *File name* field shows the name of the selected file. Type a path and file name into the box to save the file with a a specific name.

Specify a File Type

The *Save as type* field controls the display of the file list and the file type for the exported file. For example, if *BMP Windows Bitmap (*.bmp)* is listed in the *Save as type* field, only BMP files appear in the files list. Click a file to place the file name in the *File name* field.

The *Save as type* field shows the file format to be exported. To change the file format, click the down arrow and select the file type from the list. *All Files (*.*)* displays all files in a directory. **Voxler** allows .VOXB files to be saved with the **File | Save As** command. To create other file types, use the **File | Export** or **File | Save Data** commands.

Export files by typing a name into the *File name* field and selecting a file type from the *Save as type* field. For example, typing MYIMAGE in the *File name* field and choosing *BMP Windows Bitmap (*.bmp)* from the *Save as type* drop down menu results in MYIMAGE.BMP. The extension is automatically added. If a file extension is typed in the box along with the file name, the file type is determined by the typed extension. For example, if MYIMAGE.TIF is typed in the *File name* box, the file saves as a .TIF regardless of what is set in the *Save as type* field.

Save or Cancel

Click *Save* to close the **Save As** dialog and export the specified file. An **Export Options** dialog will likely appear. Click *Cancel* to close the dialog without exporting the file.

Export Format Types

- Bitmap
- AVS X-Image .X, .XIMG
- Windows Bitmap .BMP
- Encapsulated Postscript .EPS
- GIF Image .GIF
- JPEG Compressed Bitmap .JPG, .JPEG
- PDF (Raster) .PDF
- Portable Network Graphics .PNG
- Portable Bitmap .PNM, .PPM, .PGM, .PBM

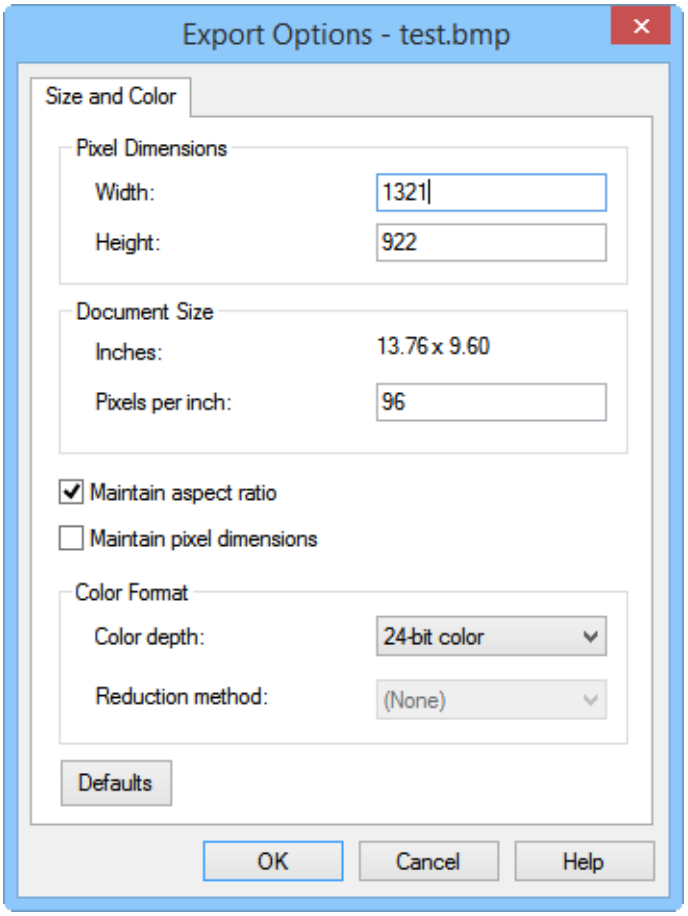
- RGB Silicon Graphics SGI RGB Image .RGB, .RGBA, .BW
- Sun Raster Image .RAS, .SUN
- Targa (TrueVision) .TGA
- Tagged Image .TIF, .TIFF
- Inventor .IV

Export Options Dialog

The **Export Options** dialog allows you to specify options to determine how the image is exported. See *Appendix A – File Formats* for file type specific **Export Options** dialogs.

Size and Color Page

This dialog appears when the **File | Export** command is chosen and an image format is selected as the output file type.



The **Size and Color** *page of the* **Export Options** *dialog controls options for image export.*

You may specify the output bitmap size in pixels (by modifying the *Width* and *Height* controls). You may also specify a resolution in dots per inch (by modifying the *Pixels per inch* control) and let the application calculate the output bitmap size for you. Higher resolution will yield a better-looking image, but at the expense of requiring more memory and disk space to hold the bitmap.

Image Size In Pixels

Choose the default *Width* and *Height* pixel settings for the output bitmap. Values are in pixels. The larger the values, the larger the output image.

Pixels Per Inch

Choose the *Pixels per inch* to change the number of pixels in the output image. The *Width* and *Height* of the *Pixel Dimensions* changes accordingly.

The .GIF file format is set to 72 *Pixels per inch* for all .GIF images and cannot be changed. GIF images are always 72 DPI by definition. For higher quality images, it is suggested that PNG, TIFF, or BMP is used instead of GIF.

Image Size

The *Document Size* section contains a *Inches* option that displays the current selected image size in inches. The image size is updated when the *Pixel Dimensions* or the *Pixles per inch* values are adjusted.

Maintain Aspect Ratio

Check the *Maintain aspect ratio* box if you want the image to maintain an equal horizontal and vertical resolution. When this option is checked, the output image maintains an aspect ratio of 1:1 in the output image. Higher resolution yields a better-looking image, but keep in mind that more memory and disk space are required to hold a high-resolution image.

Maintain Pixel Dimenions

Check the *Maintain pixel dimensions* box if you want the image to export at the selected *Width* and *Height*, but with a different number of *Pixels per inch*. This results in the same number of pixels, but a different *Document Size*.

Color Format

The *Color Format* gives you the option to output your image with *Color depth*. The greater the color depth, the more faithfully the image will represent the colors assigned to objects in your document. Different output formats support different color depths. Some output formats support 256 colors only, while others also support True Color (16 million colors).

Select one of the options from the *Color Depth* drop-down list. The options are: *8-bit grayscale*, *16-bit grayscale*, *32-bit grayscale*, *.1-bit color indexed*, *4-bit color indexed*, *8-bit color indexed*, or *24-bit true color*.

For example, Windows .BMP format supports *monochrome*, *16 colors*, *256 colors* and *True Colo*r. Greater color depth will yield a better-looking image, but at the expense of requiring more memory and disk space to hold the image.

Reduction Method

If you select a color indexed *Color depth*, you can choose a *Reduction Method*. Select one of the options from the *Reduction Method* drop-down list. The options are: *Ordered Dither*, *Diffuse Dither*, *Popularity*, *MedianCut555*, or *MedianCut888*.

If you select a color indexed *Color depth*, you can choose a *Reduction method*. Select one of the options from the *Reduction method* drop-down list. The options are: *Ordered Dither*, *Diffuse Dither*, *Popularity*, *MedianCut555*, or *MedianCut888*.

Dithering determines how similar colors are distributed among clusters of pixels in the reduced image. Ordered dithering uses a repeating pattern. Diffuse dithering uses a pseudo-random pattern.

Quantization determines how the colors for the exported image are selected from the palette of 16 million possible colors. *Popularity* uses the most frequently occurring colors in the image, and *MedianCut* selects colors based on the 'median cut' method that tries to select the most even distribution of colors over the range of colors that appear in the image. The Median method can use either 5 or 8 bits of sample data for each of the three color planes in the image. Larger sample sizes require more memory to perform the conversion for export, so the smallest sample size that produces an acceptable image is recommended.

Defaults

Click the *Defaults* button to return the export options to the default selections.

OK

Click the *OK* button to save your changes and export the file.

Cancel

Click the *Cancel* button to exit the dialog without saving your changes or exporting the file.

Copy Snapshot

Click the **Actions | Copy Snapshot** command or click the  button to copy a raster version of the current view of objects in the **Viewer** window to the clipboard. The size of the copied image is the same as it appears in the current **Viewer** window. The size is indicated in the lower right section of the status bar in the *Window size* display.

To paste the raster graphics to another program, switch to the other program and click the **Edit | Paste** command or press CTRL+V.

It is also possible to transfer graphics to another program by clicking the **File | Export** command and exporting the object to one of the many export file formats.

Capture Video

Click the **Actions | Capture Video** command or click the  button to record the current **Viewer** window. Adjust the window to the desired size before choosing the **Actions | Capture Video** command.

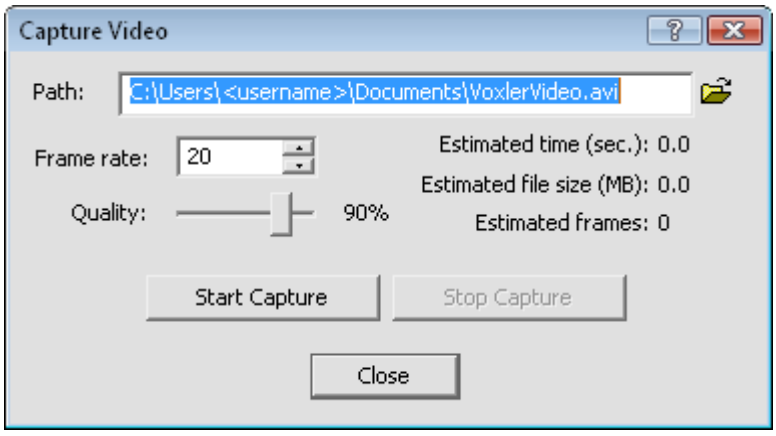
The video capture uses the size of the current **Viewer** window. The size of the window will affect the file size of the final .AVI file. The window screen size is displayed in the status bar and shows the size of the window screen in pixel width by pixel height.



The status bar displays the window size. This example shows the window as 710 pixels wide by 572 pixels tall.

Capture Video Dialog

Choose the **Actions | Capture Video** command to open the **Capture Video** dialog.



*Specify video capture settings in the **Capture Video** dialog.*

Path

The path specifies the location where you want to save the captured video. Click the  to open the **Choose Video Path** dialog to select a file location. Alternatively, enter a file name directly in the *Path* box. The default path is the *Project folder* path, and it is set in either the **Welcome to Voxler** dialog or the **Tools | Options** - **General** page.

Frame Rate

Frame rate is the frames per second used for the captured video. A frames per second value of 15-20 provides good results. A low frame rate will not be able to give the illusion of motion effectively and may produce a choppy animation. A high frame rate may produce an unnecessarily large file size without noticeable improvement in animation. Frame rates above 20 frames per second will produce a large file size, but retain better image motion. To change the value, highlight the existing value and type a new value. Alternatively, click the  to increase or decrease the value.

Quality

Use the *Quality* slider to adjust the quality of the video between 50 and 100%. A value of 100% is the highest quality. A value of 50% is the lowest quality allowed. Using a lower quality (i.e. 50%) creates a smaller file size and decreased animation quality. Using a higher quality (i.e. 100%) creates a larger file size with increased animation quality. A value of 80% or above is a reasonable selection. The default value is 90%. To change the value, click and drag the slider to the desired value.

Start Capture

Click the *Start Capture* button to begin the video capture. Any dialogs over the **Viewer** window will be captured in the video. During video capture, dialog size and location can be adjusted. All property and view changes to the **Viewer** window will be captured. If visible, the world axis triad

and legend are captured. The cursor is not captured. The *Estimated time (sec)*, *Estimated file size (MB)*, and *Estimated frames* information is dynamically displayed during capture.

Stop Capture

Click the *Stop Capture* button to end the video capture. The *Estimated time (sec)*, *Estimated file size (MB)*, and *Estimated frames* information is statically displayed when the video has successfully completed. The video is located in the *Path* location.

Close

Click the *Close* button to close the dialog.

Video Capture - Advanced Suggestions

This tutorial provides suggestions and strategies to help you get the most out of your **Voxler** video.

Voxler Capture Video

Click the **Actions | Capture Video** command to create a video in **Voxler**. The **Capture Video** dialog opens. Specify the settings for the video capture in the **Capture Video** dialog. See *Capture Video* for detailed information on the dialog.

The **Voxler** capture video tool gives you the ability to make .AVI movies of everything that appears in the **Viewer** window, including any angular spin you apply to the image. However, .AVI files can easily get quite large, and in a short amount of time. The following suggestions can help keep the quality of your .AVI files high while keeping them at a reasonable size.

1. **What screen size do you need?**

   The **Actions | Capture Video** command allows you to create an .AVI movie of the **Viewer** window. This means that when **Voxler** is set for full-screen usage the **Viewer** window can be as large as 800 X 1200 pixels. A 10-second .AVI file created at this size with a *Frame rate* of 20 and *Quality* of 90% is approximately 17.5 MB in size. However, if you reduce the **Viewer** window size to 350 X325, the .AVI file is 4.7 MB. The width by height size of the window screen is displayed as *Window size* in the status bar.

   How large a **Viewer** window you need depends on several factors, including the application where the .AVI is used, the type of image you want to capture (how much detail must be visible for maximum visual impact) and how willing you are to have a large .AVI file. Regardless, it is clear that this question is one of the first you should consider when creating your .AVI file. Adjust the size of the **Viewer** window first.

2. **What Frame Rate do you need?**

   The *Frame rate* is specified in the **Capture Video** dialog. Generally, the higher the *Frame rate* the smoother the motion in the .AVI file. However, if the rotational motion of the image in the viewer window is slow, the *Frame rate* can be set at a lower rate without sacrificing much in terms of visual smoothness. If the image is rotating more rapidly, a slower *Frame rate* can result in a jerky playback of the .AVI. The general rule is: if you use a slower rotation, you can use a lower *Frame rate* and not sacrifice image quality.

3. **What Quality Setting do you need?**

   The *Quality* is specified in the **Capture Video** dialog. The *Quality* of the image is akin to the image quality of a .JPG file. Artifacts from compression can interfere with image quality if the compression is too great, in exchange for a smaller file size. The amount of visible artifacts is

also dependent on the type of image in the .AVI file. Solid objects, such as the *Helens (ContourMap).voxb* example file, require a higher quality setting, such as 90%, to make artifacts unobtrusive. But if you are capturing a point data set, such as the *Gold (ScatterPlot).voxb* example file, you can use a lower quality setting of about 70% without seeing obvious compression artifacts. You can use a lower *Quality* in point images than in solid images.

Lower *Quality* settings, such as 50%, should be used only with very slowly rotating point source objects. Avoid using these low *Quality* settings with Lattice-based data set objects. The resulting .AVI files are virtually unusable:

4. **What duration should you make your .AVI file?**

   This seems obvious, but always remember the intended audience of this .AVI file, and what information you are trying to convey. If you feel the user needs 45 seconds to understand the basic point of the .AVI and the resulting file is too large, alter the settings to make the file size smaller. It is always a balancing act among screen size, image quality, frame rate, and duration.

5. **How rapidly should the image rotate?**

   The dramatic visual effect of a rotating three dimensional object or array is one of the primary reasons to make a **Voxler** capture video. You want to rotate the object in the **Viewer** window (by pressing the mouse button and releasing it while moving the mouse in the desired direction of motion) at the most appropriate angular speed to convey the desired information. If you rotate the image too quickly the information in the image is a blur; too slow and the effect is tedious.

   There is one point to remember regarding angular speed and file size: The faster the speed, the larger the file. For example, if you created a simple file of the *Gold (ScatterPlot).voxb* example, got it to spin as rapidly as possible, and created a ten-second movie of this whirling object with a screen size of about 500 X 330, the resulting .AVI file would have a size of over 8 MB. If you created another movie of the same image with zero angular speed (an unmoving object), the resulting .AVI file would be only 1.3 MB. Take this into consideration. If the motion effect of your movie is acceptable at a slower angular rate the resulting .AVI file size will be smaller.

   This is clearly another variable to take into account when creating your .AVI movie, because the reverse is true as well: Do not bore the audience with a glacially slow-moving image, despite the attractive option of a greatly reduced file size.

6. **What if the .AVI file is still too large**?

   It might happen that the movie you create is still too large for the intended use, no matter what settings you use in the **Capture Video** dialog. There are 3rd-party software programs that convert .AVI files to other video formats, such as .WMV or .MPEG files. In many cases the resulting converted file can be considerably smaller in size.

Other Suggestions

It is common in the video creation process to make/revise/revise through multiple iterations before completing the video satisfactorily. It is efficient to keep Windows Explorer open to the location where **Voxler** stores the created videos. This default location is the My Documents folder for the current user.

You can change the default location in **Voxler** by choosing the **Tools | Options** command. On the **General** tab click the button and navigate to the preferred default folder. This will be the default location for all files created and saved in **Voxler**.

Window Screen Size

The right section of the status bar displays the window screen size of the last redraw in the **Viewer** window. During operations in progress, the right section displays the estimated time remaining for tasks. The window size shows the size of the window screen in pixel width by pixel height.



The status bar commonly appears with the help text on the left; no text in the middle (when there is no progress to report); and window size information on the right.

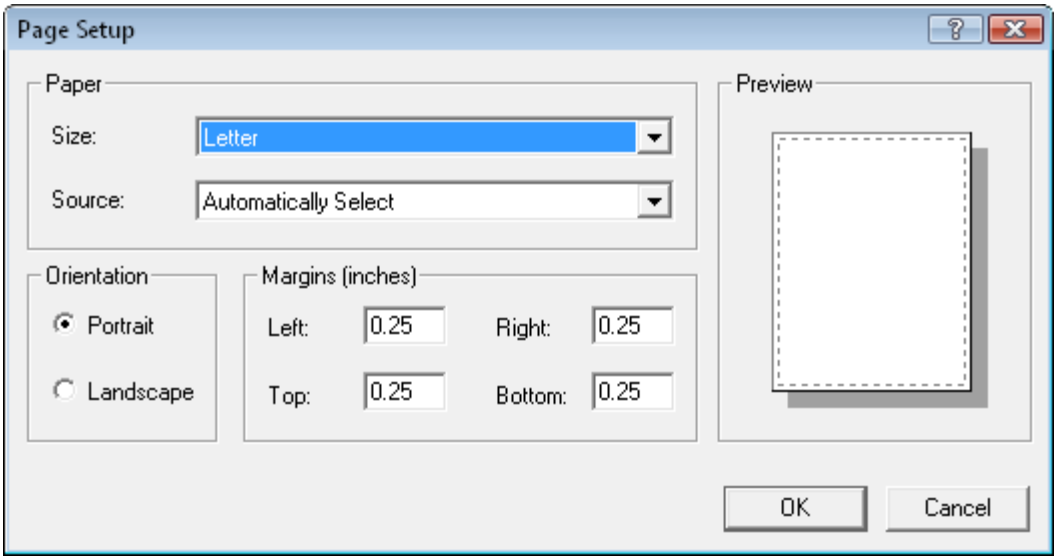
Knowing the width and height of the current window is useful when capturing a video, copying a snapshot, and exporting.

Page Setup

Click the **File | Page Setup** command to format the current document for printing. The **Page Setup** dialog appears.

The Page Setup Dialog

The **Page Setup** dialog controls the printed output from **Voxler**.



*Use the **Page Setup** dialog to specify the parameters of the printed page.*

The **Page Setup** dialog has the following options.

Paper

Use the *Paper* group to choose the paper *Size* and *Source* for the active printer. These options are based on the selected printer. Click the down arrow next to *Size* to select different paper

dimensions. *Letter* is selected by default. If your printer has multiple print trays, choose the paper *Source* by clicking the down arrow.

Orientation

The *Orientation* group controls whether the page is set to *Portrait* (vertical) or *Landscape* (horizontal) mode. Choose *Portrait* to align the printed page vertically. Choose *Landscape* to align the printed page horizontally.

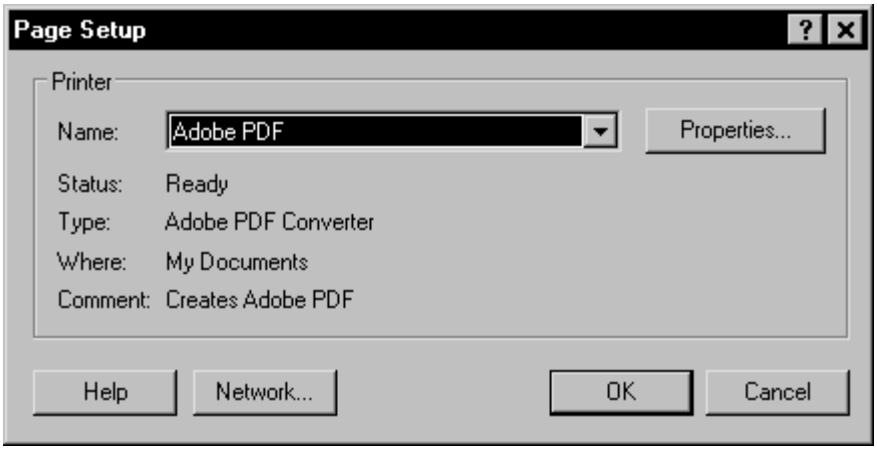
Margins (inches)

Use the *Margins (inches)* group to set the page margins for all sides of the printed page. Set the *Left*, *Right*, *Top*, and *Bottom* values in inches to any limits the selected printer will allow. Setting the margins does not move an existing output on the page. If the margins are set too far into the page, objects are truncated at the margin when printing.

The default margins are 0.25 inches (6.35 mm). Change the margins by entering new numbers in the *Left*, *Right*, *Top*, or *Bottom* boxes.

Printer

Click the *Printer* button to choose a different printer. Some options in the dialog may change depending on the new printer type chosen. Note, this option is not available in Vista, Windows 7, or Windows 8.



*Click the Printer button to display the **Page Setup** dialog with the Printer options. The options are dependent on the printer selected.*

Page Setup - Worksheet

Before printing the worksheet, the page format of the worksheet can be set through **File | Page Setup**. The **Page Setup** dialog controls items such as printing headers and footers, centering the data on the page, showing grid lines, etc. There are three pages in the **Page Setup** dialog: **Page**, **Margins**, and **Options**. There is a *Printer* button at the bottom of the dialog that allows you to set the default printer regardless of which page is selected.

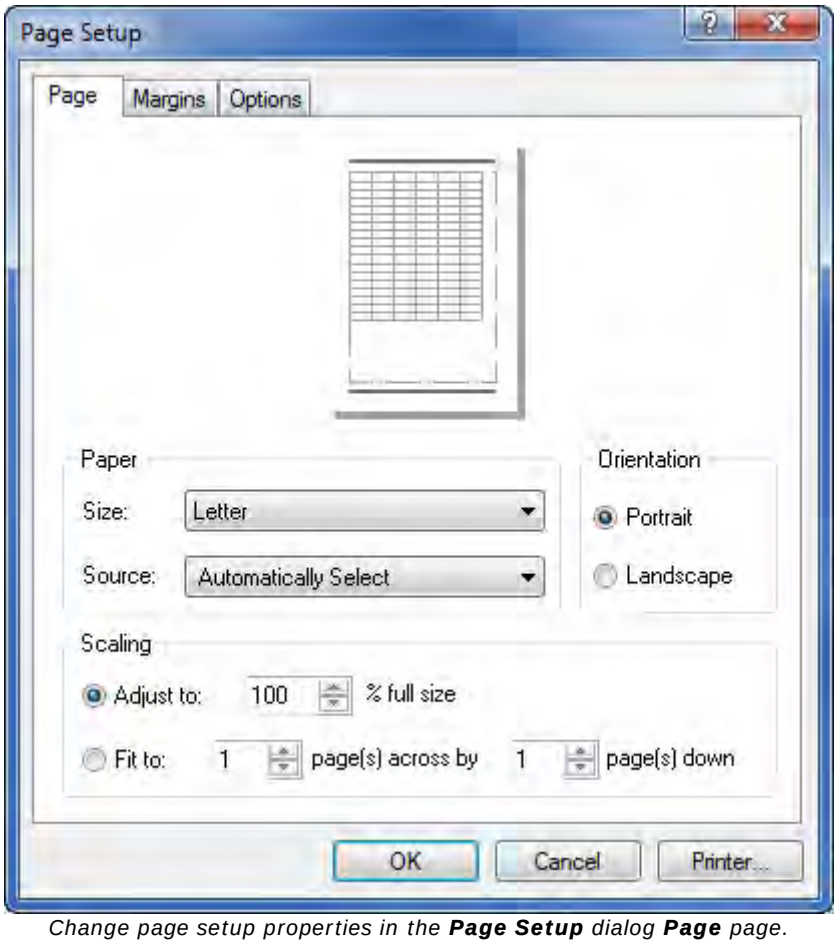
The worksheet **Page Setup** dialog has three pages:

Page	Set page size and scaling.
Margins	Set page margins, header and footer positions, and centering.
Options	Set grid lines, page order, and content of headers and footers.

Page Setup (Worksheet) - Page

Click on the **Page** tab in the **Page Setup** dialog to set page size and scaling.

Click the **File | Page Setup** command in the worksheet to open the **Page Setup** dialog.



*Change page setup properties in the **Page Setup** dialog **Page** page.*

Paper

Use the *Paper* section to choose the paper *Size* and *Source* for the active printer.

Size

Click the down arrow next to the paper *Size* to change the size of the paper. The paper size options available for your printer are listed in the list.

Source

If your printer has multiple print trays, choose the paper *Source* by clicking the down arrow.

Orientation

The *Orientation* section controls whether the page is set to *Portrait* or *Landscape* mode. Select *Portrait* to have a vertical page. Select *Landscape* to have a horizontal page.

Scaling

The *Scaling* section controls the printed size of the worksheet. There are two options with *Scaling: Adjust to* and *Fit to*.

Adjust To

The *Adjust to ___ % full size* option sets the percent of full size that the worksheet will print. The arrow buttons are used to scroll up or down from 100% (full size), or values can be typed into the box. The *Adjust to ___ % full size* setting is independent of the *Fit to* option.

Fit To

The amount of data in the worksheet determines how many pages are required to print the worksheet. The *Fit to __ page(s) across by __ page(s) down* option tells the program to print the worksheet at 100% scale or less. This option does not automatically scale the printed worksheet greater than 100%. This option is most useful when the worksheet is large and the number of printed pages needs to be limited.

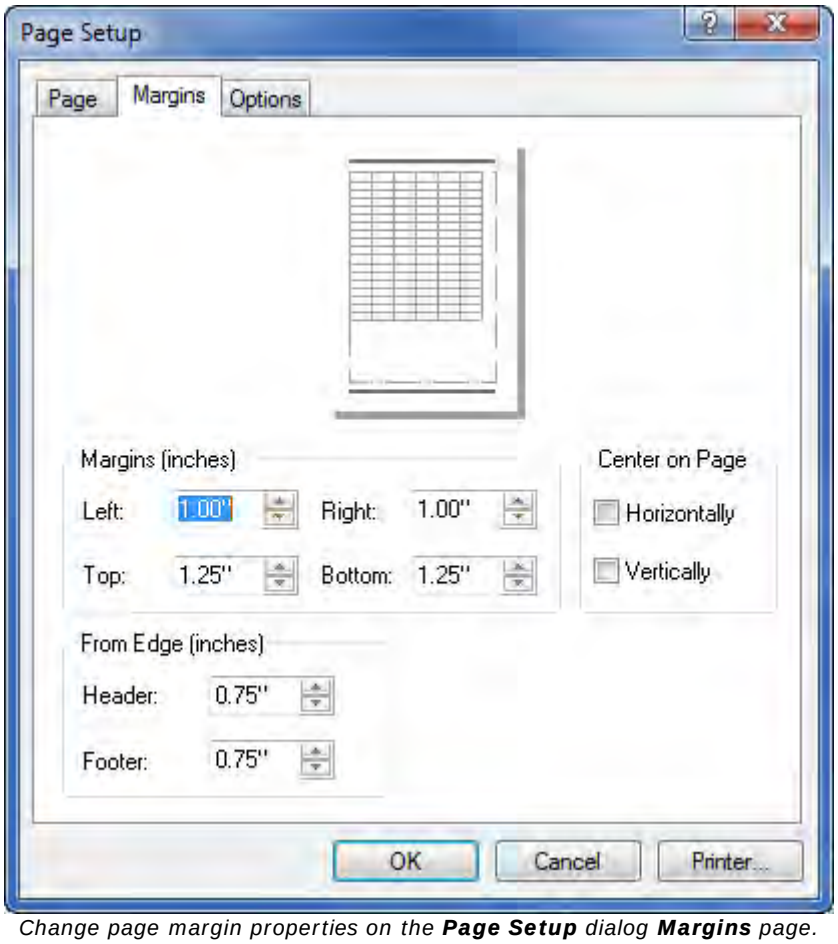
Printer

The active printer can be changed by clicking the *Printer* button at the bottom of the **Page Setup** dialog.

Page Setup (Worksheet) - Margins

Use the **Margins** page in the **Page Setup** dialog to set page margins, header and footer positions, and centering.

Click the **File | Page Setup** command in the worksheet to open the **Page Setup** dialog.



*Change page margin properties on the **Page Setup** dialog **Margins** page.*

Margins

Use the *Margins (inches)* group to set the page margins for all sides of the printed page. Set the *Left*, *Right*, *Top*, and *Bottom* values in inches to any limits the printer will allow. The margins are for the worksheet printout and are independent of the settings used for *Headers* or *Footers*.

Center on Page

The *Center on Page* group options automatically center the printout *Horizontally*, *Vertically*, or both. If neither option is selected, the worksheet prints in the upper left corner of the page.

From Edge (inches)

Headers and *Footers* are the only items that print outside of the margins. The *From Edge (inches)* group options controls how far the *Header* or *Footer* is printed from the edge of the page. If these values are greater than the *Top* or *Bottom* margins, it is possible that the worksheet data can print over the header or footer. The text that is printed for the header and footer is controlled on the **Options** page.

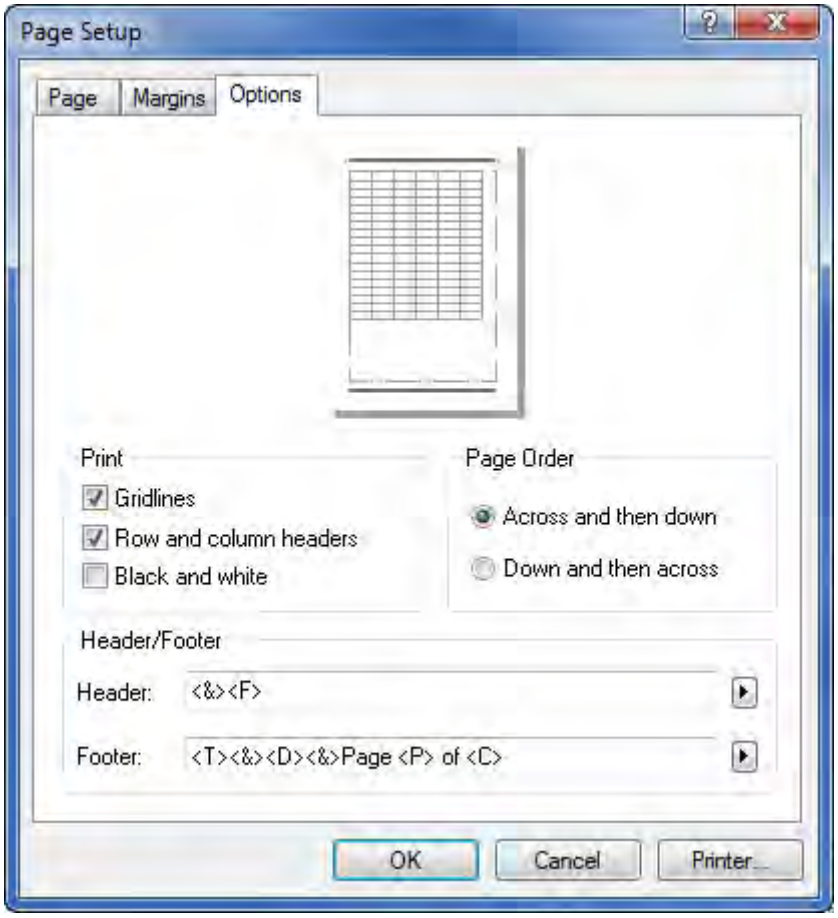
Printer

The active printer can be changed by clicking the *Printer* button at the bottom of the **Page Setup** dialog.

Page Setup (Worksheet) - Options

Use the **Options** page in the worksheet **Page Setup** dialog to set grid lines, page order, and content of the header and footer.

Click the **File | Page Setup** command in the worksheet to open the **Page Setup** dialog. Click on the **Options** tab at the top of the dialog.



*Change page option properties in the **Page Setup** dialog **Options** page.*

Print

The *Print* section controls how the worksheet information is printed.

Gridlines

Check the *Gridlines* option to draw grid lines separating each column and row.

Row and Column Headers

Check the *Row and column headers* option to print the column letters and row numbers of the worksheet.

Black and White

If cells contain color backgrounds (set from the **Format | Format Cells** command), check the *Black and white* option to print the worksheet in only black and white.

Page Order

The *Page Order* section controls the order in which multiple pages are printed. The *Across and then down* option prints from left to right first, and then moves down and prints left to right again. The *Down and then across* option prints the worksheet from top to bottom first, and then moves to the right and prints top to bottom again.

Header/Footer

The *Header/Footer* group controls the type of information included in the worksheet data print out. The plot window does not have header/footer options. The *Header* appears at the top of the page, and the *Footer* appears at the bottom of the page. The header and footer are spaced from the edge of the page based on the *From Edge* option o, the **Margins** page. Descriptive text can be typed in the *Header* and *Footer* boxes, or click the arrows to the right of the boxes and click the items in the list.

Automatic header/footer codes:

- *File Name* (< F>) prints the name of the active file. The drive and path are not included.
- *Page Number* (<P>) prints the page number for each page. When several pages are printed, the order of printing is controlled from the *Page Order* section.
- *Total Page Count* (<C>) prints the total number of pages that are required to print out the worksheet with the specified scaling parameters.
- *Current Date* (<D>) prints the current date.
- *Current Time* (< T>) prints the current time.
- *Left/Center/Right Separator* (<&>) separates the header and footer text so it is spread out across the page. Too many separators can actually push text off the page. If this happens, remove the <&> separator, and use spaces instead.

Printer

The active printer can be changed by clicking the *Printer* button at the bottom of the worksheet **Page Setup** dialog.

Examples

For a six page document, <&><&>Page <P> of <C> would print (on the right side of the first page):

```
Page 1 of 6
```

Enter Joe Smith<&>< F><&><D> to print out a name, file name, and date:

```
Joe
Smith        COLORADO.DAT        01/05/10
```

To print centered text use the "&" operator one time, such as <&>< F>.

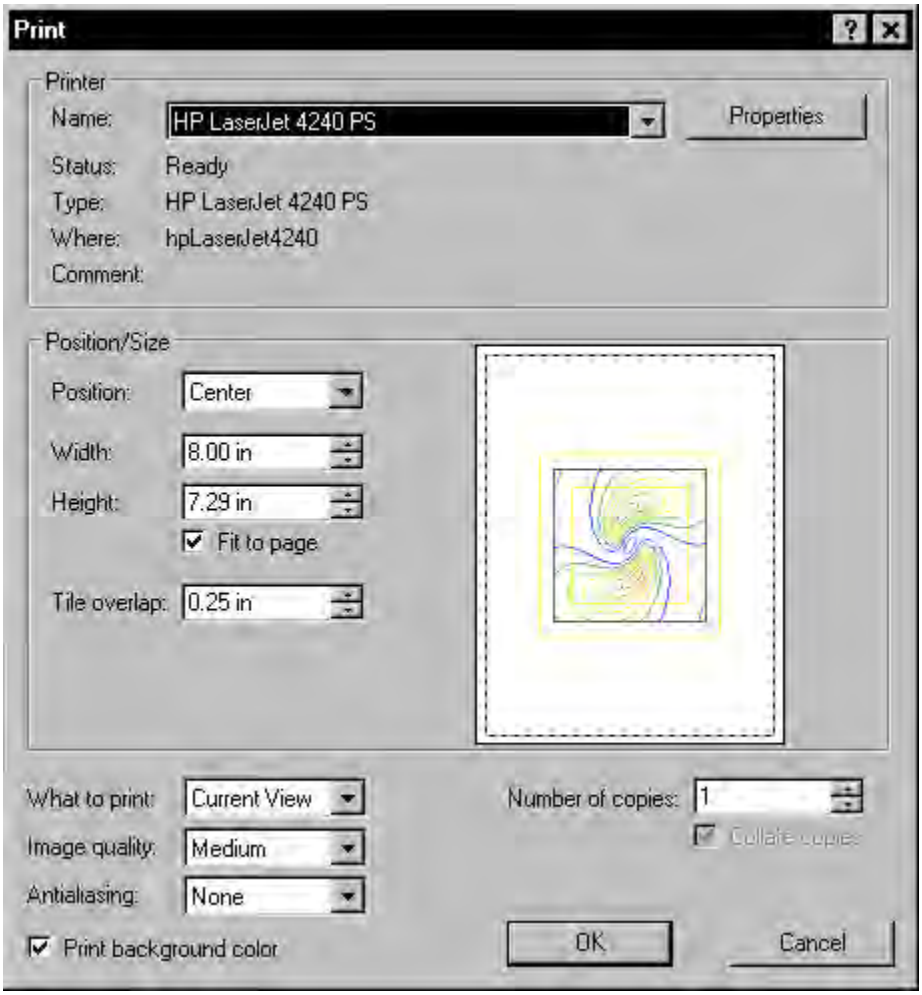
```
FILENAME.DAT
```

Print

Click the **File | Print** command or click the  button to print the active document. The **Print** dialog opens.

The Print Dialog

The **File | Print** command opens the **Print** dialog.



*Control your printed output with precision and ease. Use the **Print** dialog to change print settings and view your output before you send the print job to the printer.*

The following options are available in the **Print** dialog:

Printer

The *Printer* options specify which printer to use. The printer *Status*, *Type*, *Where*, and *Comments* are listed below the printer *Name*.

Name

The default printer is listed in the *Name* field. If more than one printer is installed on the computer, click the down arrow to the right of the *Name* field to select a different printer (if needed).

Properties

The *Properties* button controls the printer settings. A **Properties** dialog specific to your printer opens. For information on specific printer settings, see the owner's manual for the printer.

Position / Size

Position specifies where on the page the image should be printed (*Upper Left* or *Center*). Adjust the *Width* or *Height* of the printed image, if desired.

Check the *Fit to page* box to scale the image to fit within the specified margins. Margins are set and displayed with the **File | Page Setup** command.

Enter a value next to *Tile overlap* to control the overlap between tiled pages that comprise oversized documents.

What to Print

The *What to print* options control how the document pages are printed. *Current View* prints the portion of the scene that is visible in the current **Viewer** window. *All* prints the entire scene, including parts that might not be visible in the current **Viewer** window.

Image Quality

Given that DPI is the printer's resolution in dots per inch, choose *Low* (DPI/8), *Medium* (DPI/4), or *High* (DPI/2) image quality. Low quality uses less memory (both on the computer and within the printer) and prints faster at the expense of image quality. High quality uses a high internal resolution to obtain excellent output results but requires more memory and time.

Antialiasing

Adjust the degree of antialiasing of the printed image. Antialiasing reduces jagged lines by surrounding the lines with intermediate shades of gray or color, depending on the printer type, and is useful when a printer's resolution is too low to create smooth lines and images. Choosing *16-Pass* results in the smoothest image; choosing *None* results in no antialiasing.

Print Background Color

Check the box next to the *Print background color* option to include the window background color when printing. If this option is not selected, the printed background defaults to white.

Number of Copies

Specify the number of copies to print. If two or more copies of multiple page documents are printed, check the *Collate copies* box to separate the copies into collated packets.

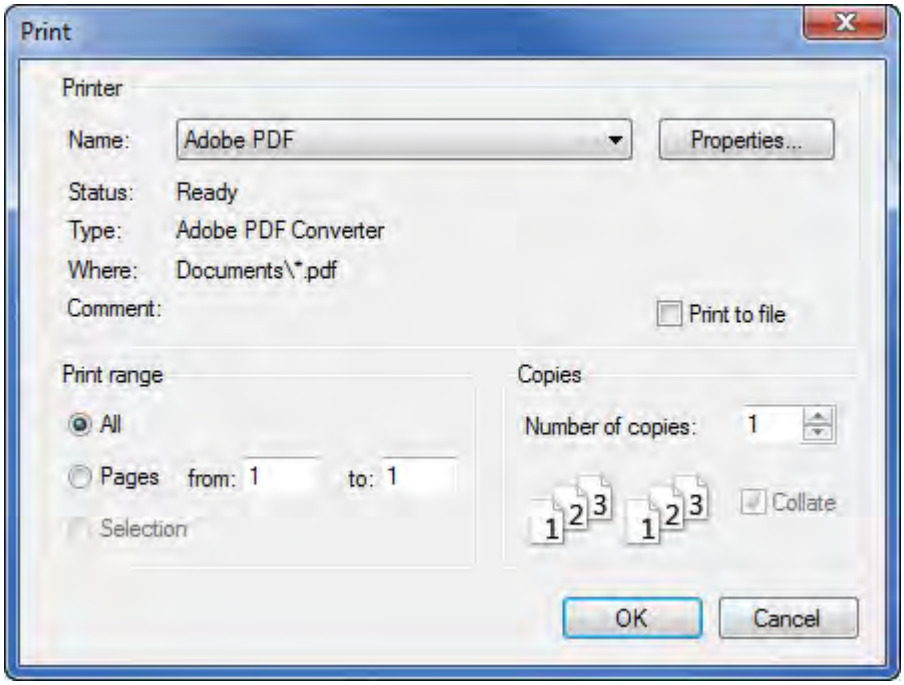
OK or Cancel

Click *OK* to start printing the document. Click *Cancel* to return to the **Voxler** window without printing the document.

Print Dialog - Worksheet

Click the **File | Print** command, or click the  button to print the contents of the worksheet to the active printer or to a . PRN file. To control the display of data on the printed page, refer the **File | Page Setup** command. While the worksheet is spooling, a dialog indicates that printing is progressing.

The **File | Print** command in the worksheet opens the **Print** dialog.



*Use the **Print** dialog to specify the printing options.*

Printer

The *Printer* section contains options to specify the printer.

- The default system printer is listed in the *Name* field. If more than one printer is installed on the computer, use the down arrow to the right of the printer name to select a different printer.
- The printer *Status*, *Type*, *Where*, and *Comment* are listed below the printer *Name*.
- Click the *Properties* button to set printer specific properties. For information on specific printer settings, see the owner's manual for the printer.
- The *Print to file* check box allows you to print the data to a .PRN file. .PRN files are ASCII text files. When this option is checked, click *OK* in the **Print** dialog after setting other printing options. The **Print to File** dialog will open. Enter a path and file name in the **Print to File** dialog, and click *Save*.

Print Range

The *Print range* options control how the worksheet pages are printed.

- *All* prints all the pages that contain data.
- *Pages* prints the pages specified. Enter the starting page in the *from* box. Enter the ending page in the *to* box.
- *Selection* prints the selected worksheet cells.

Number of Copies

Specify the number of copies to print in the *Number of copies* box. If two or more copies of multiple page documents are printed, check the *Collate* box to separate the copies into packets. Note that some printers do not allow multiple copies.

Collate

Check the *Collate* box to collate the pages when multiple page documents are printed two or more times.

OK or Cancel

Click *OK* to start the print. Click *Cancel* to abort the print and return to the worksheet window.

Chapter 21 - Editing Modules

Edit Menu Commands

Undo	Undo the last operation performed
Redo	Redo the previously undone action
Clear Undo History	Remove the **Undo** and **Redo** commands from memory
Copy	Copy the currently selected module to the clipboard
Paste	Paste the module from the clipboard to a new location
Delete	Delete the currently selected module
Delete All	Delete all modules in the network
Rename	Rename the currently selected module

Undo

Click the **Edit | Undo** command, press CTRL+Z or click the  button to reverse the last operation performed. If the last operation cannot be reversed, the **Undo** command is not available. After you close a file, the undo operations are cleared, so you cannot undo any operations performed before the file was closed. After you have undone an operation, the **Edit | Redo** command becomes highlighted, allowing you to reverse the just completed **Undo** command.

Undo Levels

Up to 100 undo levels can be set though **Tools | Options | General**. Enter a value between zero and 100 in the *Undo Levels* box.

Undo Limitations

Some actions cannot be reversed with the **Undo** command, including rotations, operations performed with a dragger, transparency changes, and **Tools | Options** commands. **Undo** is grayed after selecting these commands.

Using the **Undo** command with a *Colormap* is a special condition. When selecting **Undo** after altering a Colormap, all changes made in the Colormap are undone, reverting back to the original condition before the **Colormap Editor** dialog was opened.

Using the **Undo** command with the *Auto Update* box unchecked is a special condition. When selecting **Undo** with the *Auto Update* box unchecked, you still need to press the *Update Now* button or choose the **Network | Update Network** command to show the change made after selecting **Undo**.

Redo

Click the **Edit | Redo** command or click the  button to reverse the last **Edit | Undo** command. After you have used the **Edit | Undo** command, the **Redo** command indicates the operation that you have undone.

Clear Undo History

Click the **Edit | Clear Undo History** command to empty all commands from the **Edit | Undo** and **Edit | Redo** lists. The **Clear Undo History** command is permanent and cannot be reversed. Clearing the undo history is useful when needing to save memory.

After selecting the **Clear Undo History** command, you are prompted with a dialog. If you are sure you want to erase all the command history, click *OK*. Otherwise, click *Cancel* to return to the **Voxler** window.

Copy

To copy a module to the clipboard, click on the module in the **Network Manager** to select it. Click the **Edit | Copy** command, the  button, press CTRL+C on the keyboard, or right-click on a module in the **Network Manager** and select **Copy** from the context menu.

A copied module can be pasted into the **Network Manager** using the **Edit | Paste** command in order to create a duplicate. Modules can be pasted to a new document, as well. All properties of the original module are remembered in the pasted module.

This command is disabled (grayed out on the **Edit** menu and toolbar) if no module is selected.

Paste

To paste a module from the clipboard to the **Network Manager**, click the **Edit | Paste** command, click the  button, or click in the **Network Manager** and press CTRL+V. The module to be copied must first be placed in the clipboard using the **Edit | Copy** command in **Voxler**. A module can be pasted in the **Network Manager** in the active running instance of **Voxler** or in a new instance of **Voxler**.

Use the **Edit | Undo** command to reverse pasting a module into the **Network Manager**. Use the **Edit | Redo** command to reverse the last **Undo** command.

Delete Module

Click the **Edit | Delete** command, click the  button, right-click on a module in the **Network Manager** and select **Delete** from the context menu, or press the DELETE key on the keyboard to delete the currently selected module in the **Network Manager**. Upstream and downstream modules are disconnected from the deleted module; other modules are not deleted but remain in the **Network Manager**.

The **Delete** command deletes a single selected module. Only one module can be selected at any given time, so only one module can be deleted at a time. The **Edit | Delete All** command can be used to delete all modules in the **Network Manager**. The **Edit | Delete All** should be used with caution since it cannot be undone and all modules in the **Network Manager** are removed.

This command is disabled (grayed out on the **Edit** menu) if no module is selected.

When I delete a module, why are all of the downstream modules not deleted?

Only the selected module is deleted when you choose the **Edit | Delete** command or when you right-click on a module and select **Delete** from the context menu.

This command has the advantage of saving the other modules for use with a new input module. For example, say that you have a data module with a *BoundingBox* and *ScatterPlot* attached. You can delete the data module, load a new data module of a different type, and attach the *BoundingBox* and *ScatterPlot* modules without having to recreate the modules or any custom settings you may have specified.

Choose the **Edit | Delete All** command to delete all modules in the **Network Manager**. All modules except *Viewer Window* module are deleted.

Delete All Modules

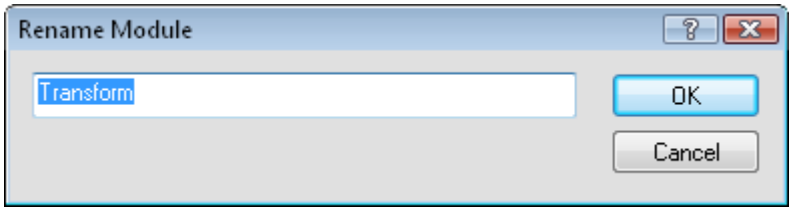
Click the **Edit | Delete All** command or press CTRL+DELETE to delete all modules in the **Network Manager** except for the *Viewer Window*. This command is functionally equivalent to the **File | New | Project** command, except you are not prompted to save the existing visualization project to a .VOXB file when using this command.

The **Delete All** command should be used with caution since it cannot be undone and all modules in the **Network Manager** are removed.

Rename Module

Click the **Edit | Rename** command or press the F2 key on the keyboard to assign a unique name to any selected module. After selecting a single module, choose the **Edit | Rename** command, press F2 on the keyboard, or right-click on the module in the **Network Manager** and select **Rename** from the context menu. The **Rename Module** dialog opens. Enter a new name in the box and click the *OK* button. The name appears in the **Network Manager** and in the Property Manager when the object is selected.

The **Rename** command requires the new name be a unique name. If the name already exists as a name of another module in the **Network Manager,** clicking the *OK* button issues a warning message and does not change the name. A new name must be entered before clicking *OK* or the warning dialog will appear again.



*Use the **Rename Module** dialog to customize a dialog name.*

This command is disabled (grayed out on the **Edit** menu) if no module is selected.

Colors and Colormaps

Color Palette

The color palette is opened by clicking the color sample or button. Once the color palette opens:

- Select a color from the palette by clicking on a color.
- Select additional standard colors or create new custom colors by clicking the *Other* button at the bottom of the palette. This opens the **Colors** dialog. In the **Colors** dialog, click the *Standard* or *Custom* tab to open the appropriate page.



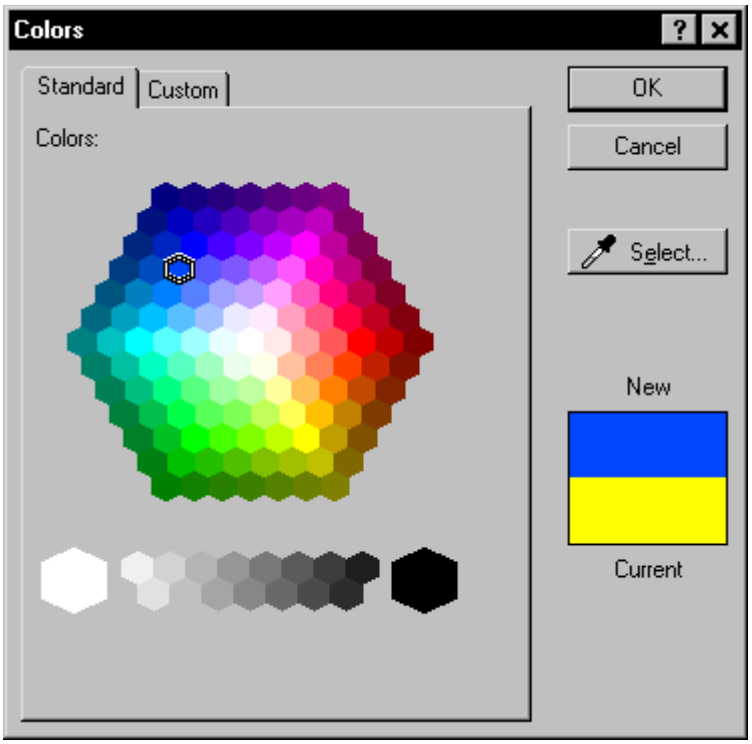
Select a predefined color or a custom color in the color palette.

Colors Dialog

Use the **Colors** dialog to load standard colors or create new colors for use by **Voxler**. The **Colors** dialog appears when the *Other* button is clicked on the color palette.

Selecting Standard Colors

The standard colors appear on the **Standard** page in a standard palette spectrum.



*Use the **Colors** dialog, **Standard** page to load standard colors from the palette.*

Colors

Click a color in the standard palette spectrum.

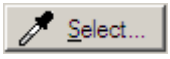
New

A preview of the selected color appears under *New* on the right side of the dialog.

Current

A preview of the current color appears above *Current* on the right side of the dialog.

Select

Click the 

button to color match to any color on the screen. The cursor changes to an eyedropper. Move the cursor around the screen and the color under *New* changes accordingly. Left-click the mouse when you find the color you want, and the color appears under *New*. Click the *OK* button to accept the new color. The **Colors** dialog closes.

OK

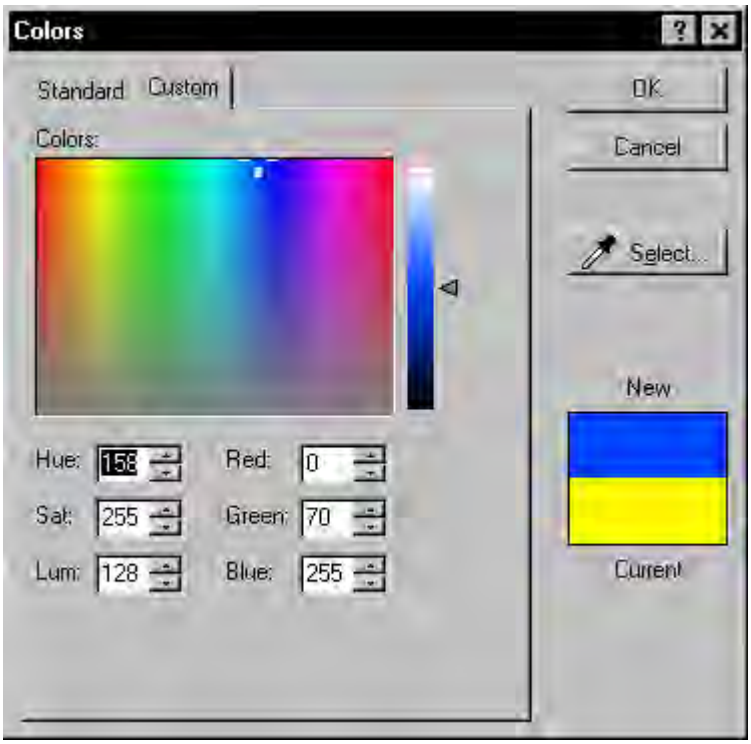
Click the *OK* button to accept the new color. The **Colors** dialog closes.

Cancel

While in the **Colors** dialog, click the *Cancel* button to close the dialog without making any color changes.

Creating Custom Colors

Create custom colors on the **Custom** page. New colors are created by adjusting *Hue*, *Sat*, *Lum*, *Red*, *Green*, or *Blue* values.



*Use the **Colors** dialog, **Custom** page to create custom colors.*

Colors

Use the *Colors* group to either select a color from the color spectrum, the color slider, or adjust the values to specify a color.

Left-click anywhere in the color slider to select a color. Drag the slider next to the color spectrum to adjust the new color's intensity. Dragging the slider will change the colors available in the color spectrum.

Left-click anywhere in the color spectrum to choose a new color.

Enter value between zero and 255 in the *Hue*, *Sat*, *Lum* or the *Red*, *Green*, Blue boxes. Changing the *Hue*, *Sat*, and/or *Lum* will automatically adjust the *Red*, *Green*, and/or *Blue* values, and vice versa.

Hue, Sat, Lum

The *Hue*, *Sat*, and *Lum* boxes show the amount of hue, saturation, and luminance used to form the color. The HSL values range from 0 to 255. To change the color amounts, enter a new value with the keyboard or click the up and down arrow buttons to adjust the values.

Red, Green, Blue

The *Red*, *Green*, and Blue boxes show the amount of each color used to form the color. The RGB values range from 0 to 255. To change the color amounts, enter a new value with the keyboard or click the up and down arrow buttons to adjust the values.

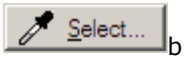
New

A preview of the new color appears under *New* on the right side of the dialog.

Current

A preview of the current color appears above *Current* on the right side of the dialog.

Select with Eyedropper

Click the 

button to color match to any color on the screen. The cursor changes to an eyedropper. Move the cursor around the screen and the color under New changes accordingly. Left-click the mouse when you find the color you want, and the color appears under *New*. Click the *OK* button to accept the new color. The **Colors** dialog closes.

OK

Click the *OK* button to accept the new color. The **Colors** dialog closes.

Cancel

While in the **Colors** dialog, click the *Cancel* button to close the dialog without making any color changes.

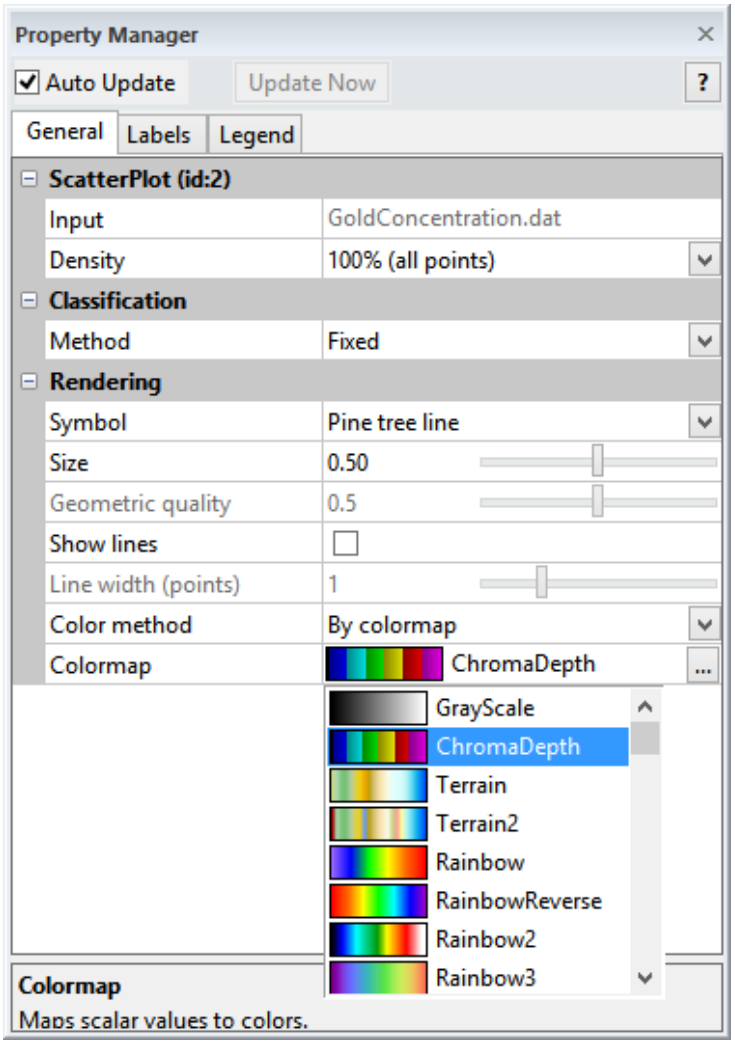
Colormap

The *Colormap* property allows you to map scalar values to color. Colormaps are often used to map color information onto one-, two-, and three-dimensional geometry. Each data value is mapped to a single color in a Colormap. This ensures the same value will always have the same color. This is a simple way to increase the information content of a visualization.

The Colormap does not display if an RGBA color lattice is used for input, as those colors are obtained directly from the lattice. The Colormap does not display if the color is a fixed color. If you expect a *Colormap* option and see a *Color* option instead, change the *Color Method* to *By Data*.

Predefined Colormaps

Select a predefined Colormap from the drop-down menu. The network is automatically updated to show the new color scheme. Use the vertical scroll bar to see all of the available predefined Colormaps. A predefined color map can be adjusted and customized by first selecting the desired Colormap and clicking the ... button to open the **Colormap Editor** dialog.



In this example, the Colormap is used to specify color for a ScatterPlot. Select a predefined color combination from the Colormap drop-down list .

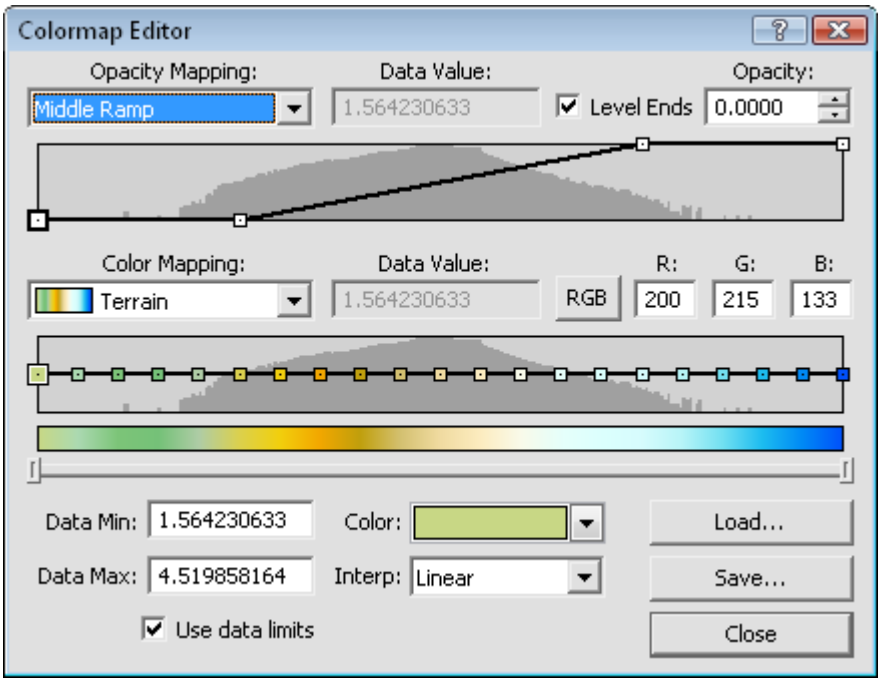
Custom Colormaps

Click the ... button to open the **Colormap Editor** dialog to create custom colors and effects. Custom Colormap files can be loaded and saved in the **Colormap Editor** dialog.

Colormap Editor

The **Colormap Editor** dialog is used to create custom colors and effects for use as a Colormap.

Click the ... button to the right of the *Colormap* option in the **Property Manager** to open the **Colormap Editor** dialog.



*Use the **Colormap Editor** dialog to create custom colors for a Colormap.*

The **Colormap Editor** dialog sections are described below.

Upper Section of Dialog: Opacity Mapping Controls

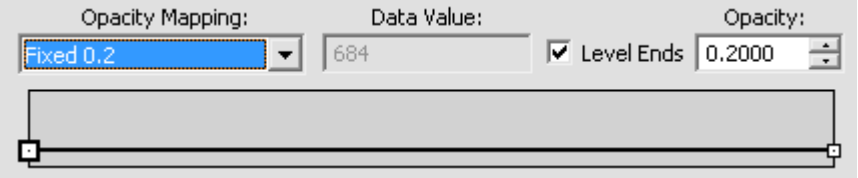
The upper section of the dialog contains the opacity mapping controls. These controls specify how a range of data values are mapped to opacity in the final output. Note that opacity is the opposite of transparency (0.0 is completely transparent, 1.0 is completely opaque).

Opacity Mapping

The *Opacity Mapping* drop-down displays a series of predefined opacity settings. These may be used to preset the opacity settings which can then be modified as desired. The *Opacity Graph* is updated to display the selection.

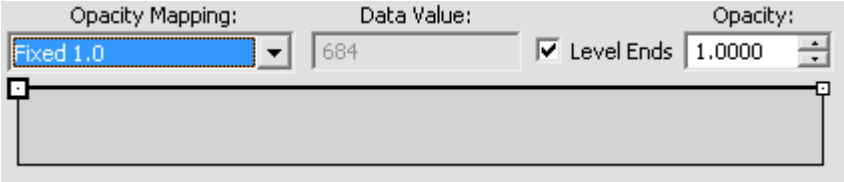
Choose one of predefined opacity maps:

- *Fixed 0.2* sets the entire range of data to a fixed opacity of 0.2.



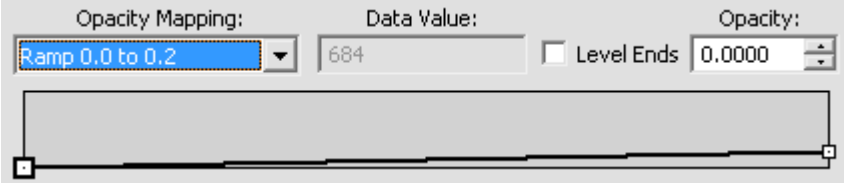
The Opacity Mapping predefined setting of Fixed 0.2 sets the Opacity to a constant 20%.

- *Fixed 1.0* sets the entire range of data to a fixed opacity of 1.0 (fully opaque).



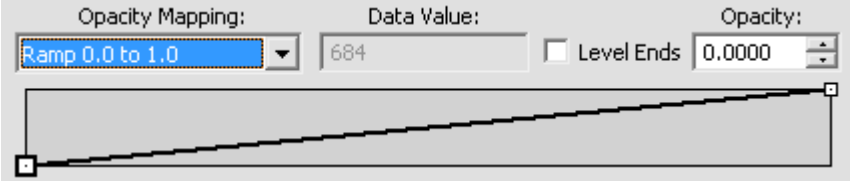
The Opacity Mapping predefined setting of Fixed 1.0 sets the Opacity to a constant 100%.

- *Ramp 0.0 to 0.2* sets the smallest data values to an opacity of 0.0 and the largest data value to an opacity of 0.2. The opacity of the middle values increases linearly.



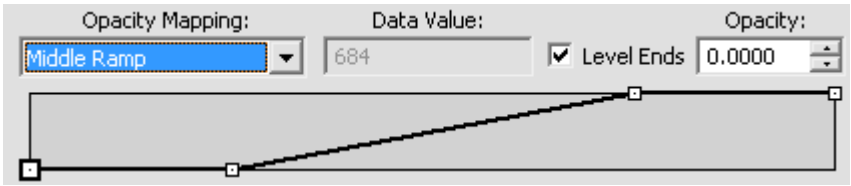
The Opacity Mapping predefined setting of Ramp 0.0 to 0.2 sets the Opacity to go from 0% to 20%.

- *Ramp 0.0 to 1.0* sets the smallest data values to an opacity of 0.0 and the largest data value to an opacity of 1.0. The opacity of the middle values increases linearly.



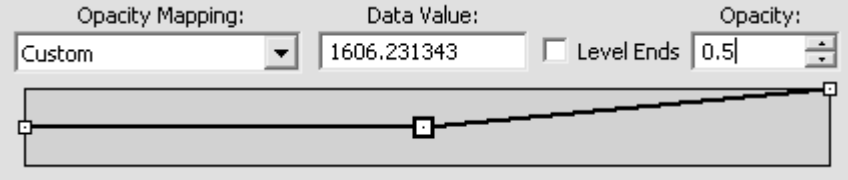
The Opacity Mapping predefined setting of Ramp 0.0 to 1.0 sets the Opacity to go from 0% (fully transparent) to 100% (fully opaque).

- *Middle Ramp* sets the lowest quartile to 0.0 opacity, followed by a linear ramp over the middle half of the data, followed by a fixed opacity of 1.0 for the last quartile.



The Opacity Mapping predefined setting of Middle Ramp sets the Opacity to go from 0% to 100% in the middle of the data range.

- *Custom* is added as an option automatically when a node is selected, created, removed, or the *Opacity* value or *Data Value* options are adjusted.



The Opacity Mapping predefined setting of Custom is created when any adjustments are made to one of the Opacity Mapping predefined settings.

Data Value

Data Value displays the data value of the node selected in the opacity graph. The selected node may be accurately repositioned by entering a new value here. The first and last nodes cannot be changed and this control is disabled (grayed out) when an end node is selected.

Level Ends

Check the *Level Ends* box to level the first and last segments of the opacity graph. When the nodes at one end of a level segment are dragged up or down, the other end is also dragged.

Opacity

Choose the level of opacity. This value ranges from 0.0 (completely transparent) to 1.0 (completely opaque). To change the opacity, click on one of the nodes. Then, highlight the existing value and type a new value or click the button to increase or decrease the opacity for the selected node.

Opacity Graph

The graph consists of a background histogram showing the distribution of data values. A series of two or more nodes (represented by white squares) are connected by line segments. The selected node is drawn as a larger square. Each node is a moveable anchor point whose horizontal position represents a data value, and whose vertical position represents the desired opacity at that data value. The opacity for data values between nodes is calculated by linear interpolation.

Here are some notes about the opacity graph:

- Left-click a node to select it. A selected node is highlighted with a thick black border.
- Use the left and right arrow keys on your keyboard to select the previous or next node.
- New nodes may be added by double-clicking the graph at the desired location or by pressing the INSERT key. When INSERT is pressed, a new node is added adjacent to the selected node, half the distance to the next node.
- Press the DELETE key to delete the selected node. The end nodes cannot be deleted.
- Left-click and drag nodes with the mouse to move them. Nodes cannot be moved past their neighbors. End nodes cannot be moved horizontally. This makes it easier to align two nodes vertically in order to get an abrupt transition.
- Nodes can be moved directly atop one another.
- The histogram uses the logarithm of the counts in order to display widely disparate data. Using the counts directly results in extreme highs and 0 everywhere else for most data sets.

Middle Section of Dialog: Color Mapping Controls

The middle section of the dialog contains the color mapping controls, which are similar to the opacity mapping controls but affect the colors assigned to the data rather than the opacity.

Color Mapping

The color mapping drop-down displays a series of predefined color maps. These may be used to preset the color map which can then be modified as desired.

Data Value

Data Value displays the data value of the node selected in the color graph. The selected node may be accurately repositioned by entering a new value here. Note that the first and last nodes cannot be changed and this control is disabled when an end node is selected.

RGB/ HSL

Click the 

or the 

button to toggle between setting values for RGB (red, green, blue) and HSL (hue, saturation, luminance) modes.

RGB specifies the colors as a combination of red, green, and blue. Each component can range from 0 to 255, e.g., black has an RGB of 0,0,0; white has an RGB of 255,255,255; and red has an RGB of 255,0,0.

HSL mode includes hue, saturation, and light. Each component can range from 0 to 255. Hue represents the dominant wavelength of the color and is often represented as an angle defining the different hues along the circumference of a circle. Saturation indicates how much of the hue is mixed into the color. Luminance is also known as brightness or intensity and represents how much light is in the color.

The three edit boxes to the right allow you to enter the individual color components for the currently selected color node. Alternatively, click on the *Color* drop-down list to pick a color from a list of examples.

Color Graph

The color graph consists of a background histogram showing the distribution of data values. A series of two or more nodes (represented by colored squares) is connected with line segments. The selected node is drawn with a larger frame around it. Each node is a moveable anchor point whose horizontal position represents a data value and whose color represents the desired color at that data value. The color for data values between nodes is calculated by the interpolation method specified in the *Interp* list.

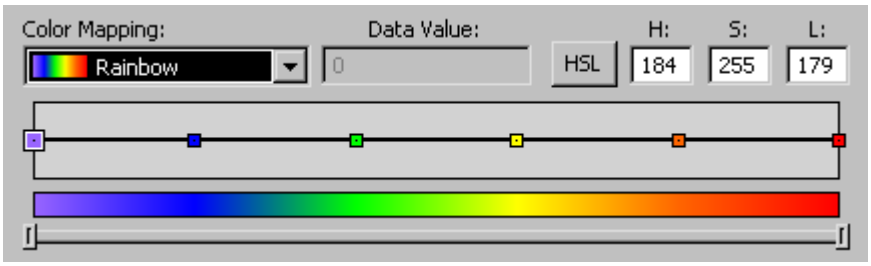
Here are some notes about the color graph:

- The histogram uses the logarithm of the counts in order to display widely disparate data. Using the counts directly results in extreme highs and 0 everywhere else for most data sets.
- Click a node to select it.
- Use the left and right arrow keys to select the previous or next node.
- New nodes may be added by double-clicking the graph at the desired location or by pressing the INSERT key. When INSERT is pressed, a new node is added adjacent to the selected node.
- Press the DELETE key to delete the selected node. Note that end nodes cannot be deleted.
- Drag nodes with the mouse to move them. Note that nodes cannot be moved past their neighbors, and that end nodes cannot be moved horizontally. This makes it easier to align two nodes vertically in order to get an abrupt transition.

- Since the vertical position of nodes within the color graph is irrelevant, coincident nodes are automatically moved up or down so they do not conflict. Note that this is only done for two coincident nodes. A third coincident node obscures the second (and so on). In this case, clicking repeatedly on the coincident nodes cycles through them, selecting each one in turn.

Colormap Preview

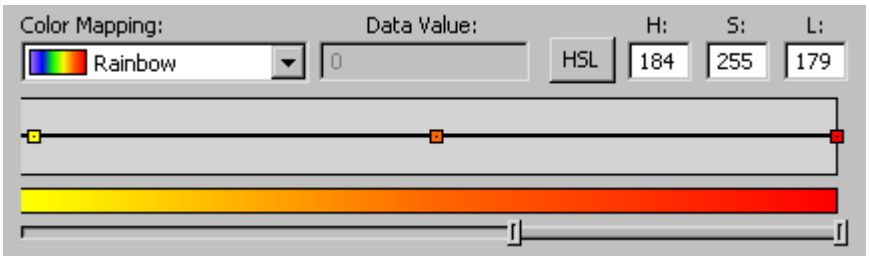
A sample of the generated *Colormap* appears directly below the color graph.



The gradational purple to red Colormap example display appears below the color graph section with nodes for each color.

Scroll Control

The scroll control appears as a three-dimensional horizontal bar with draggable end handles. Drag a handle left or right to zoom the graphs in or out. When zoomed in, the center section becomes blue. Drag the center blue section to scroll the visible portion left or right. Double-click the center section to return it to the fully visible state.



The scroll bar below the Colormap Example allows you to zoom in on a section of the Colormap. In this example, the scroll bar was dragged to zoom in on the yellow, orange, and red nodes.

Bottom Section of Dialog: Colors, Interpolation, and Data Values

The bottom section of the dialog allows selection of data minimum and maximum values, individual color picks, and how the color map should be interpolated. In addition, you can load or save a colormap or close the dialog.

Data Min, Data Max

Colormaps process data with values over a specified range. The *Data Min* and *Data Max* values correspond with the left and right edges of the *Colormap*, respectively. Data outside this range are clipped at the corresponding edge of the *Colormap*. Clipped data are not displayed. The minimum and maximum fields can be set to any desired value. If you wish to reverse the color map, change the *Data Min* value so that it is greater than the *Data Max* value.

Use Data Limits

Check the *Use data limits* box to set the *Data Min* and *Data Max* values to the limits of the input data. This results in the *Colormap* mapping the entire range of the input data without any clipping at the ends.

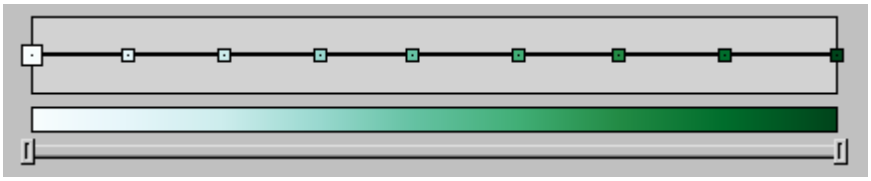
Color

Click the *Color* box to specify the color of the currently selected node. The color palette opens. Click the *More Colors* button to open the **Colors** dialog.

Interp

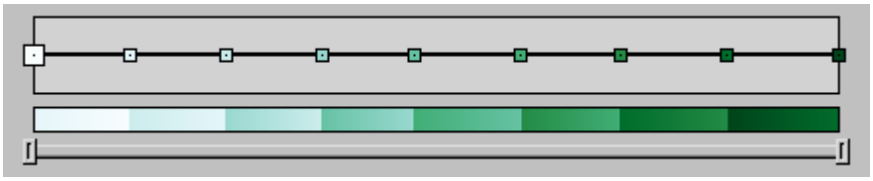
Choose the interpolation method used between nodes in the *Colormap*.

- *Linear* interpolation uses weighted averages to move over nodes from the first color to the last color. This linear interpolation between each color node results in a nice gradational transition and smooth appearance.



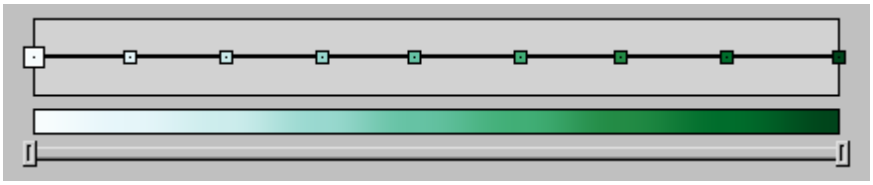
The Linear interpolation results in a smooth transition between nodes.

- *Reverse* interpolation is linear interpolation in reverse and the colors between the nodes are reversed. The colors are linearly interpolated using weighted averages to move over nodes from the first color to the last color. This reverse linear interpolation between each color node results in a segmented appearance.



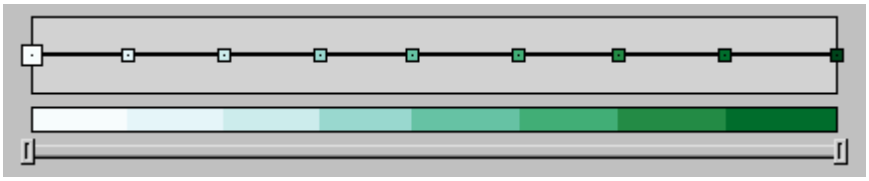
The Reverse interpolation results in sharp transitions between nodes.

- *Cosine* interpolation uses 180 degree segments of the cosine wave. The rate of change of the color is faster than the *Linear* interpolation. This interpolation results in a smooth appearance.

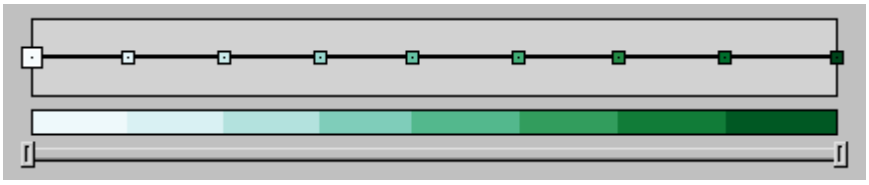


The Cosine interpolation results in a smooth transition between nodes.

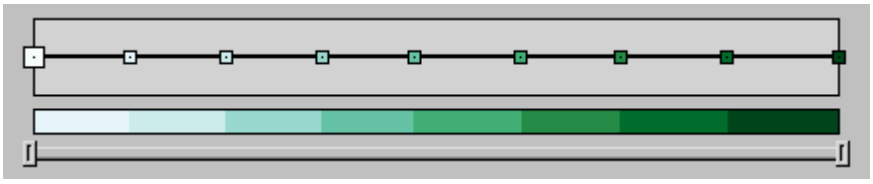
- *Flat Start*, *Flat Middle,* and *Flat End* methods of interpolation fill any gaps between nodes with color.



The FlatStart interpolation results in sharp transitions between nodes.
Flat Start uses the first color node (the node on the left).

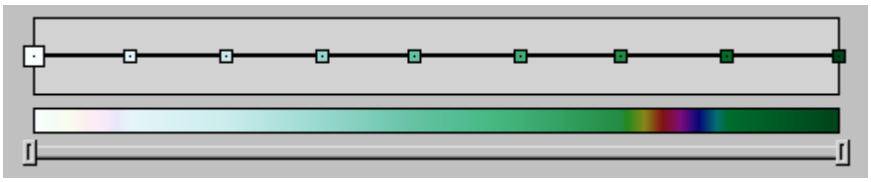


The FlatMiddle interpolation results in sharp transitions between nodes.
Flat Middle uses interpolation to determine a color in the middle of the two nodes.

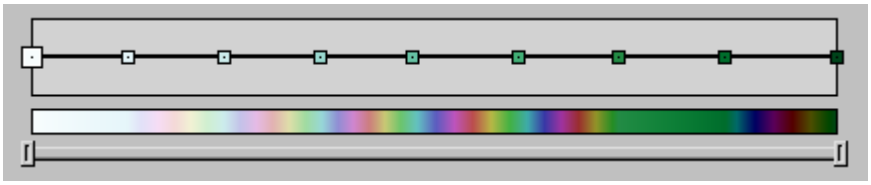


The FlatEnd interpolation results in sharp transitions between nodes.
Flat End uses the last color node (the node on the right).

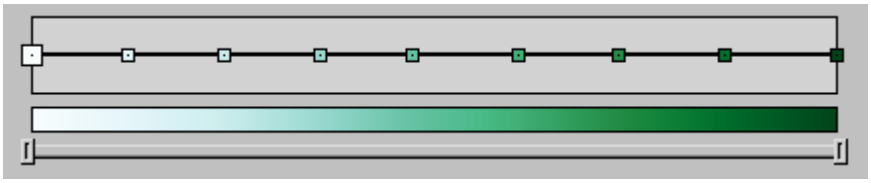
- *HSL Clockwise, HSL Counter-clockwise, HSL Shortest, and HSL Longest* methods of interpolation are based on the HSL color wheel. The H (hue), S (saturation), and L (luminance) values are interpreted linearly from the color wheel.



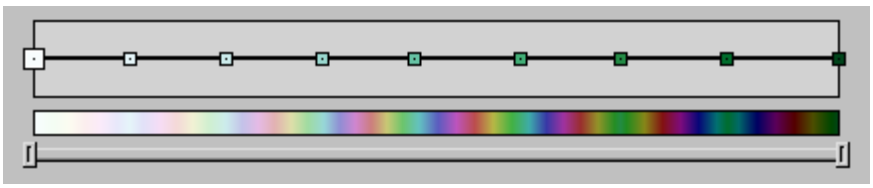
HSL Clockwise interpolates the HSL color wheel in a clockwise direction.



HSL Counter Clockwise interpolates the HSL color wheel
in a counterclockwise direction.



HSL Shortest interpolation is the shortest distance between
Clockwise and Counter Clockwise.



The HSL Longest interpolation is the longest distance between
Clockwise and Counter Clockwise.

Load Colormap

Click the *Load* button to open the **Open** dialog. Select any saved *Colormap File (*.clr)* and click the *Open* button. The Colormap is loaded.

Save Colormap

Click the *Save* button to save the new Colormap properties. The **Save As** dialog opens. The dialog defaults to saving the Colormap to the *Save as type* of *Colormap Files (*.clr)*.

Close

Click the *Close* button to exit the **Colormap Editor** dialog and save your changes.

Undo

Use the **Edit | Undo** command to reverse the changes made before the **Colormap Editor** dialog was opened.

Chapter 22 - Automating Voxler

Introducing Scripter

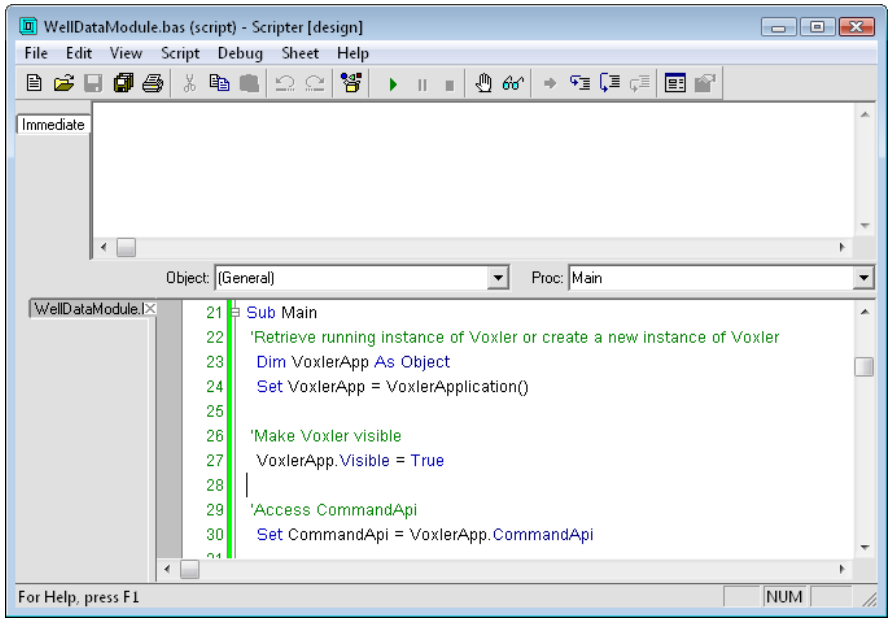
Welcome to **Scripter™**, the easy way to automate **Voxler**. Golden Software's **Scripter** is a program for developing and running scripts. A script is a text file containing a series of instructions carried out when the script is run. Instructions are written in a Visual BASIC-like programming language. **Scripter** offers many features to help you write, edit, and debug scripts. Its features include language syntax coloring, a list of the procedures defined in the script, an object browser for examining procedures available in external objects, a visual dialog editor, break points, single-step execution (including options to step over and to step out of procedures), a watch window for displaying the values of script variables, and more.

Voxler operations can be controlled through automation scripts. You can do almost everything with a script that you can do manually with the mouse or from the keyboard. Scripts are used to automate repetitive tasks, consolidate a complicated sequence of steps, or act as a "front end" to help novice users access **Voxler's** capabilities without having to become familiar with **Voxler**. Since **Voxler** exposes its services through automation, you can use any programming tool that accesses automation objects. Such tools include Visual BASIC, Windows Scripting Host, and many of the Microsoft Office applications, among others.

To start the **Scripter** program, select it from the Windows Start menu. **Scripter** is installed in the same program group as **Voxler**. To open **Scripter**, click the Windows Start button and navigate to **Programs | Golden Software Voxler 4 | Scripter**. If **Scripter** is not present, the installation of **Scripter** may have been skipped when **Voxler** was installed. See the README.RTF file for information about the installation process.

Scripter Windows

When **Scripter** is first started, you are presented with a text editor window containing the lines `Sub Main`, followed by a blank line, and then `End Sub`. This is the code editor window where script instructions are typed and where the contents of script files are displayed.



*The **Scripter** user interface includes several sections.*

Code Window

The code window acts as a text editor, similar to the Windows Notepad program, with a few enhancements to facilitate script writing:

- After you press the ENTER key, tabs or spaces are automatically inserted at the beginning of the next line to maintain the same indentation level as the previous line.
- Key words and symbols of the BASIC language are displayed in different colors. You can use the **View | Colors** command to modify the colors used to display the various elements of the programming language.
- A light horizontal divider line is automatically drawn between sections of your script. The divider lines help you to locate the start of subroutine and function definitions.

Object and Procedure Lists

Above the code editor window is a bar containing the *Object* and *Proc* (procedure) lists. Selecting items from these lists moves the various sections of your script file into view. The object and procedure lists are useful when your script file becomes large.

Immediate Window

Above the object and procedure lists, you may see a blank window area with a tab on top that reads **Immediate**. If this window is not visible, select the **View | Always Split** command to make it appear. The immediate window is used to execute one-line instructions immediately. When you type an instruction into this window and press the ENTER key, **Scripter** carries out the instruction.

In addition to being a scratch area for evaluating language statements, the immediate window shows debugging information. The output from the `Debug.Print` statement and the value of variables selected with the **Debug | Quick Watch** command are printed in the immediate window. While a script program is running, **Watch**, **Stack**, and **Loaded** tabs are added at the top of the immediate window area. Click these tabs for information that may be useful for debugging. See *Debugging Scripts* for more information on the immediate, watch, stack, and loaded windows.

Sheet Tabs

Along the left edge of the code window are code sheet tabs. When you select either the **File | New** command or the **File | Open** command, **Scripter** creates a new code sheet and inserts a new sheet tab. Each tab corresponds to one of the code sheets. Clicking once on a tab makes that sheet the current sheet. Double-clicking a tab closes the sheet.

Break Bar

Between the sheet tabs and the code window is an area called the "break bar." When a script is paused, a yellow arrow in the break bar shows which line is next to execute. The break bar also shows which lines have break points. Set a break point by clicking on the break bar. A red dot appears in the break bar, and the adjacent line in the code window is highlighted. When a line marked as a break point is about to be executed, **Scripter** pauses program execution. To clear a break point, click on the red dot in the break bar. See *Debugging Scripts* for more information on break points.

Status Bar

A status bar along the bottom of the **Scripter** window shows information about the current state of the program. The **View | Status Bar** command hides or reveals the status bar. Before running a

script, make sure that the status bar is visible because messages about typographical and syntax errors are displayed in the status bar.

Three-Minute Tour

We have included several sample files with **Voxler** so that you can quickly see some of **Voxler's** capabilities. The sample files do not include all of **Voxler's** many data types, modules, and features. After opening a sample file, the Network Manager is a good source of information as to what is included in each file. Sample files are located at C:\Program Files\Golden Software\Voxler 4\Samples, by default.

Sample files are a great way to quickly display projects made in **Voxler** by Golden Software. Browse the sample files to get ideas and view different possibilities that **Voxler** has to offer. Sample files can be customized and saved to a new location.

Using Scripter

Tasks can be automated in **Voxler** using Golden Software's **Scripter** program or any ActiveX Automation-compatible client, such as Visual BASIC. A script is a text file containing a series of instructions for execution when the script is run. **Scripter** can be used to perform almost any task in **Voxler**. You can do practically everything with a script that you can do manually with the mouse or from your keyboard. Scripts are useful for automating repetitive tasks and consolidating a sequence of steps. **Scripter** is installed in the same location as **Voxler**. Refer to the *Voxler Automation* help book for more information about **Scripter**. We have included several example scripts so that you can quickly see some of **Scripter's** capabilities.

Sample Script Files

A variety of automation examples are available. You can run the script as is or you can customize the script. Example scripts are located at C:\Program Files\Golden Software\Voxler 4\Samples\Scripts, by default.

To run a sample script in Scripter:

1. Open **Scripter** by navigating to the installation folder, C:\Program Files\Golden Software\Voxler 4\Scripter. Double-click on the Scripter.EXE application file.
2. Click the **File | Open** command and select a sample script .BAS file in the C:\Program Files\Golden Software\Voxler 4\Samples\Scripts folder.
3. Use the **Script | Run** command and the script is executed.

Working with Scripts

You can create new scripts, edit existing scripts, save scripts, and close scripts in **Scripter**.

New Script

To create a new script, select the **File | New** command. A blank script sheet is created. You can start typing script instructions into this sheet. If you edit more than one sheet at a time, click the sheet tabs to switch between them or select the **Sheet | 1**, **Sheet | 2**, etc. menu commands. You can edit up to nine code sheets at the same time.

New Modules

To create a custom ActiveX object, select the **File | New Module** command, and choose either **Object Module** or **Class Module** (choosing **Code Module** is the same as the **File | New** command).

Existing Scripts and Modules

To open an existing script, select the **File | Open** command. To open a script you opened recently, click its name at the bottom of the **File** menu. To open other modules used by your script, select the **Sheet | Open Uses** command.

Saving Scripts

Once a script is complete, you can save the script by using the **File | Save** or **File | Save As** commands. If a script has not been changed since the last save, the **Save** command is disabled.

Closing Scripts

To close the active script, use the **File | Close** command, the **Sheet | Close** command, or double-click the sheet tab of the sheet. Close all open scripts with the **Sheet | Close All** command.

Writing Scripts

To create a script, the script text is typed into the **Scripter** code window or an existing script is edited. When you want to create a new script, you will most likely start with an empty **Scripter** window and type the entire script. If you want to perform a routine task such as creating an isosurface module, you can probably open an existing script file and edit the file to meet your specific needs. **Voxler** comes with several sample scripts that you can modify as desired.

Consider a script that creates an isosurface module:

```
Sub Main
 'Declares VoxlerApp as an object
  Dim VoxlerApp As Object

 'Creates an instance of the Voxler application object
 'and assigns it to the variable named "VoxlerApp"
  Set VoxlerApp = CreateObject("Voxler.Application")

 'Make Voxler visible
  VoxlerApp.Visible = True

 'Access CommandApi
  Set CommandApi = VoxlerApp.CommandApi

 'Create a new Voxler document
  CommandApi.Construct ("New")
  CommandApi.DoOnce()
```

```
 'Load the data file
  CommandApi.Construct("Import")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option ("DefaultPosition", "True")
  CommandApi.Option ("Path", VoxlerApp.Path+"\samples\GoldConcentration.dat")
  CommandApi.Option ("Filter", "dat")
  CommandApi.Option ("Options", "Defaults=1;EatWhitespace=1;Delimiter=Space,_
  tab,comma,semicolon;TextQualifier=doublequote,quote")
  CommandApi.Option ("GuiEnabled", "False")
  CommandApi.Do()

 'Add a gridder module
  CommandApi.Construct("CreateModule")
  CommandApi.Option ("AutoConnect", "True")
  CommandApi.Option ("SourceModule", "GoldConcentration.dat ")
  CommandApi.Option ("Type", "Gridder")
  CommandApi.Do()

 'Grid the data
  CommandApi.Construct("ModifyModule")
  CommandApi.Option("Module","Gridder")
  CommandApi.Option("Gridder3Auto", "true")
  CommandApi.Option("Gridder3DoIt", "true")
  CommandApi.Do()

 'Add a isosurface module
  CommandApi.Construct("CreateModule")
  CommandApi.Option ("AutoConnect", "True")
  CommandApi.Option ("SourceModule", "Gridder")
  CommandApi.Option ("Type", "Isosurface")
  CommandApi.Do()
End Sub
```

When you execute the script, **Voxler** is automatically started and a viewer window is displayed. The data is loaded and the map is created. When the script execution is complete, the **Voxler** window remains open.

Running Scripts

Scripts are placed in the code window by typing a new script from scratch or by loading the script with the **File | Open** command. To run the script in the **Scripter** code window, select the **Script | Run** command, press the F5 key, or click the  button. **Scripter** examines the script instructions, and, if all the instructions are recognized, it begins to perform each instruction in turn.

More often than not, however, a newly typed script does not work correctly the first time it is run. Even simple typographical errors cause scripts to fail. For information on finding and fixing errors in scripts, see the *Debugging Scripts*.

Select the **Script | End** command or click the  button to stop executing a script. This may be necessary when you want to edit a script after a run-time error occurs, or when you accidentally start a script and you want to cancel the execution.

Running Scripts from the Command Line

You can run scripts from a command prompt without having to manually load and execute the script in **Scripter**. The same commands that you would type at a command prompt may also be entered as the "target" for a shortcut in order to link a shortcut button to a script. Enter the following to run a script from the command line or to link a shortcut to particular script file:

```
<Scripter path> -x filename.bas
```

where <Scripter path> represents the path to the **Scripter** program file (for example, C:\ProgramFiles\Golden Software\Voxler 4\Scripter\Scripter.exe") and filename.bas represents the name of the script to run. The space between the –x and the file name is required. This command opens the **Scripter** window, loads the specified script file, and runs the specified script. When the script terminates - either successfully or unsuccessfully - the **Scripter** window closes.

To load a script file but not execute it, the following command can be used:

```
<Scripter path> filename.bas
```

This opens the **Scripter** window and automatically loads the specified script file. The **Scripter** window remains open.

Passing a Command Line Argument to the Script

A single command line argument can be passed to a script. The command line argument can be any text that is used in the script. For example, a file name can be passed and used to create a map. Enter the following to run a script with a command line argument:

<Scripter path> -x filename.bas command

where command represents the commands that should be passed to the script. Access the command line argument from the script using the Command$ option. For example, if the script name is variable.bas and it can be run with any grid file, you could use:

"c:\program files\golden software\Surfer 13\scripter\scripter.exe" -x "c:\temp\variable.bas c:\temp\sample.grd

Do not quote the arguments in the command. To use the data in the script, use:

```
'Return the argument to the Immediate window
Debug.Print Command$
```

```
'Create a contour map from the grid
Plot1.Shapes.AddContourMap(GridFileName:=Command$)
```

Debugging Scripts

Bugs are errors in a script that keep it from performing as intended. Debugging is the process of locating and fixing these errors. The most common bugs are typographical errors in scripts or malformed instructions. **Scripter** detects these types of errors immediately when you try to run a script. The program beeps, highlights the line containing the error in red (or whatever color has been set with **View | Colors** command), and displays an error message on the status bar.

Viewing Errors

Before running a script, verify that the **View | Status Bar** command is enabled, or you will not see the error message. To resolve the errors that **Scripter** immediately detects, you usually must interpret the error message and correct the indicated problem. Typical errors are typing mistakes, unbalanced parentheses, misuse of a BASIC language instruction, or failure to declare variables in a DIM statement (if you use the OPTION EXPLICIT statement). If you do not see an obvious problem, refer to the online BASIC language help to make sure you are using the right syntax.

Run-Time Errors

Scripts that encounter errors midway through script execution may be fixed much the same way as syntax errors. The error message should guide your actions. Some run-time errors cannot be detected until they are executed, such as when you try to open a file that does not exist. In these cases, you need to check for error conditions in your scripts. For example, use the DIR function to make sure a file exists before trying to open it. Alternatively, you can use the ON ERROR GOTO statement to specify an error handling section to execute when a procedure encounters a run-time error:

```
Sub OpenFile(CommandApi As Object, filename As String)
  On Error Goto ErrLabel
     'Open an existing file
      CommandApi.Construct ("Open")
      CommandApi.Option ("Path", filename)
      CommandApi.DoOnce()
Exit Sub

ErrLabel:
MsgBox "Unable to open file " & filename
Exit ' Must use RESUME or EXIT at end of error handling code
End Sub
```

Script Runs Incorrectly

Most difficult to correct are scripts which run but do not work as expected. Fixing these scripts is hard because you do not know which line or statement is causing the problem. **Scripter** provides a number of debugging features to help you locate the source of problems.

Debug.Print

Probably the simplest debugging technique is to insert instructions into your script to monitor the progress of the script and display the values of variables at various points throughout the script. Use the Debug.Print statement to display information in the **Scripter** immediate window:

```
Debug.Print "The value of variable X is "; X
```

To clear the contents of the immediate window, select the text in the window and press either DEL or BACKSPACE.

Stop or Pause

Insert the STOP instruction to pause script execution where you think there might be a problem. While the script is paused, you can examine and change the values of program variables. If a running script appears unresponsive, it may be stuck in an infinite loop. Select the **Script | Pause** command or click the  button to pause the script. To resume executing a paused script, select the **Script | Run** command or click .

Viewing Variable Values

While a script is paused, there are several ways to view the value of a variable:

- In the immediate window, type a question mark followed by the variable name and press ENTER. The current value of the variable is displayed.
- In the code window, place the cursor on the variable name you want to examine (that is, click on the variable name in the code window). Press SHIFT+F9, select the **Debug | Quick Watch** command, or click the  button on the toolbar. The current value of the variable is displayed in the immediate window.
- To continuously monitor a variable's value, click on the variable name in the code window, and press CTRL+F9 or select the **Debug | Add Watch** command. Alternatively, type the variable name in the watch window and press ENTER. The variable name and its value are displayed in the watch window. Every time script execution pauses, the variable value is automatically updated. To clear a variable from the watch window, highlight the line showing the variable value and press the DEL or BACKSPACE key.

Changing Variable Values

To change the value of a variable, type an assignment expression in the immediate window and press ENTER. For example, type "A=5" (without quotes) and press ENTER to assign a new value to the variable named "A."

Step

A powerful debugging technique is to watch **Scripter** execute your script one line at a time. This lets you check the effect of each instruction and verify that the script is doing what you expect. While stepping through a script, you can examine and change the values of script variables. Select the **Script | Run** command or click  to resume script execution at full speed after stepping through script instructions.

To execute your script one line at a time press the F8 key or select the **Debug | Step Into** command. The first line of the script is executed (or, if the script was paused, the next highlighted line is executed). The next line is highlighted and a yellow arrow appears to the left of the next line. To execute the highlighted instruction, press F8 again.

If a statement calls a subroutine or function that is defined within your script, the highlight will move into the called procedure. To keep from tracing execution into a called procedure, press

SHIFT+F8 or select the **Debug | Step Over** command. This executes the whole subroutine or function in a single step.

If you accidentally step into a procedure, press CTRL+F8 or select the **Debug | Step Out** command. This executes all remaining instructions in a procedure, and returns the highlight to the instruction that called the procedure.

If you do not see the next highlighted instruction, select the **Debug | Show Next Statement** command to scroll the highlighted line into view.

Sometimes you may want to skip the execution of some instructions or you may want to execute the same instructions several times without restarting the script. To change the next instruction line, click on the line you want to execute next and select the **Debug | Set Next Statement** command.

Breakpoint

Watching **Scripter** execute every line of the script may be too time consuming. In this case, a breakpoint pauses the script where you think there might be a problem. A breakpoint is a line of code that you mark. When **Scripter** encounters a line marked as a breakpoint, it pauses the script just as if it had executed a STOP instruction. Breakpoints are more convenient than STOP instructions because they may be set and cleared while a script is paused, whereas STOP instructions may be changed only after a script has ended.

To set a breakpoint, click in the break bar area next to the line you want to mark. The break bar is the area to the left of the code window, between the sheet tabs and the code window. Alternatively, click on the line you want to mark, and press F9 or select the **Debug | Toggle Break** command. The line becomes highlighted in red, and a round marker appears in the break bar area.

To clear a breakpoint, click on the round marker, or move the cursor to the marked line and press F9 or select the **Debug | Toggle Break** command again. You can clear all breakpoints by pressing SHIFT+CTRL+F9 or selecting the **Debug | Clear All Breaks** command.

A quick alternative to setting a breakpoint is the **Debug | Step To Cursor** command. This command has the same effect as setting a breakpoint on the current line, running the script, and then clearing the breakpoint after script execution has paused on the current line.

Trace

To check flow of execution through your script without having to watch each line of the script being executed, try using the TRACE function. To activate the trace function type "Trace" (without the quotes) in the immediate window and press ENTER. Trace On is displayed in the immediate window. As the script is run, the location of every instruction being executed is printed in the immediate window. After the script finishes, the trace function is automatically disabled.

Stack

If you nest procedure calls (that is, one procedure calls another procedure, which calls yet another procedure, and so forth), the stack window may be useful. When a script is paused, the stack window lists the procedures that have been called, and the order in which they were called. For instance, if the Main procedure calls procedure "A" which in turn calls procedure "B," the stack window displays three lines, one for each of the called procedures. Clicking on a line in the stack window moves the corresponding procedure into view in the code window.

Module Files

Click the loaded window tab in the immediate window area to see which module files are currently being interpreted by **Scripter**. The loaded files include the current script file and any modules it includes with the `'#Uses` statement.

Type Library References

Many application programs expose their services with objects. These objects may be used in your scripts, just as the **Voxler** automation objects are used. Before you can use another application's objects, you must add a reference the application's type library. A type library is a file that describes the objects, properties, and methods provided by an application. The **Voxler** type library is automatically referenced in **Scripter**.

To add a type library reference to the current script module in **Scripter**:

1. Select the **Edit | References** command to open the **References** dialog.
2. Click on the check box next to the type library whose objects you want to use in your script.
3. Click on the up and down arrows to adjust the relative priority of the checked references. If two libraries describe items with the same name, the item described in the higher-listed library is used by your script.
4. Click the *OK* button when finished adding type library references to your script.

Scripter BASIC Language

The online help describes the major elements of the **Scripter** BASIC programming language, but it does not explain the concepts of writing computer programs. Many good books on the subject of programming with BASIC (**B**eginner's **A**ll-purpose **S**ymbolic **I**nstruction **C**ode) have been written. If you are not moderately familiar with writing computer programs, we suggest that you refer to one of the books listed in the *Suggested Reading* topic.

Scripts are text files that contain a sequence of instructions to be carried out by the **Scripter** program. Each instruction specifies a task such as defining a variable or displaying a message on the screen. When the **Scripter** program processes the script, the instructions are carried out one at a time, from top to bottom.

Execution of a script begins with the first statement of the subroutine called `Main`. All scripts must therefore include the `Sub Main` and `End Sub` statements. Execution proceeds line-by-line until the end of the `Main` procedure, until an `End` statement is executed, or until an error occurs.

Program Statements

Statements are individual instructions to **Scripter** that carry out a specific operation. Statements are case insensitive and are typically written one to a line. To enter two or more statements on the same line, separate the statements with colons. For example:

```
a = 5 : b = 5 * a
```

Scripter BASIC requires flow control statements (IF, WHILE, DO, etc.) and declaration statements (DIM, PUBLIC, TYPE, SUB, etc) to be placed on a line by themselves.

Line Continuation

To break a line into two lines in **Scripter**, use a space followed by an underscore " _". You must include the space for the continuation to work properly in the script. A backslash continuation " \ " from earlier versions of **Scripter** is not supported. Comments are not allowed after the continuation character.

Example

```
Function ComputeSomething( filename As String, _
value_array() As Double ) As Double
```

Comments

Writing comments in your scripts to explain how they work can save you time and frustration when you later need to modify the script. The apostrophe character (') signals the start of a comment. **Scripter** ignores all text following the apostrophe up to the end of the line. Comments can be placed on their own line, or they may be placed at the end of a line.

Example 1

```
  VoxlerApp.Visible = True       'Make Voxler visible
```

Example 2

```
'Make Voxler visible
  VoxlerApp.Visible = True
```

In addition, you can use the REM statement to add a remark in the script. However, REM statements can only be used at the beginning of a line.

Double Quotes and Text

In **Scripter**, text strings must be enclosed in double quotes. File names, for example, must be surrounded by double quotes. If quotes are missing, the text may be mistaken for a variable name.

```
Debug.Print "This text string is printed in Scripter's immediate window"
```

Operators

Operators are symbols that direct a script to perform basic calculations, such as addition, exponentiation, string concatenation, number comparison, and others. The language supports several arithmetic, comparison, and logical operators. In **Scripter**, choose the **Help | BASIC Language Help** command and search for "Operators" to see a complete list.

Flow Control

When you run a script, execution starts with the `Sub Main` statement and continues line-by-line until the `End Sub` statement at the end of the main procedure or until an `End` statement is encountered. Several flow control statements allow you to change this line-by-line progression according to conditions encountered by your script. The **Scripter** BASIC language includes a variety of looping and branching statements that is typical for modern programming languages. The flow

control statements include the following (see the online BASIC language help for details on the syntax of these statements):

IF…END IF

`IF...END IF` executes a statement only if a condition if true.

IF…END IF

The alternate `IF…ELSE…END IF` form executes one statement if a condition is true and a different statement if the condition is false.

SELECT CASE…END SELECT

`SELECT CASE...END SELECT` branches to one of several statements. This compares the value of an expression to several test values and executes the statements associated with the test value that matches the expression value.

DO…LOOP

`DO...LOOP` is the basic looping statement. This statement loops either while a condition is true or until a condition becomes true and tests the condition either at the top of the loop or at the bottom of the loop.

This and all other loop structures may be stopped before the test condition has been met by placing an `Exit` statement in the body of the loop.

WHILE…WEND

`WHILE...WEND` loops while a condition is true and tests the condition at the top of the loop.

FOR…NEXT

`FOR...NEXT` loops a number of times and increments (or decrements) an index variable each time through the loop.

FOR EACH…NEXT

`FOR EACH...NEXT` iterates through all the elements in a collection object. The `For...Each` statement is a convenient way to process each element in a collection.

Optional Arguments

Many procedures accept a large number of arguments. Some of the arguments are required. Every required argument must be supplied or the script fails to run. Some arguments are optional. Optional arguments may be omitted and the procedure assumes a default value for the missing arguments.

For example, the *ConnectModules* option of the *Construct* method accepts six parameters (`SourceModule, SourceModuleID, SourcePort, TargetModule, TargetModuleID, TargetPort`), some of which are optional. This section of code shows the command with only the required options:

```
CommandApi.Construct ("ConnectModules")
CommandApi.Option("SourceModule", "GoldConcentration.dat")
CommandApi.Option("TargetModule", "Gridder")
CommandApi.Do()
```

Since the arguments are optional, you can skip all or some of them when calling the command.

Subroutines and Functions

Writing a long or complicated script may be easier to manage if you divide the script into smaller pieces called procedures. A procedure is a separate sequence of instructions that you can call from multiple places within your script. The BASIC language provides many pre-defined procedures for performing frequently needed tasks, and, in fact, the methods provided by the **Voxler** automation objects are themselves procedures.

When you call a procedure, the instructions in the procedure are executed. When the procedure finishes its task, it returns control to the instruction that called the procedure. The **Scripter** BASIC language distinguishes two types of procedures: functions and subroutines. Functions return a value whereas subroutines do not.

Subroutines and functions may accept one or more values, called arguments. Arguments are passed to a procedure by listing them after the procedure name. If there is more than one argument, the arguments must be separated by commas. For example:

```
'Returns the cosine of the argument (returns 1)
 x = Cos(0)

'Returns the left-most characters (returns " Ag")
 a$ = Left(" AgAuPb",2)

'Waits for 5 seconds
 Wait 5
```

`Cos`, `Left`, and `Wait` are procedures built-in to the BASIC language. `Cos` and `Left` are functions which return values to the caller. `Wait` is a subroutine, and, by definition, it does not return a value. The `Wait` subroutine waits for the number of seconds specified by its argument (5 seconds in this example) before returning control to the calling location.

The arguments passed to a function must be enclosed in parentheses if the function's return value is used. If the function's return value is not used, the arguments may be listed without enclosing them in parentheses. Arguments passed to a subroutine are never enclosed in parentheses.

Writing Subroutines

To define subroutines within a script, use the `Sub` statement. Subroutine and function definitions cannot be nested within other procedures. That is, the `Sub` statement must appear after the `End Sub` statement of any preceding subroutine definitions. The syntax for a subroutine definition is:

```
Sub name ( arguments )
    statements
End Sub
```

where `name` represents the name you want to give to the subroutine, `arguments` represents a list of arguments names and types, and `statements` represents the instructions that comprise the body of

the subroutine. There is no limit to the number of instructions you can include between the Sub and the End Sub lines. Consider the definition of a Main procedure and another subroutine:

```
Sub Main
   MultipleBeep 25   'call the MultipleBeep subroutine
End Sub

Sub MultipleBeep (count As Integer)
   For i = 1 To count
     Beep
     Wait 0.5 'Wait one-half second between beeps
   Next
End Sub
```

Each time the MultipleBeep procedure is called, the instructions between its Sub and End Sub statements are executed.

If the subroutine accepts arguments, the arguments are defined following the subroutine name using a format similar to the Dim statement. The argument definition list must be enclosed in parentheses, and argument definitions are separated by commas if there is more than one. When a subroutine is called, the variables listed in the argument list are automatically assigned the values passed in from the calling procedure.

Writing Functions

Functions are defined using the Function statement much the same as subroutines are defined with the Sub statement. Like subroutines, function definitions cannot be nested within other procedures. Unlike subroutines, functions can return a value to the calling procedure. The syntax of a function definition is:

```
Function name ( arguments ) As type
    statements
End Function
```

where name is the function name you want to use, arguments is a list of arguments names and types, type is the type of the value returned by the function, and statements are the instructions in the body of the function. To return a value from a function, assign a value to a variable with the same name as the function itself. For example:

```
Function hypotenuse(a As Double, b As Double) As Double
   'The built-in Sqr function computes the square root
    c = a * a + b * b

   'Set the function's return value
   hypotenuse = Sqr(c)
End Function
```

The list of arguments accepted by a function is defined the same way as you define the arguments accepted by subroutines.

Built-in Functions and Procedures

Numerous useful functions and subroutines are built into the **Scripter** BASIC language. These routines can help you perform some of the most commonly required programming tasks. Functions

for processing strings, performing mathematical calculations, error handling, working with disk files, and many others are available.

If you are not already familiar with the Visual BASIC for Applications programming language, it will be worth your time to review the list of available routines. This list is found by selecting **Help | BASIC Language Help** in **Scripter**.

Variables

In **Scripter**, a variable is a symbolic name for a value. A variable name starts with a letter and may contain digits. Variable names cannot be the same as a reserved word. Because the **Scripter** code window displays variable names in black and reserved words in color, you can see when you have selected a variable name that conflicts with a reserved word.

Variables may be one of several types. The type of a variable determines what kind of data it may contain. See the following table for the possible variable types. In addition to the built-in data types, the **Scripter** language supports user-defined compound data types, user-defined enumeration types, and user-defined objects (defined in object modules and class modules).

The type of a variable is declared in a `DIM` statement. The syntax of a `DIM` statement is:

```
Dim varname As type
```

where varname is the name of the variable being declared and type is the variable's data type. Variables not declared in a DIM statement are a variant type, unless the variable name ends with one of the type-definition characters. If a variable name ends with one of the special type-definition characters, listed below, its type is recognized based on this character.

Type	Type-Definition Character	Description of Type
Integer	%	A 16-bit integer value
PortInt (Portable Integer)	?	A 16- or 32-bit integer value
Long	&	A 32-bit integer value
Single	!	A 32-bit floating-point value
Double	#	A 64-bit floating-point value
Currency	@	A 64-bit fixed-point value
String	$	A text string of any length
Byte	(none)	An 8-bit unsigned integer value
Boolean	(none)	A true or false value
Date	(none)	A 64-bit floating-point value
Object	(none)	A reference to an object

Variant	(none)	Capable of holding any type of value

Using the DIM statement to declare the variable type is optional. Variables can be used without first being declared in a DIM statement, but this practice is not recommended for any script longer than a few dozen lines. To enforce this policy, an OPTION EXPLICIT statement should be placed at the top of long scripts. The OPTION EXPLICIT statement makes it an error to use any variable without first declaring it. Using this option lets you find typographical errors in variable names before a script is run. Without this option, typographical errors in variable names are usually detected only when the script fails to produce the expected results.

Object Variables

In **Scripter**, object variables contain references to ActiveX objects. Creating the program Application object is an example of declaring an object variable:

```
Dim VoxlerApp As Object
Set VoxlerApp = CreateObject("Voxler.Application")
```

In this example, a DIM statement declares that the variable named VoxlerApp holds a reference to an object. The built-in CreateObject function returns a reference to a **Voxler** Application object, and the SET statement assigns this object reference to the VoxlerApp variable. Unlike variables of other types, which can be assigned new values simply with an equal sign (=), object variables must be assigned values with a SET statement.

Array Variables

Array variables store a list or table of values. A single variable name refers to the entire collection, and individual values are distinguished by their numeric indices (their "subscripts"). The maximum number of values that can be stored in an array must be defined in a Dim statement. The elements of an array are accessed by using the variable name followed by a left parenthesis, the index of an array element, and a right parenthesis.

```
Dim month(11) As String
month(0) = "January"
month(1) = "February"
…
month(11) = "December"
```

Array subscripts begin with zero, unless an Option Base statement is used at the start of a script. Notice that in the previous example an array whose maximum subscript value is 11 actually has room for twelve elements because the subscripts start with zero.

The Dim statement can reserve only a constant number of elements for an array. If the maximum number of elements cannot be known in advance, a dynamic array may be used. A dynamic array is an array whose number of elements can be changed while a script is running. The Redim statement changes the maximum number of values that can be stored in a dynamic array. Refer to **Help | BASIC Language Help** in **Scripter** for more information on Dim and ReDim.

User-Defined Types

A collection of related variables can be grouped together under one name. The TYPE statement defines the elements of a user-defined type.

Type measurement
julianday As Integer
level As Double
End Type

The TYPE definitions must appear at the top of a script file, before any subroutines. The TYPE…END TYPE statement defines a new type; it does not create a variable of that type. Variables of the user-defined type must be declared in a Dim statement. The elements of a user-defined type variable are accessed by using the variable name followed by a period and the element name:

```
Dim m As measurement
m.julianday = 192
m.level = 12.3
Debug.Print m.julianday ' prints 192 in the Immediate window
Debug.Print m.level ' prints 12.3 in the Immediate window
```

Global Variables

In **Scripter**, variables declared in the body of a subroutine or function are available only within that procedure. If you want to share the same variable throughout a script file, then you can define it at the top of the file, before any subroutine definitions. Variables declared at the top of the file are available to all subroutines in the file; hence, they are called "global" variables.

The PUBLIC keyword may be substituted for the Dim keyword to allow a global variable to be used in other modules.

Module Types

If you create very large scripts, or frequently reuse the same subroutines in several different scripts, you may want to split your script code into more than one file. Each script file is called a module.

A script can call subroutines and functions defined in other modules. In addition to procedures, global variables, type definitions, and enumeration definitions may be shared among modules. Just as procedures make long scripts easier to manage and debug, modules make large script projects easier to manage.

The **File | New Module** command in **Scripter** adds new code sheets to the workspace. Each sheet is stored in a separate file. When routines in one code sheet are used by other sheets, the code sheets are called modules. **Scripter** supports three types of modules:

- Code modules are used for stand-alone scripts and for storing libraries of useful procedures that can be called from other modules. The scripts described in this chapter are code modules, which contain a Main subroutine. Code modules without a Main subroutine cannot be run, but the routines contained in them can be shared by other scripts. Code modules are stored in files with a .BAS extension.
- Class modules are used to define objects that you can use in other modules. A class module defines the properties and methods that the object supports. Other modules access the object's

properties and methods using the same syntax that is used to access **Voxler** automation objects. Unlike **Voxler** objects, new instances of the object defined in a class module are created using the NEW keyword. Class modules are stored in files with a . CLS extension.

- Object modules are identical to class modules, except that when a script uses the object defined in an object module, one instance of the object is automatically created. Additional instances of an object defined in an object module can be created with the NEW keyword. Object modules are stored in files with an . OBM extension.

The '#Uses Line

Before using the procedures and objects defined in another module, a script must indicate the name of the file containing the procedure or object definitions. You must place a `'#Uses` statement at the beginning of a script, before any procedure definitions, to instruct **Scripter** to load all modules used by the script.

Example

```
'#Uses " c:\ utils.bas"
'#Uses " test.cls"
Sub Main
   'use the procedures and object defined in UTILS.BAS and TEST.CLS
End Sub
```

Scripter does not permit cyclic `'#Uses` statements. That is, if module A uses module B, module B cannot use procedures from module A.

Private and Public Definitions

By default, all subroutines, functions, and user-defined types (including enumeration definitions) may be accessed from other modules. To prevent other modules from accessing procedures or user-defined types precede the definition with the `Private` keyword:

```
Private Sub MyBeep
   Beep : Beep
End Sub
```

In contrast to procedures, the global variables defined at the top of one module are not available to other modules unless they are declared using a `Public` statement. When used for variable declarations, the Public statement has the same syntax as the `Dim` statement:

```
Public universal_data As String
```

The names of all definitions, even private ones, are visible in other modules. To avoid errors due to name conflicts you must avoid using the same procedure, type, and variable names in more than one module. A common technique for avoiding name conflicts is to append a prefix to the names of global variables and procedures. For example, if you write a module of text-processing functions, you might prefix each function name with txt, e.g., `txtFunction1`.

Module Properties

To set the name by which other modules refer to an object defined in a class or object module, select the **Edit | Properties** command in **Scripter**. The **Edit Class Module Properties** dialog appears. Type the name that you want other scripts to use when referring to the object defined in the module. The instancing options control how other applications access the object defined in the module, but these options are not relevant to scripts executed within **Scripter**. Code modules do not have module properties.

When an object module is used in a script, one instance of the object defined in the module is automatically created. The name of the object that is automatically created is the name specified in the **Edit Object Module Properties** dialog.

Defining Object Properties and Methods

Class and object modules define the properties and methods of objects. To define the methods for an object, simply define public subroutines and functions. All the public procedures in a class or object module are methods that users of the object can call.

The properties of an object typically correspond to private global variables defined in the module. To allow users of the object to access the variable values, you provide "property get" and "property set" procedures. Use the Property Get statement to define a function that returns the value of a property. Use the `Property Let` statement (or the `Property Set` statement if the property is an object reference) to define a subroutine that changes the value of a property.

Two special subroutines are called when an object is first created and just before it is finally destroyed. In a class module, these subroutines are called `Class_Initialize` and `Class_Terminate`. In an object module, these subroutines are called `Object_Initialize` and `Object_Terminate`. These subroutines do not take any arguments.

Class Module Example

The following class module demonstrates how to define an object. The sample defines a property named *Radius* and a method named *Draw*.

```
'Declare a private global variable for
'storing the property called "Radius"
 Dim cirRadius As Double

'Define the initialization subroutine
 Private Sub Class_Initialize
   cirRadius = 99
End Sub

'Define the termination subroutine
Private Sub Class_Terminate
End Sub

'Define the "property get" function to
'retrieve the Radius property
Property Get Radius() As Double
   Radius = cirRadius
End Property

'Define the "property let" procedure to
'change the Radius value
Property Let Radius(val As Double)
   cirRadius = val
End Property

Sub Draw
   'Method performs some action here
End Sub
```

Creating Dialogs

Scripter contains a dialog editor that you can use to design customized dialogs for use with automation. Select the **Edit | UserDialog Editor** command to visually design a dialog. You can control the size and placement of the components of the dialog, as well as customize the text included in the dialog.

Adding Items to the Dialog

To add a component to a dialog, first select from the palette of components at the left side of the dialog editor. After clicking a palette button, drag the mouse pointer diagonally in the dialog design area where you want to place the component. As you design the dialog, you can edit the properties of components you have placed in the dialog. To edit the properties of a component, double-click the item, click the right mouse button on the item, or select the component and click the  button. Every dialog must include an `OKButton` or a `CancelButton`, or both.

Saving a Dialog

When you have finished designing the dialog, click the  button. The code for the dialog is inserted into the script.

Editing a Dialog

To edit the dialog template after it has been inserted into the script, first move the cursor in the code window to any line between the `BEGIN DIALOG` statement and the `END DIALOG` statement. Next, select the **Edit | UserDialog Editor** command. The previously saved state of the dialog is shown in the dialog editor. When you save the dialog again, the previous dialog template is replaced with your changes.

Displaying the Dialog

To show your custom dialog in a script, first use the DIM statement to declare a variable as the `UserDialog` type, and then call the `DIALOG` function to display the dialog (see the example). The `DIALOG` function takes a user dialog variable as its argument and returns a number indicating which button was clicked to end the dialog. The `DIALOG` function returns -1 if the OK button was clicked, 0 if the cancel button was clicked, or an integer greater than zero if a push button was clicked (1 for the first push button listed in the dialog template, 2 for the second push button in the dialog template, and so forth).

If the return value is not needed, the `DIALOG` instruction may be called as a subroutine rather than as a function. In this case, do not enclose the dialog variable in parentheses. If the `DIALOG` instruction is called as a subroutine, however, the script ends with a run-time error if a cancel button is clicked.

Creating Multiple Custom Dialogs

To define more than one custom dialog in a script, you must place each dialog template in its own subroutine or function. If you try to define more than one custom dialog in the same procedure, **Scripter** shows an error indicating that the UserDialog type has already been defined.

Values in Dialogs

The values contained by dialog controls are accessed the same the way the fields of user-defined variable types are accessed. Type the dialog variable name, followed by a period, followed by the

field name of the dialog component. Option button values cannot be accessed directly, but are accessed via the field name of their associated option group. The value of an `OptionGroup` is the number of the selected option button (the first option button in the group is 0, the second option button is 1, and so forth). You can initialize the values contained by dialog controls prior to showing the dialog, and retrieve the values entered in the dialog after it has been invoked.

UserDialog Items

Item	Description	Properties
GroupBox	A rectangle used to group related controls	Field, Caption
Text	A text label, requires no response from the user	Field, Caption
TextBox	An edit box used to enter and edit text	Field, Type (single line, multiple line, password)
CheckBox	A box which is checked and unchecked as the user clicks on it, a three-state check box has a disabled (grayed) state	Field, Caption, Type (2 state, 3 state, 3 state auto check)
OptionButton	A round button for choosing from a set of options, only one of a group of option buttons may be checked	Field, Caption, Option Group
ListBox	A window that contains a list of items that can be selected by the user	Field, Array of Items, Type (list, sorted list)
DropListBox	A list that is visible when opened by the user, the text may be editable or not and the list may be sorted or not	Field, Array of Items, Type (list, text, sorted list, sorted text)
ComboBox	A text box with an attached list box, the list may be sorted or not	Field, Array of Items, Type (text, sorted)
Picture	Displays a bitmap in the dialog	Field, Caption, Type (from file, from clipboard)
PushButton	A push button	Field, Caption
OKButton	Push button with the OK caption	Field
CancelButton	Push button with Cancel caption	Field
Dialog	Definition of dialog box	Dialog Function, Caption, Centered

Properties:

Dialog Function

The dialog function is the name of a special function that is called when various events happen in a dialog. Define a dialog function to control the behavior of a dialog and to retrieve its input.

Field Name

The field name is the name used to refer to a component.

Caption

The caption is the text displayed within a component.

Quoted

When not quoted, the caption property gives the name of variable that contains the text to display for the caption. When quoted, the caption property is the literal text to display.

Array of Items

The array of items is the name of a string array variable that contains the strings to display in a list. The array variable must be initialized before the dialog is invoked.

Type

The type is the behavior of some components varies depending on which option is selected. Refer to the online help for descriptions of the available component types.

Option Group

For option buttons, the field name is used to refer to a group of option buttons. Only one option button within a group may be checked.

Comment

A comment is the text to insert in the dialog definition block.

UserDialog Example

The following function demonstrates how to define, display, and extract the values entered in a user dialog.

```
 Function MyInputBox As String
'Define the dialog template. This definition
'is inserted by the UserDialog editor.
 Begin Dialog UserDialog 250,112,"Caption"
  TextBox 10,14,230,28,.Text1
  CheckBox 20,49,160,14,"Check Box",.Check1
 OKButton 20,77,90,21
 CancelButton 130,77,90,21
 End Dialog

'Declare a dialog variable
 Dim dlgvar As UserDialog

'Initialize the dialog controls
 dlgvar.Text1 = "This is the initial text to display"
 dlgvar.Check1 = True ' start with check box checked

'Display the dialog and wait for the OK or Cancel
'button to be pressed
 result = Dialog(dlgvar)

'Extract the information entered into the dialog
 If result = -1 Then ' check to see if OK button was clicked
```

```
 MyInputBox = dlgvar.Text1
If dlgvar.Check1 Then Debug.Print "The Check Box was Checked!"
End If
End Function
```

To perform processing while a user dialog is active, define a special "dialog function." The dialog function is called when various dialog events occur. To define a dialog function:

1. While designing the dialog, double-click in a blank portion of the dialog design area to activate the **Edit UserDialog Properties** dialog.

2. Enter a name for the Dialog Function property of the dialog. This property gives the name of a function that is called when dialog events occur.

3. When you save the dialog, **Scripter** asks you if it should create a skeleton dialog function. Click the *Yes* button, and **Scripter** inserts the basic instructions for a dialog function into your script.

Refer to the DialogFunc help topic in the **Help | BASIC Language Help** for more information about how to process dialog events in a dialog function.

Using Scripter Help

For information on **Scripter** program menu commands, select the **Help | Contents** command in **Scripter**. Press the F1 key for information about the **Scripter** windows or the active dialog. The **Help | Voxler Automation Help** command shows all **Voxler**-specific methods and properties.

The online help, shown when you select the **Help | BASIC Language Help** command (or press SHIFT+F1), explains all of the BASIC language statements and functions. Each help topic describes the purpose of a statement, and shows the syntax (the order of keywords and other symbols) to use when writing an instruction. The syntax examples in the online help use a shorthand method to explain the possible variations in usage:

Sub, End, True	Words with the initial letter capitalized indicate language-specific keywords.
name	Lower-case words with an underline (dotted in **Scripter** help) are placeholders for information you supply.
param	Items inside square brackets are optional. These may be omitted from the statement.
{Until | While}	Items between curly braces, and separated by a vertical bar are lists of options. You must use one of the items in list.
[Private | Public]	Items between square braces and separated by a vertical bar are lists of optional choices. You may use one of the items in the list, or none of them.
...	An ellipsis mark indicates that the preceding item in the syntax example is repeated. Repeated items are usually optional and separated by commas.
; , . ( )	Other symbols must be typed as shown in the syntax example, with the exception of the underscore "_" character, which is used to show that a sample line has been split.

Suggested Reading - Scripter

For additional help in learning how to program or for more information about the Visual BASIC for Applications (VBA) language (which is nearly identical to the **Scripter** BASIC language) we recommend the following books:

Harris, Matthew (1997), *Teach Yourself Visual BASIC 5 for Applications in 21 Days*, Third Edition, SAMS Publishing, Indianapolis, IN, 1248 pp.

Lomax, Paul (1998), *VB and VBA in a Nutshell: The Languages*, O'Reilly & Associates, Inc., Sebastapol, CA, 633 pp.

Wang, Wallace (1998), *Visual BASIC 6.0 for Windows for Dummies*, IDG Books Worldwide, Foster City, CA, 477 pp.

Visual BASIC Compatibility

The **Scripter** BASIC programming language is compatible with the Visual BASIC for Applications language (VBA). Scripts that run in **Scripter** work in a VBA environment with few or no modifications. **Scripter** programs also work under Microsoft Visual BASIC. Unlike most Visual BASIC programs, however, **Scripter** programs are not event-driven. **Scripter** programs are procedural. They start with the first statement of the Main procedure, and end when the Main procedure ends.

VBA to Scripter

Some statements available in VBA are not supported in **Scripter** BASIC:

The VBA Collection object

The VBA Clipboard object

GoSub

On... GoSub

On...Goto

GoSub…Return

All Financial functions

Resume at current line

Erl

Option Compare

Conditional compilation

With Events

LinkExecute

LinkPoke

LinkRequest

LinkSend

Line numbers

LoadPicture

Multiple statements on one line (separated by ":")

Scripter to VBA

Conversely, some features of the **Scripter** BASIC language are not supported by VBA. Do not use the following features if you want to transfer your scripts from **Scripter** into VBA:

Clipboard function
CallersLine
User dialogs
PortInt
MacroRunThis
MacroDir
Wait instruction
MacroRun
DDEExecute
DDEPoke
DDERequest
DDEInitiate
DDETerminateAll
DDETerminate

Automation Model

The model chart shows you which objects provide access to other objects in the hierarchy. The *Application* object is at the top of the hierarchy and all objects are directly accessible from the *Application* object. To access many objects you must traverse from the Application object through one or more layers of sub-objects. People often refer to "drilling" or "boring" through the hierarchy to describe this traversal through several objects to obtain an object you want to use.

Application Object

The *Application* object represents the **Voxler** program. It is a single instance of **Voxler** and it is the root of all objects in **Voxler**. External programs typically create an instance of the *Application* object during initialization. In VB this is done using the *CreateObject* function as in:

```
'Create new Voxler application object
 Set VoxlerApp= CreateObject("Voxler.Application")
```

The *CreateObject* function activates a new instance of **Voxler**, and returns a reference to the *Application* object to the script. The *GetObject* function activates an existing instance of **Voxler** and returns the reference to the *Application* object to the script.

```
'Access existing Voxler object
  Set VoxlerApp = GetObject (,"Voxler.Application")
```

When **Voxler** is started by a script, its main window is initially hidden. To make the **Voxler** window visible, you must set the *Application* object's Visible property to *True*:

```
Set VoxlerApp = CreateObject("Voxler.Application")
VoxlerApp.Visible = True
```

The *Application* object provides access to the next level of objects in the hierarchy. Use the *CommandApi* property to obtain a reference to the **Voxler** commands.

Properties

Application
CommandApi
Name
FullName
Path
Version
Visible

Methods

Quit

Example

The following script demonstrates how the *Application* object is used.

```
Sub Main
 'Declares VoxlerApp as an object
  Dim VoxlerApp As Object

 'Creates an instance of the Voxler application object
 'and assigns it to the variable named "VoxlerApp"
  Set VoxlerApp = CreateObject("Voxler.Application")

 'Make Voxler visible
  VoxlerApp.Visible = True

End Sub
```

Application Property

The *Application* property returns an *Application* object. This is a read-only property.

Syntax

object.Application
object.Application = Application

Example

This example demonstrates how to return the application object.

```
'Return an application object
  Debug.Print VoxlerApp.Application.Name
```

Used by: Application object

CommandApi Object

The *CommandApi* object contains all of the properties of the various modules in the **Voxler** program. *CommandApi* refers to the accessing the commands from the *Application programming interface*. Using the *CommandApi* object requires accessing the property with the *Construct* method, specifying any settings with the *Option* method, and making the action with the *Do* or *DoOnce* method.

The *Do* method does the action and adds the action to the **Edit | Undo** list in the **Voxler** program. The *DoOnce* method does the action one time. The item is not added to the undo list.

All module and property additions, deletions, or changes are accessed through the *Construct* command in **Voxler** automation.

Methods

Construct

Do

DoOnce

Option

Redo

Reset

Undo

Example

The following script shows how the *CommandApi* object is accessed and demonstrates using the *CommandApi* object to open an existing file.

```
Sub Main
 'Declares VoxlerApp as an object
  Dim VoxlerApp As Object

 'Creates an instance of the Voxler application object
 'and assigns it to the variable named "VoxlerApp"
  Set VoxlerApp = CreateObject("Voxler.Application")

 'Make Voxler visible
  VoxlerApp.Visible = True

 'Access CommandApi
  Set CommandApi = VoxlerApp.CommandApi

 'Open an existing file
  CommandApi.Construct ("Open")
```

```
  CommandApi.Option ("Path", VoxlerApp.Path+"\Samples\Gold (ScatterPlot).voxb")
  CommandApi.DoOnce()

End Sub
```

Used by: Application object

CommandApi Property

The *CommandApi* property returns a *CommandApi* object. This allows access to all of the commands in **Voxler**.

Syntax

object.CommandApi

object.CommandApi = CommandApi

Example

This example accesses the command interface.

```
'Access the command interface
  Set CommandApi = VoxlerApp.CommandApi
```

Used by: Application object

Construct Method

The *Construct* method tells **Voxler** which command to create. The *Construct* method is accessed from the *CommandApi* object.

The *Construct* method must be used in combination with the *Option* method and the *Do* or *DoOnce* method to create an object.

Syntax

object.Construct("Construct_type")

Construct Types

CheckForUpdate

ClearHistory

ConnectModules

CreateModule

DeleteAllModules

DeleteModule

Export

Import

ModifyModule
MoveModule
New
NewNet
NewWks
Open
RenameModule
Save
ShowModule
ShowWindow
ViewDirection
ViewFitToWindow
ViewSize

Example

This example demonstrates how to open an existing file using the *Construct* method.

```
CommandApi.Construct ("Open")
CommandApi.Option ("Path", VoxlerApp.Path+"\Samples\Gold (ScatterPlot).voxb")
CommandApi.DoOnce()
```

Used by: CommandApi object

Import

The *Construct* method tells **Voxler** which command to create. The *Construct* method is accessed from the *CommandApi* object. The *Import* type allows the **Voxler** program to import an existing file into the **Network Manager**. This is similar to the **File | Import** command.

The *Construct* method must be used in combination with the *Option* method and the *Do* or *DoOnce* method to create an object.

Note that when importing point or well data (ACCDB, CSV, DAT, DBF, LAS, MDB, SLK, TXT, XLS, or XLSX), the columns are specified in the *DataSrc* options.

Syntax

object.Construct("Import")

Construct Type

Import

Parameter	Type	Description
AutoConnect	Boolean	If True, the new data is automatically connected. If False, the new data is not automatically connected.
ClearOptions	Boolean	If True, clear all filter options before adding a new option. If False, a new option will be concatenated to existing options.
ClearPath	Boolean	If True, clear current paths to import before adding a new path. If False, the new path will be concatenated to existing paths.
Filter	String	Filter id. Specify the file type, based on the filter ID.
GuiEnabled	Boolean	True enables dialogs in the Voxler window. False disables dialogs.
Option	String	One or more (key, value) pairs to be passed to the filter. Packed as "key=value", with multiple pairs comma or semicolon-separated.
Options	String	One or more (key, value) pairs to be passed to the filter. Packed as "key=value", with multiple pairs comma or semicolon-separated.
Path	String	The full path to a file to be imported. One or more paths can be specified by calling this method multiple times.
PersistOptions	Boolean	If True, persist filter options in the registry. If False, options are not saved.
ProgressEnabled	Boolean	True displays progress bars. False disables progress bars.
UndoRedoEnabled	Boolean	True allows undo and redo commands. False disallows undo and redo commands on the import.

Remarks

The *Import* type imports data files into the **Network Manager**. To open a data file in the worksheet, use the *Open* construct type.

Example

This example loads a VDAT file and attaches a volrender to it.

```
Sub Main

'Declares VoxlerApp as an object
Dim VoxlerApp As Object

'Creates an instance of the Voxler application object
'and assigns it to the variable named "VoxlerApp"
Set VoxlerApp = CreateObject("Voxler.Application")
'Set VoxlerApp = GetObject("Voxler.Application")

'Make Voxler visible
```

```
VoxlerApp.Visible = True

'Access CommandApi
Set CommandApi = VoxlerApp.CommandApi

'Load a VDAT file format
 CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "vdat")
  CommandApi.Option("Path", "c:\Program Files\Golden Software\Voxler
4\Samples\Jaw.vdat")
  CommandApi.Option("PersistOptions", "True")
 CommandApi.Do()

'Attach a volrender to the VDAT
 CommandApi.Construct ("CreateModule")
 CommandApi.Option ("Type", "VolRender")
 Commandapi.Option ("AutoConnect", "True")
 CommandApi.Option ("SourceModule", "Jaw.vdat")
 CommandApi.Do()

End Sub
```

Used by: CommandApi object

CreateModules

The *Construct* method tells **Voxler** which command to create. The *Construct* method is accessed from the *CommandApi object*. The *CreateModule* type allows the **Voxler** program to add a new module to the **Network Manager**. This is similar to the Network Menu commands.

The *Construct* method must be used in combination the *Option* method and the *Do* or *DoOnce* method to create an object.

Syntax

object.Construct("CreateModule")

Construct Type

CreateModule

Parameter	Type	Description
AutoConnect	Boolean	True connects the new module to the source module
DefaultPosition	Boolean	True places the module in the Network Manager at the default position. False allows you to specify the position with the XPosition and YPosition values.
SourceModule	string	Name of the source module
SourceModuleID	unsigned	Numeric ID of the source module
SourcePort	unsigned	The zero-based port index of the source module output port
Type	string	The type of module being created.
XPosition	integer	The numeric X location of the module in the Network Manager if DefaultPosition is set to False. In Pixels.
YPosition	integer	The numeric Y location of the module in the Network Manager if the DefaultPosition is set to False. In Pixels.

Remarks

XPosition and *YPosition* are in pixels. A position of 0, 0 places the module at the top left corner of the **Network Manager**.

Example

This example opens the Gold (ScatterPlot) example file. It adds a *Gridder module* at a specific Network Manager location. The *Gridder* module is automatically connected to the *Data module.*

```
Sub Main
 'Declares VoxlerApp as an object
  Dim VoxlerApp As Object

 'Creates an instance of the Voxler application object
 'and assigns it to the variable named "VoxlerApp"
  Set VoxlerApp = CreateObject("Voxler.Application")

 'Make Voxler visible
  VoxlerApp.Visible = True

 'Access CommandApi
  Set CommandApi = VoxlerApp.CommandApi

'Open an existing file
  CommandApi.Construct ("Open")
  CommandApi.Option ("Path", VoxlerApp.Path+"\Samples\Gold (ScatterPlot).voxb")
  CommandApi.DoOnce()

'Add a gridder module at a specific location. Automatically connect to the data.
```

```
  CommandApi.Construct ("CreateModule")
  CommandApi.Option("AutoConnect","True")
  CommandApi.Option("DefaultPosition", "False")
  CommandApi.Option("SourceModule", "GoldConcentration.dat")
  CommandApi.Option("Type","Gridder")
  CommandApi.Option("XPosition",40)
  CommandApi.Option("YPosition",100)
  CommandApi.Do()

End Sub
```

Used by: CommandApi object

ModifyModule

The *Construct* method tells **Voxler** which command to create. The *Construct* method is accessed from the *CommandApi object*. The *ModifyModule* type changes the properties of a module in the **Network Manager**.

The *Construct* method must be used in combination with the *Option* method and the *Do* or *DoOnce* method to create an object.

Syntax

object.Construct("ModifyModule")

Construct Type

ModifyModule

Parameter	Type	Description
Module	string	Name of the module to be moved
ModuleID	unsigned	Numeric ID of the module to be moved
Property	string	Name of the property to be modified.
PropertyID	unsigned	Numeric ID of the property to be modified.
PropertyValue	string	A target property value encoded as a string.

Remarks

The *Property*, *PropertyId*, and *PropertyValue* are dependent on the *Module* or *ModuleId* selected. See module pages for specific properties.

Example

This example opens the Gold (ScatterPlot) example file. It then changes the scatter plot color map to "Rainbow".

```
Sub Main
```

```
 'Declares VoxlerApp as an object
  Dim VoxlerApp As Object

 'Creates an instance of the Voxler application object
 'and assigns it to the variable named "VoxlerApp"
  Set VoxlerApp = CreateObject("Voxler.Application")

 'Make Voxler visible
  VoxlerApp.Visible = True

 'Access CommandApi
  Set CommandApi = VoxlerApp.CommandApi

'Open an existing file
  CommandApi.Construct ("Open")
  CommandApi.Option ("Path", VoxlerApp.Path+"\Samples\Gold (ScatterPlot).voxb")
  CommandApi.DoOnce()

 'Change color of scatter plot symbols
  CommandApi.Construct ("ModifyModule")
  CommandApi.Option ("Module", "ScatterPlot")
  CommandApi.Option ("ScatterPlotColormap","Rainbow")
  CommandApi.Do()

End Sub
```

Used by: CommandApi object

ConnectModules

The *Construct* method tells **Voxler** which command to create. The *Construct* method is accessed from the *CommandApi object*. The *ConnectModules* type allows the **Voxler** program to connect two modules in the **Network Manager**.

The *Construct* method must be used in combination with the *Option* method and the *Do* or *DoOnce* method to create an object.

Syntax

object.Construct("ConnectModules")

Construct Type

ConnectModules

Parameter	Type	Description
SourceModule	string	Name of input module

SourceModuleId	unsigned	Numeric ID of the input module
SourcePort	unsigned	The zero-based port index of the source module output port.
TargetModule	string	Name of output module
TargetModuleId	unsigned	Numeric ID of the output module
TargetPort	unsigned	The zero-based port index of the target module output port.

Remarks

SourceModule or SourceModuleID is required.

TargetModule or TargetModuleId is required.

Example

This example opens the *Gold (ScatterPlot)* example file. It adds a *Gridder module* and connects the *Gridder module* to the *Data module*.

```
Sub Main
 'Declares VoxlerApp as an object
  Dim VoxlerApp As Object

 'Creates an instance of the Voxler application object
 'and assigns it to the variable named "VoxlerApp"
  Set VoxlerApp = CreateObject("Voxler.Application")

 'Make Voxler visible
  VoxlerApp.Visible = True

 'Access CommandApi
  Set CommandApi = VoxlerApp.CommandApi

'Open an existing file
  CommandApi.Construct ("Open")
  CommandApi.Option ("Path", VoxlerApp.Path+"\Samples\Gold (ScatterPlot).voxb")
  CommandApi.DoOnce()

 'Add a gridder module using the CreateModule Construct command
  CommandApi.Construct ("CreateModule")
  CommandApi.Option("AutoConnect","False")
  CommandApi.Option("DefaultPosition", "True")
  CommandApi.Option("SourceModule", "GoldConcentration.dat")
  CommandApi.Option("Type","Gridder")
  CommandApi.Do()

 'Connect the gridder and data modules
```

```
  CommandApi.Construct ("ConnectModules")
  CommandApi.Option("SourceModule", "GoldConcentration.dat")
  CommandApi.Option("TargetModule", "Gridder")
  CommandApi.Do()

End Sub
```

Used by: CommandApi object

Export

The *Construct* method tells **Voxler** which command to create. The *Construct* method is accessed from the *CommandApi object*. The *Export* type allows the **Voxler** program to export an existing model from the Viewer window. This is similar to the **File | Export** command.

The *Construct* method must be used in combination with the *Option* method and the *Do* or *DoOnce* method to create an object.

There are two subtleties to exporting a module. First, exporting a file depends upon the correct module ID being selected. If you wish to export the full viewer window, the Module ID must always be 0. To export any other module, select the appropriate module ID. This idea is reiterated in the example scripts. Second, what file formats you can export to depends upon the file format of the imported file. Information about file formats is listed in **Voxler's** help.

Syntax

object.Construct("Export")

Construct Type

Export

Parameter	Type	Description
ClearOptions	Boolean	If True, clear all filter options before adding a new option. If False, a new option will be concatenated to existing options.
Filter	String	Optional filter id.
Module	string	Name of the module to be exported
ModuleID	unsigned	Numeric ID of the module to be exported
Option	String	One or more (key, value) pairs to be passed to the filter. Packed as "key=value", with multiple pairs semicolon-separated.
Options	String	One or more (key, value) pairs to be passed to the filter. Packed as "key=value", with multiple pairs semicolon-separated.
Path	String	The full path to the export file. Only one path can be specified.

PersistOptions	Boolean	If True, persist filter options in the registry. If False, do not save options.

Example

This example opens an existing VOXB file and exports it to a BMP file.

```
Sub Main

 'Declares VoxlerApp as an object
  Dim VoxlerApp As Object

 'Creates an instance of the Voxler application object
 'and assigns it to the variable named "VoxlerApp"
  Set VoxlerApp = CreateObject("Voxler.Application")

 'Make Voxler visible
  VoxlerApp.Visible = True

 'Access CommandApi
  Set CommandApi = VoxlerApp.CommandApi

 'Open an existing file
  CommandApi.Construct ("Open")
  CommandApi.Option ("Path", VoxlerApp.Path+"\Samples\Helens (ContourMap).voxb")
  CommandApi.DoOnce()

 'Export the viewer window to a bitmap
 'Change the export Path to one that exists on your computer
  CommandApi.Construct("Export")
   CommandApi.Option("GuiEnabled", "False")
   CommandApi.Option("ClearOptions", "False")
   CommandApi.Option("Filter", "bmp")
   CommandApi.Option("Options", "Height=418; Width=576")
   CommandApi.Option("Path", "C:/path/to/example/file.bmp")
   CommandApi.Option("PersistOptions", "True")
   CommandApi.Option("ModuleId", "0")
  CommandApi.Do()

End Sub
```

Used by: CommandApi object

Do Method

The *Do* method tells **Voxler** to do the action listed in the *Construct method*. The *Do* method is accessed from the *CommandApi object*. When the *Do* method is used, the command is added to the **Edit | Undo** list in **Voxler**.

The *Do* method must be used in combination with the *Construct method* and the *Option* method to create an object.

Syntax

object.**Do()**

Example

This example demonstrates how to edit a *Scatterplot* module's color map properties using the *Do* method. The *Scatterplot* color map change is added to the **Edit | Undo** list.

```
'Change color of scatter plot symbols
  CommandApi.Construct ("ModifyModule")
  CommandApi.Option ("Module", "ScatterPlot")
  CommandApi.Option ("ScatterPlotColormap","Terrain")
  CommandApi.Do()
```

Used by: CommandApi object

DoOnce Method

The *DoOnce* method tells **Voxler** to do the action listed in the *Construct* method. The *DoOnce* method is accessed from the *CommandApi object*. Use the *DoOnce* method when you do not want to be able to undo a command.

The *DoOnce* method must be used in combination with the *Construct method* and the *Option* method to create an object.

Syntax

object.DoOnce()

Example

This example demonstrates how to edit a *Scatterplot* module's color map properties using the *Do* method. The *Scatterplot* color map change is not added to the **Edit | Undo** list.

```
'Change color of scatter plot symbols
  CommandApi.Construct ("ModifyModule")
  CommandApi.Option ("Module", "ScatterPlot")
  CommandApi.Option ("ScatterPlotColormap","Terrain")
  CommandApi.DoOnce()
```

Used by: CommandApi object

Automation Examples

- Automation examples are located in several areas of the documentation.
- Each object, method, and property contains an example in the automation help. The methods and properties contain only a few lines for the example for those who only need small hints to use the method or property.
- Click on the *Used By* link for an example on how to tunnel through the hierarchy to the method or property. The *Used By* link takes you to the main object. You can copy the main object example plus the method or property example (in most cases) to create a full script.
- Some methods and properties contain a link showing a full example of the object. Full sample scripts are located in the *Automation Examples* help book.
- The SCRIPTS folder in the **Voxler** directory contains sample .BAS files to use in **Scripter**. By default, the SCRIPTS folder is located at: C:\Program Files\Golden Software\Voxler 4\Samples\Scripts.
- Sample scripts are available for download on the Golden Software home page.
- The online support forum is a location to search previous script discussion for examples. You may also post specific script questions on the forum.

Automation Example - HeightField

This is an example for HeightField.

```
'*************************************************************************************
*********
' HeightfieldModule.bas
' This script loads a test lattice and adds a heightfield module.
' It then changes all the properties of the heightfield.
'
'                                    -by SKP 4/2010
'
'*************************************************************************************
*********
Sub Main

 'Declares VoxlerApp as an object
  Dim VoxlerApp As Object

 'Creates an instance of the Voxler application object
 'and assigns it to the variable named "VoxlerApp"
  Set VoxlerApp = CreateObject("Voxler.Application")

 'Make Voxler visible
  VoxlerApp.Visible = True

 'Access CommandApi
  Set CommandApi = VoxlerApp.CommandApi

 'Create a new Voxler document
```

```
  CommandApi.Construct ("New")
  CommandApi.DoOnce()

 'Create the test lattice
  CommandApi.Construct("CreateModule")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option ("DefaultPosition", "True")
  CommandApi.Option ("Type", "TestLattice")
  CommandApi.Do()

 'Add a heightfield module
  CommandApi.Construct("CreateModule")
  CommandApi.Option ("AutoConnect", "True")
  CommandApi.Option ("SourceModule", "TestLattice")
  CommandApi.Option ("Type", "HeightField")
  CommandApi.Do()

 'Change the component being shown by the heightfield
 'Only available if more than one component exists in the lattice
  CommandApi.Construct("ModifyModule")
  CommandApi.Option("Module","HeightField")
  CommandApi.Option("HeightFieldComponent", "1")
  CommandApi.Do()

 'Change the height field drawing style
 'this is 0, 1, or 2 for shaded, lines, or points
  CommandApi.Option("HeightFieldDrawStyle", "1")
  CommandApi.Do()

 'Change the height field orientation
 'this is 0, 1, or 2 for XY Plane (Axial), XZ Plane (Coronal), or YZ Plane
(Sagittal)
 'this is only available when more than one grid node exists in each direction
  CommandApi.Option("HeightFieldOrientation", "1")
  CommandApi.Do()

 'Change the height field slice number
 'this is a number from 1 to the maximum grid row in the orientation direction
 'this is only available when more than one grid node exists in each direction
  CommandApi.Option("HeightFieldSlice", "18")
  CommandApi.Do()

 'Change the height field opacity
 'this is a number from 0 to 1
  CommandApi.Option("HeightFieldOpacity", "0.4")
  CommandApi.Do()
```

```
'Change the height field color map
 CommandApi.Option("HeightFieldColormap", "Rainbow")
 CommandApi.Do()

'Change the height field scale
 CommandApi.Option("HeightFieldScale", "0.6")
 CommandApi.Do()

'Display the height field legend
 CommandApi.Option ("HeightFieldLegendEnable", "True")
 CommandApi.Do()

'Change the legend orientation
'0 is for horizontal, 1 is for vertical
 CommandApi.Option ("HeightFieldLegendOrientation", "0")
 CommandApi.Do()

'Change the legend X position
'Ranges from 0-1
 CommandApi.Option ("HeightFieldLegendXPos", "0.3")
 CommandApi.Do()

'Change the legend Y position
'Ranges from 0-1
 CommandApi.Option ("HeightFieldLegendYPos", "0.9")
 CommandApi.Do()

'Change the legend width
'Ranges from 0-200
 CommandApi.Option ("HeightFieldLegendWidth", "20")
 CommandApi.Do()

'Change the legend length
'Ranges from 0-1024
 CommandApi.Option ("HeightFieldLegendLength", "400")
 CommandApi.Do()

'Change the legend title
 CommandApi.Option ("HeightFieldLegendTitle", "Legend Title")
 CommandApi.Do()

'Change the legend title font size
'Ranges from 4-72
 CommandApi.Option ("HeightFieldLegendTitleHeight", "20")
```

```
  CommandApi.Do()

 'Change the number of labels displayed in a legend
  CommandApi.Option ("HeightFieldLegendNumLabels", "3")
  CommandApi.Do()

 'Change the legend to use custom labels
  CommandApi.Option ("HeightFieldLegendUseCustomLabels", "True")
  CommandApi.Do()

 'Set the custom labels for the legend
  CommandApi.Option ("HeightFieldLegendCustomLabels", "1.7:low, 2.1:medium,
2.9:intermediate, 3.2:elevated, 3.7:high")
  CommandApi.Do()

 'Set the height for the legend labels
  CommandApi.Option ("HeightFieldLegendLabelHeight", "8")
  CommandApi.Do()

 'Set the label format type for the legend
  CommandApi.Option ("HeightFieldLabelFormatType", "0")
  CommandApi.Do()

 'Set the number of digits to display on the labels
  CommandApi.Option ("HeightFieldLabelFormatNumDigits", "2")
  CommandApi.Do()

 'Set the legend label prefix
  CommandApi.Option ("HeightFieldLabelFormatPrefix", "pre-")
  CommandApi.Do()

 'Set the legend label postfix
  CommandApi.Option ("HeightFieldLabelFormatPostfix", "-post")
  CommandApi.Do()

 'Set the legend font
  CommandApi.Option ("HeightFieldLegendFont", "Arial")
  CommandApi.Do()

 'Turn on or off antialiasing for the legend
  CommandApi.Option ("HeightFieldLegendAntialias", "True")
  CommandApi.Do()

 'Set the line and text color for the legend
  CommandApi.Option ("HeightFieldLegendFGColor", "Blue")
  CommandApi.Do()
```

```
 'Set the background color for the legend
  CommandApi.Option ("HeightFieldLegendBGColor", "10% Gray")
  CommandApi.Do()

 'Turn on or off the display of the legend background
  CommandApi.Option ("HeightFieldLegendShowBackground", "True")
  CommandApi.Do()

End Sub
```

Automation Example - ScatterPlot

This is an example for ScatterPlot.

```
'*************************************************************************************
*********
' ScatterPlotModule.bas
' This script loads a data file and adds a scatter plot to it.
' It then changes all the properties of the scatter plot.
'
'                                    -by SKP 4/2010
'
'*************************************************************************************
*********
Sub Main

 'Declares VoxlerApp as an object
  Dim VoxlerApp As Object

 'Creates an instance of the Voxler application object
 'and assigns it to the variable named "VoxlerApp"
  Set VoxlerApp = CreateObject("Voxler.Application")

 'Make Voxler visible
  VoxlerApp.Visible = True

 'Access CommandApi
  Set CommandApi = VoxlerApp.CommandApi

 'Create a new Voxler document
  CommandApi.Construct ("New")
  CommandApi.DoOnce()

 'Load the data file
  CommandApi.Construct("Import")
  CommandApi.Option("AutoConnect", "False")
```

```
  CommandApi.Option ("DefaultPosition", "True")
  CommandApi.Option ("Path", VoxlerApp.Path+"\Samples\GoldConcentration.dat")
  CommandApi.Option ("Options", "Defaults=1;EatWhitespace=1;_
      Delimiter=Space,tab,comma,semicolon;TextQualifier=doublequote,quote")
  CommandApi.Option ("Filter", "dat")
  CommandApi.Option ("GuiEnabled", "False")
  CommandApi.Do()

 'Add a scatter plot
  CommandApi.Construct("CreateModule")
  CommandApi.Option ("AutoConnect", "False")
  CommandApi.Option ("SourceModule", "GoldConcentration.dat ")
  CommandApi.Option ("Type", "ScatterPlot")
  CommandApi.Do()

 'Connect the isosurface and data modules
  CommandApi.Construct ("ConnectModules")
  CommandApi.Option ("SourceModule", "GoldConcentration.dat ")
  CommandApi.Option ("TargetModule", "ScatterPlot")
  CommandApi.Do()

 'This section changes the symbol
 'This is a number between 0 and 30.  0 is the top marker in the list.  30 is the
bottom marker in the list.
  CommandApi.Construct ("ModifyModule")
  CommandApi.Option ("Module", "ScatterPlot")
  CommandApi.Option ("ScatterPlotMarker", "17")
  CommandApi.Do()

 'This changes the symbol size for symbol marker numbers 0 to 29.
 'This is a number between 0 and 2. 0 is the smallest, 2 is the largest sized
symbol.
  CommandApi.Option ("ScatterPlotSizeMarker", "2")
  CommandApi.Do()

  Wait (4)

 'This changes the symbol to the last symbol
  CommandApi.Option ("ScatterPlotMarker", "30")
 CommandApi.Do()

 'This changes the symbol size for marker number 30
 'This is a number between 0 and 48.  The larger the number, the larger the symbol.
  CommandApi.Option ("ScatterPlotSizePoint", "3.7")
  CommandApi.Do()
```

```
'This changes the density of points to 50%
'This is a number between 0 and 7.
'0 = 100%, 1 = 50%, 2 = 33%, 3 = 25%, 4 = 20%, 5 = 10%, 6 = 5%, 7 = 1%
 CommandApi.Option ("ScatterPlotDensity", "3")
 CommandApi.Do()

'This shows lines connecting the scatter points
 CommandApi.Option ("ScatterPlotShowLines", "True")
 CommandApi.Do()

'This changes the line width of the scatter plot lines
 CommandApi.Option ("ScatterPlotLineWidth", "3.2")
 CommandApi.Do()

'This changes the color method to Fixed
 CommandApi.Option ("ScatterPlotColorMethod", "0")
 CommandApi.Do()

'This changes the color of the fixed scatter plot points
 CommandAPi.Option ("ScatterPlotColorFixed", "Magenta")
 CommandApi.Do()

 Wait (2)

'This changes the color method to By Data
 CommandApi.Option ("ScatterPlotColorMethod", "1")
 CommandApi.Do()

'This changes the color map being used to the "Rainbow" map
 CommandApi.Option ("ScatterPlotColormap", "Rainbow")
 CommandApi.Do()

'This changes the color component, when the data set has more than one component
 CommandApi.Option ("ScatterPlotColorComponent", "2")
 CommandApi.Do()

'Display the ScatterPlot legend
 CommandApi.Option ("ScatterPlotLegendEnable", "True")
 CommandApi.Do()

'Change the legend orientation
'0 is for horizontal, 1 is for vertical
 CommandApi.Option ("ScatterPlotLegendOrientation", "0")
 CommandApi.Do()
```

```
 'Change the legend X position
 'Ranges from 0-1
  CommandApi.Option ("ScatterPlotLegendXPos", "0.3")
  CommandApi.Do()

 'Change the legend Y position
 'Ranges from 0-1
  CommandApi.Option ("ScatterPlotLegendYPos", "0.9")
  CommandApi.Do()

 'Change the legend width
 'Ranges from 0-200
  CommandApi.Option ("ScatterPlotLegendWidth", "20")
  CommandApi.Do()

 'Change the legend length
 'Ranges from 0-1024
  CommandApi.Option ("ScatterPlotLegendLength", "400")
  CommandApi.Do()

 'Change the legend title
  CommandApi.Option ("ScatterPlotLegendTitle", "Legend Title")
  CommandApi.Do()

 'Change the legend title font size
 'Ranges from 4-72
  CommandApi.Option ("ScatterPlotLegendTitleHeight", "20")
  CommandApi.Do()

 'Change the number of labels displayed in a legend
  CommandApi.Option ("ScatterPlotLegendNumLabels", "3")
  CommandApi.Do()

 'Change the legend to use custom labels
  CommandApi.Option ("ScatterPlotLegendUseCustomLabels", "True")
  CommandApi.Do()

 'Set the custom labels for the legend
  CommandApi.Option ("ScatterPlotLegendCustomLabels", _
      "1.7:low, 2.1:medium, 2.9:intermediate, 3.2:elevated, 3.7:high")
  CommandApi.Do()

 'Set the height for the legend labels
  CommandApi.Option ("ScatterPlotLegendLabelHeight", "8")
  CommandApi.Do()
```

```
 'Set the label format type for the legend
  CommandApi.Option ("ScatterPlotLabelFormatType", "0")
  CommandApi.Do()

 'Set the number of digits to display on the labels
  CommandApi.Option ("ScatterPlotLabelFormatNumDigits", "2")
  CommandApi.Do()

 'Set the legend label prefix
  CommandApi.Option ("ScatterPlotLabelFormatPrefix", "pre-")
  CommandApi.Do()

 'Set the legend label postfix
  CommandApi.Option ("ScatterPlotLabelFormatPostfix", "-post")
  CommandApi.Do()

 'Set the legend font
  CommandApi.Option ("ScatterPlotLegendFont", "Arial")
  CommandApi.Do()

 'Turn on or off antialiasing for the legend
  CommandApi.Option ("ScatterPlotLegendAntialias", "True")
  CommandApi.Do()

 'Set the line and text color for the legend
  CommandApi.Option ("ScatterPlotLegendFGColor", "Blue")
  CommandApi.Do()

 'Set the background color for the legend
  CommandApi.Option ("ScatterPlotLegendBGColor", "10% Gray")
  CommandApi.Do()

 'Turn on or off the display of the legend background
  CommandApi.Option ("ScatterPlotLegendShowBackground", "True")
  CommandApi.Do()

End Sub
```

Automation Example - VolRender

This is an example for VolRender.

```
'*************************************************************************************
*********
' VolRenderModule.bas
' This script loads a test lattice and adds a VolRender module to it.
```

```
' It then changes all the properties of the VolRender module.
'
'                                    -by SKP 4/2010
'
'*******************************************************************************
*********
Sub Main

 'Declares VoxlerApp as an object
  Dim VoxlerApp As Object

 'Creates an instance of the Voxler application object
 'and assigns it to the variable named "VoxlerApp"
  Set VoxlerApp = CreateObject("Voxler.Application")

 'Make Voxler visible
  VoxlerApp.Visible = True

 'Access CommandApi
  Set CommandApi = VoxlerApp.CommandApi

 'Create a new Voxler document
  CommandApi.Construct ("New")
  CommandApi.DoOnce()

 'Load the test lattice
  CommandApi.Construct("CreateModule")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option ("DefaultPosition", "True")
  CommandApi.Option ("Type", "TestLattice")
  CommandApi.Do()

 'Add a VolRender module
  CommandApi.Construct("CreateModule")
  CommandApi.Option ("AutoConnect", "True")
  CommandApi.Option ("SourceModule", "TestLattice")
  CommandApi.Option ("Type", "VolRender")
  CommandApi.Do()

 'Change the volrender component
  CommandApi.Construct("ModifyModule")
  CommandApi.Option ("Module", "VolRender")
  CommandApi.Option ("VolRenderComponent", "2")
  CommandApi.Do()

 'Change the volrender drawing style
```

```
'Value is 0 or 1 for 2D Textures or 3D Textures
 CommandApi.Option ("VolRenderDrawStyle", "1")
 CommandApi.Do()

'Change the volrender color map
 CommandApi.Option ("VolRenderColormap", "Rainbow")
 CommandApi.Do()

'Change the number of slices
'Option only available if VolRenderDrawStyle is set to 1
 CommandApi.Option ("VolRenderNumSlices","200")
 CommandApi.Do()

'Change the opacity of the volrender
'Varies between 0 and 1. 0 is fully transparent, 1 is fully opaque
 CommandApi.Option ("VolRenderOpacity", "0.7")
 CommandApi.Do()

'Change the method used to combine voxels along the viewing rays
'Value is 0, 1, or 2 for Maximum Intensity, Sum Intensity, or Alpha Blending
 CommandApi.Option ("VolRenderComposition","2")
 CommandApi.Do()

'Change the interpolation method
'Value is 0 or 1, for Nearest Neighbor or Trilinear
 CommandApi.Option ("VolRenderInterpolation", "1")
 CommandApi.Do()

'Turn off paletted textures for volrender
 CommandApi.Option ("VolRenderUsePalette", "False")
 CommandApi.Do()

'Display the VolRender legend
 CommandApi.Option ("VolRenderLegendEnable", "True")
 CommandApi.Do()

'Change the legend orientation
'0 is for horizontal, 1 is for vertical
 CommandApi.Option ("VolRenderLegendOrientation", "0")
 CommandApi.Do()

'Change the legend X position
'Ranges from 0-1
 CommandApi.Option ("VolRenderLegendXPos", "0.3")
 CommandApi.Do()
```

```
 'Change the legend Y position
 'Ranges from 0-1
  CommandApi.Option ("VolRenderLegendYPos", "0.9")
  CommandApi.Do()

 'Change the legend width
 'Ranges from 0-200
  CommandApi.Option ("VolRenderLegendWidth", "20")
  CommandApi.Do()

 'Change the legend length
 'Ranges from 0-1024
  CommandApi.Option ("VolRenderLegendLength", "400")
  CommandApi.Do()

 'Change the legend title
  CommandApi.Option ("VolRenderLegendTitle", "Legend Title")
  CommandApi.Do()

 'Change the legend title font size
 'Ranges from 4-72
  CommandApi.Option ("VolRenderLegendTitleHeight", "20")
  CommandApi.Do()

 'Change the number of labels displayed in a legend
  CommandApi.Option ("VolRenderLegendNumLabels", "3")
  CommandApi.Do()

 'Change the legend to use custom labels
  CommandApi.Option ("VolRenderLegendUseCustomLabels", "True")
  CommandApi.Do()

 'Set the custom labels for the legend
  CommandApi.Option ("VolRenderLegendCustomLabels", _
      "0.07:low, 0.21:medium, 0.45:intermediate, 0.72:elevated, 0.97:high")
  CommandApi.Do()

 'Set the height for the legend labels
  CommandApi.Option ("VolRenderLegendLabelHeight", "8")
  CommandApi.Do()

 'Set the label format type for the legend
  CommandApi.Option ("VolRenderLabelFormatType", "0")
  CommandApi.Do()
```

```
 'Set the number of digits to display on the labels
  CommandApi.Option ("VolRenderLabelFormatNumDigits", "2")
  CommandApi.Do()

 'Set the legend label prefix
  CommandApi.Option ("VolRenderLabelFormatPrefix", "pre-")
  CommandApi.Do()

 'Set the legend label postfix
  CommandApi.Option ("VolRenderLabelFormatPostfix", "-post")
  CommandApi.Do()

 'Set the legend font
  CommandApi.Option ("VolRenderLegendFont", "Arial")
  CommandApi.Do()

 'Turn on or off antialiasing for the legend
  CommandApi.Option ("VolRenderLegendAntialias", "True")
  CommandApi.Do()

 'Set the line and text color for the legend
  CommandApi.Option ("VolRenderLegendFGColor", "Blue")
  CommandApi.Do()

 'Set the background color for the legend
  CommandApi.Option ("VolRenderLegendBGColor", "10% Gray")
  CommandApi.Do()

 'Turn on or off the display of the legend background
  CommandApi.Option ("VolRenderLegendShowBackground", "True")
  CommandApi.Do()

End Sub
```

Automation Example - WellData and WellRender

This is an example for WellData and WellRender.

```
'*************************************************************************
*********
' WellDataModule.bas
'
' This imports data into a WellData module and modifies its properties.  The
' wells are loaded from three sheets from an Excel workbook to demonstrate
' how complex well data can be loaded from multiple sources and appended to
```

```
' an existing well node.
'*********************************************************************************
*********

' If Voxler is running, retrieve the current instance of Voxler.  Otherwise,
' create a new instance of Voxler.
Function VoxlerApplication()
   On Error Resume Next
   Set VoxlerApplication = GetObject(, "Voxler.Application")
   If Err.Number <> 0 Then
      Set VoxlerApplication = CreateObject("Voxler.Application")
   End If
   On Error GoTo 0
End Function

Sub Main

 'Retrieve running instance of Voxler or create a new instance of Voxler
  Dim VoxlerApp As Object
  Set VoxlerApp = VoxlerApplication()

 'Make Voxler visible
  VoxlerApp.Visible = True

 'Access CommandApi
  Set CommandApi = VoxlerApp.CommandApi

 'Create a new Voxler document
  CommandApi.Construct ("New")
  CommandApi.DoOnce()

   'Load the well data from three sheets.  The first sheet contains the collars, the
  'second sheet contains the trajectories, and the third sheet contains example
data.
  'Import the Collars table and set the columns
  'Import the Collars table
  CommandApi.Construct ("Import")
  CommandApi.Option ("Path", VoxlerApp.Path+"Samples\SampleWellData.xlsx")
  CommandApi.Option ("Options", "Defaults=1;Sheet=Collars")
  CommandApi.Option ("GuiEnabled", "False")
  CommandApi.Do()

  'Set OutputType for the Collars table to Wells (0 = Points, 1= Wells)
  CommandApi.Construct("ModifyModule")
  CommandApi.Option("Module","SampleWellData.xlsx - Collars")
  CommandApi.Option ("GuiEnabled", "True")
```

```
  CommandApi.Option ("ProgressEnabled", "True")
  CommandApi.Option ("UndoRedoEnabled", "True")
  CommandApi.Option ("OutputType", "1")
  CommandApi.Do()

  'Set WellSheetType for the Collars table to Collars (0 = All, 1 = Collars, 2 = Dir
Survey, 3 = From/To, 4 = Logs/Curves, 5 = XYZ path)
  CommandApi.Construct("ModifyModule")
  CommandApi.Option("Module","SampleWellData.xlsx - Collars")
  CommandApi.Option ("WellSheetType", "1")
  CommandApi.Do()

  'Set Well ID (WellColID) For Collars table
  CommandApi.Construct("ModifyModule")
  CommandApi.Option("Module","SampleWellData.xlsx - Collars")
  CommandApi.Option ("WellColID", "1")
  CommandApi.Do()

  'Set Top X column (WellColTopX) For Collars table
  CommandApi.Construct("ModifyModule")
  CommandApi.Option("Module","SampleWellData.xlsx - Collars")
  CommandApi.Option ("WellColTopX", "1")
  CommandApi.Do()

  'Set Top Y column (WellColTopY) For Collars table
  CommandApi.Construct("ModifyModule")
  CommandApi.Option("Module","SampleWellData.xlsx - Collars")
  CommandApi.Option ("WellColTopY", "2")
  CommandApi.Do()

  'Set Top Z column (WellColTopZ) For Collars table
  CommandApi.Construct("ModifyModule")
  CommandApi.Option("Module","SampleWellData.xlsx - Collars")
  CommandApi.Option ("WellColTopZ", "3")
  CommandApi.Do()

  'Set Azimuth column (WellColAz) For Collars table
  CommandApi.Construct("ModifyModule")
  CommandApi.Option("Module","SampleWellData.xlsx - Collars")
  CommandApi.Option ("WellColAz", "4")
  CommandApi.Do()

  'Set Vertical Direction to Dip (WellColVertType) For Collars table (0 = Dip, 1 =
Inclination)
  CommandApi.Construct("ModifyModule")
  CommandApi.Option("Module","SampleWellData.xlsx - Collars")
```

```
  CommandApi.Option ("WellColVertType", "0")
  CommandApi.Do()

  'Set Dip column (WellColVertDip) For Collars table
  CommandApi.Construct("ModifyModule")
  CommandApi.Option("Module","SampleWellData.xlsx - Collars")
  CommandApi.Option ("WellColVertDip", "5")
  CommandApi.Do()

  'Set Azimuth column (WellColTotalDepth) For Collars table
  CommandApi.Construct("ModifyModule")
  CommandApi.Option("Module","SampleWellData.xlsx - Collars")
  CommandApi.Option ("WellColTotalDepth", "6")
  CommandApi.Do()

  'Import the Trajectories table and set the columns
  'Import the Trajectories table
  CommandApi.Construct ("Import")
  CommandApi.Option ("Path", VoxlerApp.Path+"Samples\SampleWellData.xlsx")
  CommandApi.Option ("Options", "Defaults=1;Sheet=Trajectories")
  CommandApi.Option ("GuiEnabled", "False")
  CommandApi.Do()

  'Set OutputType for the Trajectories table to Wells (0 = Points, 1= Wells)
  CommandApi.Construct("ModifyModule")
  CommandApi.Option("Module","SampleWellData.xlsx - Trajectories")
  CommandApi.Option ("GuiEnabled", "True")
  CommandApi.Option ("ProgressEnabled", "True")
  CommandApi.Option ("UndoRedoEnabled", "True")
  CommandApi.Option ("OutputType", "1")
  CommandApi.Do()

  'Set WellSheetType for the Trajectories table to Dir Survey (0 = All, 1 = Collars,
2 = Dir Survey, 3 = From/To, 4 = Logs/Curves, 5 = XYZ path)
  CommandApi.Construct("ModifyModule")
  CommandApi.Option("Module","SampleWellData.xlsx - Trajectories")
  CommandApi.Option ("WellSheetType", "2")
  CommandApi.Do()

  'Set Well ID (WellColID) For Trajectories table
  CommandApi.Construct("ModifyModule")
  CommandApi.Option("Module","SampleWellData.xlsx - Trajectories")
  CommandApi.Option ("WellColID", "1")
  CommandApi.Do()

  'Set Azimuth column (WellColAz) For Trajectories table
```

```
  CommandApi.Construct("ModifyModule")
  CommandApi.Option("Module","SampleWellData.xlsx - Trajectories")
  CommandApi.Option ("WellColAz", "2")
  CommandApi.Do()

  'Set Vertical Direction to Inclination (WellColVertType) For Trajectories table (0
= Dip, 1 = Inclination)
  CommandApi.Construct("ModifyModule")
  CommandApi.Option("Module","SampleWellData.xlsx - Trajectories")
  CommandApi.Option ("WellColVertType", "1")
  CommandApi.Do()

  'Set Dip column (WellColVertInc) For Trajectories table
  CommandApi.Construct("ModifyModule")
  CommandApi.Option("Module","SampleWellData.xlsx - Trajectories")
  CommandApi.Option ("WellColVertInc", "2")
  CommandApi.Do()

  'Set Measured Depth column (WellColVertDip) For Trajectories table
  CommandApi.Construct("ModifyModule")
  CommandApi.Option("Module","SampleWellData.xlsx - Trajectories")
  CommandApi.Option ("WellColMeasDepth", "1")
  CommandApi.Do()

  'Import the Samples table and set the columns
  'Import the Samples table
  CommandApi.Construct ("Import")
  CommandApi.Option ("Path", VoxlerApp.Path+"Samples\SampleWellData.xlsx")
  CommandApi.Option ("Options", "Defaults=1;Sheet=Samples")
  CommandApi.Option ("GuiEnabled", "False")
  CommandApi.Do()

  'Set OutputType for the Samples table to Wells (0 = Points, 1= Wells)
  CommandApi.Construct("ModifyModule")
  CommandApi.Option("Module","SampleWellData.xlsx - Samples")
  CommandApi.Option ("GuiEnabled", "True")
  CommandApi.Option ("ProgressEnabled", "True")
  CommandApi.Option ("UndoRedoEnabled", "True")
  CommandApi.Option ("OutputType", "1")
  CommandApi.Do()

  'Set WellSheetType for the Samples table to From/To Logs (0 = All, 1 = Collars, 2
= Dir Survey, 3 = From/To, 4 = Logs/Curves, 5 = XYZ path)
  CommandApi.Construct("ModifyModule")
  CommandApi.Option("Module","SampleWellData.xlsx - Samples")
  CommandApi.Option ("WellSheetType", "3")
```

```
CommandApi.Do()

'Set From column (WellColFrom) For Samples table
CommandApi.Construct("ModifyModule")
CommandApi.Option("Module","SampleWellData.xlsx - Samples")
CommandApi.Option ("WellColFrom", "1")
CommandApi.Do()

'Set To column (WellColTo) For Samples table
CommandApi.Construct("ModifyModule")
CommandApi.Option("Module","SampleWellData.xlsx - Samples")
CommandApi.Option ("WellColTo", "2")
CommandApi.Do()

'Set number of logs (WellColLogCount) For Samples table
CommandApi.Construct("ModifyModule")
CommandApi.Option("Module","SampleWellData.xlsx - Samples")
CommandApi.Option ("WellColLogCount", "2")
CommandApi.Do()

'Set the first log column (ColLog-1) For Samples table
CommandApi.Construct("ModifyModule")
CommandApi.Option("Module","SampleWellData.xlsx - Samples")
CommandApi.Option ("ColLog-1", "2")
CommandApi.Option ("ColLog-2", "4")
CommandApi.Do()

'Create WellData and WellRender modules and connect them to the data
'Add a WellData module using the CreateModule Construct command
CommandApi.Construct ("CreateModule")
CommandApi.Option("AutoConnect","False")
CommandApi.Option("DefaultPosition", "True")
CommandApi.Option("SourceModule", "SampleWellData.xlsx - Collars")
CommandApi.Option("Type","WellData")
CommandApi.Do()

'Connect the Weldata and data modules
CommandApi.Construct ("ConnectModules")
CommandApi.Option("SourceModule", "SampleWellData.xlsx - Collars")
CommandApi.Option("TargetModule", "WellData")
CommandApi.Do()
CommandApi.Construct ("ConnectModules")
CommandApi.Option("SourceModule", "SampleWellData.xlsx - Trajectories")
CommandApi.Option("TargetModule", "WellData")
CommandApi.Option("TargetPort", "2")
```

```
 CommandApi.Do()
 CommandApi.Construct ("ConnectModules")
 CommandApi.Option("SourceModule", "SampleWellData.xlsx - Samples")
 CommandApi.Option("TargetModule", "WellData")
 CommandApi.Option("TargetPort", "3")
 CommandApi.Do()

 'Add a WellRender module using the CreateModule Construct command
 CommandApi.Construct ("CreateModule")
 CommandApi.Option("AutoConnect","True")
 CommandApi.Option("DefaultPosition", "True")
 CommandApi.Option("SourceModule", "WellData")
 CommandApi.Option("Type","WellRender")
 CommandApi.Do()

 'Add a bounding box for reference
 CommandApi.Construct ("CreateModule")
 CommandApi.Option ("Type", "BoundingBox")
 CommandApi.Option ("AutoConnect", "True")
 CommandApi.Option ("SourceModule", "WellData")
 CommandApi.Do()

 'Display the data using a default WellRender node
 CommandApi.Option ("Type", "WellRender")
 CommandApi.Do()
 CommandApi.Construct ("ModifyModule")
 CommandApi.Option ("Module", "WellRender")
 CommandApi.Option ("WellRenderShowLabels", "True")
 CommandApi.Do()

'Label the wells by their name and modify some label properties
 CommandApi.Construct ("ModifyModule")
 CommandApi.Option ("Module", "WellRender")
 CommandApi.Option ("WellRenderShowLabels", "True")
 CommandApi.Do()
 CommandApi.Option ("WellRenderXOffset", "20")
 CommandApi.Do()
 CommandApi.Option ("WellRenderYOffset", "30")
 CommandApi.Do()
 CommandApi.Option ("WellRenderShowLeaderLines", "True")
 CommandApi.Do()

 'Modify path properties
 CommandApi.Option ("WellRenderPathColor", "Yellow")
 CommandApi.Do()
```

```
 CommandApi.Option ("WellRenderPathSize", "3")
 CommandApi.Do()
 Wait(5)

 'Show interval data.  Logs for interval color, interval size, path color, path
size,
 'etc., are retrieved by their name.
 CommandApi.Option ("WellRenderIntDataLog", "MnO")
 CommandApi.Do()

 'Set the color method and log, use 0 for fixed color or 1 for color by log value
 CommandApi.Option ("WellRenderIntDataColorMethod", "1")
 CommandApi.Do()
 CommandApi.Option ("WellRenderIntDataColorLog", "MnO")
 CommandApi.Do()

 'Change the color map
 CommandApi.Option ("WellRenderIntDataColormap", "Rainbow")
 CommandApi.Do()

 'Set the size method and log, use 0 for fixed size or 1 for size by log value
 CommandApi.Option ("WellRenderIntDataSizeMethod", "1")
 CommandApi.Do()
 CommandApi.Option ("WellRenderIntDataSizeLog", "TiO2")
 CommandApi.Do()

 'Turn the interval display on
 CommandApi.Option ("WellRenderShowIntData", "True")
 CommandApi.Do()
 Wait(5)

 'Turn two wells off.  Note the use of the optional parameter on CommandApi.Option
 'to identify the well.
 CommandApi.Option ("WellRenderShowWell", "False", "Well=MW-5")
 CommandApi.Do()
 CommandApi.Option ("WellRenderShowWell", "False", "Well=MW-2")
 CommandApi.Do()

End Sub
```

Appendix A - File Formats

The following file formats are supported for import and export from **Voxler**.

Abbreviations are listed below the chart. Click the green links to display the column header and abbreviation information on this page. Click the blue links to navigate to a new page with detailed information on the selected file format.

File Extension	File Format	Import	Import Data Type	Export Method	Export Data Type
?10g	GLOBE DEM	2	U	-	-
ACCDB	Access Database	2,6	P	-	-
ADF	Esri Arc/Info Binary Grid	2	U	3b 3d	U
AM, COL	Amira Mesh	2	U3, R3, C3, P	3a 3b 3d	U3, R3, C3
AN1, AN2	ACR-NEMA Medical Image	2	U2, U3, B, I	-	-
ASC AIG, AGR, GRD	Esri ArcInfo ASCII Grid	2	U	3b 3d	U
ASC, DAT, GRD, CPS, CPS3	CPS-3 Grid	2	U	3b 3d	U
ASC, DAT, GRD, XYZ, ZMAP, ZYC, ZYCOR	Z-MAP Plus Grid	2	U	3b 3d	U
ASI	Amira Stacked Image	2	U3, R3	-	-
BIL, BIP, BSQ	Banded Lattice	2	U	3b 3d	U

BLN	Golden Software Blanking	2, 6	V2, V3 S	7	P
BMP	Windows Bitmap	2	B, I	3d 4	B, I
BNA	Atlas Boundary	2, 6	P	3a, 3b, 3d, 7	P
CSV	Data	2, 6	P	3a, 7	P
DAT	Golden Software Data	2, 6	P	3a, 3b, 7	P
DAT	XYZ Grid	-	-	3d	U
DBF	DBF Database	2, 6	P	-	-
DDF, TAR, TAR.GZ, ZIP, TGZ	SDTS Raster (DEM)	2	U2	-	-
DDF, TAR, TAR.GZ, ZIP, TGZ	SDTS Vector (TVP)	2	V2, S	-	-
DEM	USGS Digital Elevation Model	2	U2	3b 3d	N
DIC, DCM, AN?	DICOM3 Medical Image	2	U2, U3, B, I	-	-
DLG, LGO, LGS	USGS Digital Line Graph	2	V2, S	-	-
DOS, DAT	USGS ETOPO5	2	U	-	-
DT0, DT1, DT2, DT?	DTED	2	U2	-	-
DXF	AutoCAD DXF Drawing	2	V3, S	3c, 3d	V3, S
E00	Esri ArcInfo Export	2	V2, S	-	-
E00	Esri Grid File	2	U2	-	-
ECW	ER Mapper	2	B, I	-	-

EPS	Encapsulated Postscript	-	-	3d 4	B, I
ERS	ER Mapper Grid	2	U	3b 3d	U
FLD	AVS Field	2	U3, R3, C3	3a 3b 3d	U3, R3, C3
FLT	Esri Float Grid	2	U	3b 3d	U
GIF	GIF Image	2	B, I	3d 4	B, I
GRB, GRIB	GRIB Global Weather Data Grid	2	U2	-	-
GRD	Surfer Grid	2	U2	3b 3d	N
GRD, GGF	Geosoft Binary Grid	2	U	3b 3d	U
GSB	Golden Software Boundary	2	V2, S	-	-
GSI	Golden Software Interchange	2	V2, S	-	-
GXF	Grid eXchange	2	U	3b 3d	U
HDF	Hierarchical Data Format	2	U3, R3, C3	3a 3b 3d	U3, R3, C3
HDR	GTOPO-30	2	U2	-	-
HDR (data set), IMG (image)	Analyze 7.5 Medical Image	2	U3	3b 3d	U3
HGT, ZIP	NASA SRTM Grid Data	2	U2	-	-
IMG	ERDAS Imagine Image	2	U2, I	-	-
INFO	Leica Confocal Raw Slices	2	U3	-	-
IV	Inventor, SGI Open Inventor	2	G	3c 4	G

JPG, JPEG	JPEG Compressed Bitmap	2	B, I	3d 4	B, I
JP2, J2K, JPC, JPT, JPEG2000, J2000	JPEG-2000 File Interchange	2	B, I	3d 4	B, I
KML, KMZ	Google Earth Keyhole Markup	2	V2, P	-	-
LAS	LAS LiDAR Binary File	2, 6	P	-	-
LAS	LAS Well Data	2	P	-	-
LAT	Iris Explorer	2	U3, R3, C3	3a 3b 3d	U3, R3, C3
MDB	Microsoft Access	2, 6	P	-	-
MIF	MapInfo Interchange	2	V2, S	-	-
NC	NetCDF Lattice	2	U	3b 3d	U
P3D, XYZ	PLOT-3D	2	C3, P	3a 3b 3d	C3
PCX	ZSoft Paintbrush Bitmap	2	B, I	3d 4	B, I
PDF	PDF (Vector)	2	G, I	-	-
PDF	PDF (Raster)	2	G, I	3d 4	G, I
PLT	Golden Software PlotCall	2	V2	-	-
PLY	Stanford Polygon	2	V3, G	-	-
PNG	Portable Network Graphic	2	B, I	3d 4	B

PNM, PPM, PGM, PBM	Portable Bitmap	2	B, I	3d 4	B, I
RAS, SUN	Sun Raster	2	B, I	3d 4	B, I
RAW, BIN	Raw Binary Lattice	2	U3	3b 3d	U3
RGB, RGBA, BW	Silicon Graphics (SGI) RGB Image	2	B, I	3d 4	B, I
RST, IMG	Idrisi Raster	2	U	-	-
SEG, SP1	SEG-SP1 Data File Format	2, 6	P	-	-
SGY, SEGY	SEG-Y Seismic Data	2	U2, V2	-	-
SHP, TAR.GZ, TAR, ZIP, TGZ	Esri Shapefile	2	V3, S	-	-
SID	LizardTech MrSID Image	2	B, I	-	-
SLK	Sylk Spreadsheet	2, 6	P	3a, 7	P
STK	Metamorph STK	2	U2 U3	-	-
TAR	SDTS Raster (DEM)	2	U2	-	-
TAR	SDTS Vector (TVP)	2	V2, S	-	-
TAR.GZ	SDTS Raster (DEM)	2	U2	-	-
TAR.GZ	SDTS Vector (TVP)	2	V2, S	-	-
TGA	TrueVision Targa	2	B, I	3d 4	B, I
TGZ	SDTS Raster (DEM)	2	U2	-	-
TGZ	SDTS Vector (TVP)	2	V2, S	-	-

TIF, TIFF	Tagged Image File Format	2	U2, U3, B, I	3d 4	B, I
TXT	Text Data	2, 6	P	3a, 7	P
VDAT	Voxler Data	2	U2, R2, C2, U3, R3, C3, P	3a 3b 3d	U2, R2, C2, U3, R3, C3, P
VOXB	Voxler Network File	6	-	5	-
VTK	Visualization Toolkit	2	U3, R3, C3, I	3a 3b 3d	U3, R3, C3, I
X, XIMG	AVS X-Image	2	B, I	3d 4	B, I
XLS	Excel Spreadsheet	2, 6	P	3a, 7	P
XLSM	Excel 2007 Macro-Enabled Spreadsheet	2, 6	P	-	-
XLSX	Excel 2007 Spreadsheet	2, 6	P	3a, 7	P
ZIP	SDTS Raster (DEM)	2	U2	-	-
ZIP	SDTS Vector (TVP)	2	V2, S	-	-
File Extension	File Format	Import Method	Formats Supported for Import[1]	Export Method	Formats Supported for Export[1]

1 The table uses the following format abbreviations:

U = uniform lattice

U2 = 2D uniform lattice

U3 = 3D uniform lattice

R = rectilinear lattice

R2 = 2D rectilinear lattice

R3 = 3D rectilinear lattice

C = curvilinear lattice

C2 = 2D curvilinear lattice

C3 = 3D curvilinear lattice

N = Lattice has one component per node and only one slice

I = images (planar uniform lattice, 1 to 4 color components)

V2 = 2D Vector Graphics

V3 = 3D Vector Graphics

P = point data

G = geometry or scene data (iv)

W = well data or log data

- = not supported

2 Choose the **File | Import** command.

3a Select a data source module and choose the **File | Save Data** command.

3b Select a lattice source module (i.e. *Gridder*) and choose the **File | Save Data** command

3c Select an IV data module and choose the **File | Save Data** command

3d Select an image module and choose the **File | Save Data** command

4 Choose the **File | Export** command.

5 Choose the **File | Save** command.

6 Choose the **File | Open** command.

7 Choose the **File | Save As** command in the worksheet.

File Descriptions

GLOBE DEM Data .?10g File Description

The GLOBE DEM data .?10g files can be imported. The GLOBE DEM data ?10s files are not supported.

File Description

GLOBE DEM stands for *Global Land One-Kilometer Base Elevation (GLOBE) digital elevation model (DEM)*. The files are 30-arc-second (1-km) gridded, quality-controlled global digital elevation models.

GLOBE DEM is a global data set covering 180 degrees West to 180 degrees East longitude and 90 degrees North to 90 degrees South latitude. The horizontal grid spacing is 30 arc-seconds (0.008333... degrees) in latitude and longitude, resulting in dimensions of 21,600 rows and 43,200 columns. At the Equator, a degree of latitude is about 111 kilometers. GLOBE has 120 values per degree, giving GLOBE slightly better than 1 km gridding at the Equator, and progressively finer longitudinally toward the Poles. The horizontal coordinate system is seconds of latitude and longitude referenced to World Geodetic System 84 (WGS84). The vertical units represent elevation in meters above Mean Sea Level. The elevation values range from -407 to 8,752 meters on land. In GLOBE Version 1.0, ocean areas have been masked as "no data" and have been assigned a value of -500. Due to the nature of the raster structure of the DEM, small islands in the ocean less than approximately 1 square kilometer (specifically, those that are not characterized by at least one 30" grid cell and/or do not have coastlines digitized into Digital Chart of the World or World Vector Shoreline) may not be represented.

GLOBE's accuracy can be subdivided into horizontal and vertical accuracy, and again into absolute and relative accuracy. Horizontal accuracy can be affected by errors in horizontal positioning of features, or errors in recording horizontal datum. Horizontal accuracy may differ between sources. The absolute vertical accuracy of GLOBE varies by location according to the source data. Generally, areas derived from raster source data have higher accuracy than those derived from vector source data.

GLOBE is a data base, a data management philosophy, a working environment, and a file format. GLOBE began with the conceptual opening of a two-dimensional thirty-arc-second (30") latitude-longitude digital data array, and the hope to populate it with both the Best Available Data (B.A.D.), and the Globally Only Open-Access Data (G.O.O.D.). The former could include copyright data that might be made available for distribution by GLOBE with minimal restrictions, while the latter could not contain any restricted data. Allowing for both options has enabled GLOBE, for example, to work with the Australian Surveying and Land Information Group to develop a DEM much better than could otherwise be included while respecting the intellectual property rights of the Australian government.

The Global Land One-km Base Elevation Project (GLOBE) Task Team was established by the Committee on Earth Observation Satellites. It was part of Focus I of the International Geosphere-Biosphere Programme - Data and Information System.

References

Hastings, David A., and Paula K. Dunbar, 1999. Global Land One-kilometer Base Elevation (GLOBE) Digital Elevation Model, Documentation, Volume 1.0. Key to Geophysical Records Documentation (KGRD) 34. National Oceanic and Atmospheric Administration, National Geophysical Data Center, 325 Broadway, Boulder, Colorado 80303, U.S.A.

File Format

In addition to the GLOBE .DEM file, associated files include a source/lineage file. This source/lineage file provides a mask of ocean coverage (by using category 0 of the source/lineage file), so that the user may reassign the -500 flag values for ocean coverage to 0, then later re-separate those values from 0 values on land. This is 16-bit signed integer data in a simple binary raster. There are no header or trailer bytes embedded in the image. The data are stored in row major order (all the data for row 1, followed by all the data for row 2, etc.). All files have 10800 columns, and either 4800 or 6000 rows."

The digital elevation file format has two file types. Files .?10G and .?10B ("?" is the wildcard notation for tile letters "A" through "P") are provided as 16-bit signed integer data in a simple binary raster. There are no header or trailer bytes embedded in the image. The data are stored in row major order (all the data for row 1, followed by all the data for row 2, etc.). All files have 10800 columns, and either 4800 or 6000 rows.

The following diagram depicts the organization of the files:

```
bytes1/2.................................bytes21599/21600
bytes21601/21602.........................bytes43199/43200
.........................................................
.........................................................
etc.
.............................(last byte-1)/(last byte)
```

The data are in little-endian byte order (for IBM-compatible PCs, Digital Equipment VAXes, etc.) UNIX workstations using big-endian byte order can swap bytes using the command:

```
dd if=inputfilename of=outputfilename conv=swab
```

where "inputfilename" and "outputfilename" are replaced with the user's selection of input and output file names

Listed below is the projection information for each data file in GLOBE.

```
Projection       Geographic (latitude/longitude)
        Datum            WGS84
        Zunits           Meters above mean sea level
        Hunits           30 arc-seconds of latitude and longitude
        Spheroid         WGS84
        Xshift           0.0000000000
        Yshift           0.0000000000
        Cell Referencing Each cell is nominally bound by 30" intervals
                         of latitude and longitude, beginning with any
                         whole degree (e.g. 0.0000 degrees)
        Parameters       NONE other than those above
```

File Name Extensions

.DEM

Import Method

Choose the **File | Import** command.

Import Restrictions/Limitations

GLOBE DEM data .?10s files are not supported.

Esri ArcInfo Binary Grid .ADF File Description

The Esri ArcInfo Binary Grid filter imports and exports Esri Arc/Info binary grid files .ADF.

File Description

The Arc/Info Binary Grid format is the internal working format of the Arc/Info Grid product. This format should not be confused with the Arc/Info ASCII Grid format which is the interchange format for elevation grids. An Esri grid is a raster GIS file format developed by Esri, which has two formats:

1. A proprietary **binary** format, also known as an *ARC/INFO GRID*, *ARC GRID* and *many* other variations
2. A non-proprietary **ASCII** format, also known as an *ARC/INFO ASCII GRID*

The formats were introduced for ARC/INFO. The binary format is widely used within Esri programs, such as ArcGIS, while the ASCII format is used as an exchange, or export format, due to the simple and portable ASCII file structure.

File Description

A binary Esri grid is stored in several files contained in at least two directories: the *name* directory and an *info* directory, where *name* has strict naming conventions.

The grid defines geographic space as an array of equally sized square grid points arranged in rows and columns. Each grid point stores a numeric value that represents a geographic attribute (such as elevation or surface slope) for that unit of space. Each grid cell is referenced by its X, Y coordinate location.

A grid coverage (or data set) actually consists of a number of files. A grid normally lives in its own directory named after the grid. For example, the grid nwgrd1 lives in the directory nwgrd1, and has the following component files:

- nwgrd1/dblbnd.adf - file containing the bounds of the grid.
- nwgrd1/hdr.adf - header file
- nwgrd1/sta.adf - statistics file
- nwgrd1/vat.adf - value attributes file
- nwgrd1/w001001.adf - the actual raster data file
- nwgrd1/w001001x.adf - file containing file offset and size of each block in the raster data file.

Import Method

Choose the **File | Import** command.

Import Options

Specify import options in the **Lattice Import Options** dialog.

Export Method

Select a lattice or image module and choose the **File | Save Data** command.

Amira Mesh .AM, .COL Files

The Amira filter imports and exports Amira Mesh .AM, .COL files.

File Description

Amira Mesh files can contain uniform lattices, rectilinear lattices, curvilinear lattices, point data, or geometric figures.

File Name Extensions

.AM, .COL

Format(s) Supported for Import

- 3D uniform lattice; 8-, 16-, 32-bit integer, float, double
- 3D rectilinear lattice; 8-, 16-, 32-bit integer, float, double
- 3D curvilinear lattice; 8-, 16-, 32-bit integer, float, double
- point set; double

Format(s) Supported for Export

- 3D uniform lattice; 8-, 16-, 32-bit integer, float, double
- 3D rectilinear lattice; 8-, 16-, 32-bit integer, float, double
- 3D curvilinear lattice; 8-, 16-, 32-bit integer, float, double

Import Method

Choose the **File | Import** command.

Export Method

Select a module and choose the **File | Save Data** command.

Export Options

When a file contains blanked values, the Export Blanking Options dialog appears.

Import Restrictions/Limitations

Amira data sets that contain geometric figures are imported as point sets, e.g., "HxLineSet" and "Tetrahedra."

Export Restrictions/Limitations

The current implementation does not support exporting of geometric data.

ACR-NEMA Medical Image .AN? File Description

The AN? ACR-NEMA Medical Image filter imports images and lattices from DICOM 3 medical image data sets.

File Description

Digital Imaging and Communications in Medicine (DICOM) is a standard for handling, storing, printing, and transmitting information in medical imaging. It includes a file format definition and a network communications protocol. The communication protocol is an application protocol that uses TCP/IP to communicate between systems. DICOM files can be exchanged between two entities that are capable of receiving image and patient data in DICOM format. The National Electrical Manufacturers Association (NEMA) holds the copyright to this standard. It was developed by the *DICOM Standards Committee*, whose members are also partly members of NEMA.

DICOM enables the integration of scanners, servers, workstations, printers, and network hardware from multiple manufacturers into a picture archiving and communication system (PACS). The different devices come with DICOM conformance statements which clearly state the DICOM classes they support. DICOM has been widely adopted by hospitals and is making inroads in smaller applications like dentists' and doctors' offices.

File Format

DICOM differs from other data formats in that it groups information into data sets. That means that a file of a chest X-Ray image, for example, actually contains the patient ID within the file, so that the image can never be separated from this information by mistake.

A DICOM data object consists of a number of attributes, including items such as name, ID, etc., and also one special attribute containing the image pixel data (i.e. logically, the main object has no "header" as such - merely a list of attributes, including the pixel data). A single DICOM object can only contain one attribute containing pixel data. For many modalities, this corresponds to a single image. But note that the attribute may contain multiple "frames", allowing storage of cine loops or other multi-frame data.

DICOM uses three different Data Element encoding schemes. With Explicit VR Data Elements, for VRs that are not OB, OW, OF, SQ, UT, or UN, the format for each Data Element is: GROUP (2 bytes) ELEMENT (2 bytes) VR (2 bytes) LengthInByte (2 bytes) Data (variable length). For the other Explicit Data Elements or Implicit Data Elements, see section 7.1 of Part 5 of the DICOM Standard.

The same basic format is used for all applications, including network and file usage, but when written to a file, usually a true "header" (containing copies of a few key attributes and details of the application which wrote it) is added.

File Name Extensions

DICOM .DIC, .DCM and ACR-NEMA .AN1, AN2.

Format(s) Supported for Import

- device-independent bitmap; 8, 24, 32 bit per pixel
- uniform lattice; 8-, 16-, 32-bit integer, float, double

Import Method

Choose the **File | Import** command.

Import Options

Specify import options in the **DICOM Import Options** dialog.

Import Restrictions/Limitations

The DICOM specification allows an unusually wide variety of different formats and encodings within the same file format. While this software can read most of the common variants of DICOM, it would not be practical to develop software to read every possible variant. Some of the known deficiencies in this implementation include:

- DICOM images that contain bit per pixel counts other than 8, 12, 16, 24 or 32 may not be readable depending on the encoding and alignment of the data.
- DICOM images that are encoded with photometric interpretation models other than RGB, grayscale, or monochrome may not be readable. In particular, some YUV encodings cannot be imported.
- Some lossless JPEG images embedded in DICOM data sets do not import. In particular, images encoded with the "Cornell" JPEG codec are not always readable.
- Some of the obscure compression algorithms allowable under the DICOM specification are not supported by this software.
- Some ACR-NEMA files do not import.
- This filter is also able to read some files written in the obsolete ACR-NEMA format (from which the DICOM format was derived); however, Golden Software does not officially support the obsolete ACR-NEMA file formats; however, **Voxler** does import many ACR-NEMA files successfully.

Arc/Info ASCII Grid File .ASC, .AIG, .AGR, .GRD Description

The ASC Arc/Info ASCII filter imports and exports ASC Arc/Info ASCII grid files.

File Description

An Esri grid is a raster GIS file format developed by Esri. The ASCII format is also known as Arc/Info ASCII grid. The ASCII format is used as an elevation grid exchange, or export format, with a simple ASCII file structure.

File Format

The first six lines of the file indicate the reference of the grid, followed by the values listed in the order they would appear (left to right and top to bottom).

Example

```
ncols         4
nrows         6
xllcorner     0.0
yllcorner     0.0
cellsize      50.0
NODATA_value -9999
 -9999 -9999 5 2
-9999 20 100 36
3 8 35 10
32 42 50 6
88 75 27 9
13 5 1 -9999
```

Where:

- *ncols* are the numbers of rows (represented as integers);
- and *nrows* are the numbers of rows (represented as integers);
- *xllcorner* and *yllcorner* are the western (left) X coordinate and southern (bottom) Y coordinates, such as easting and northing (represented as real numbers with an optional decimal point)
- *cellsize* is the length of one side of a square cell (a real number)
- *nodata_value* is the value that is regarded as "missing" or "not applicable"; this line is optional, but highly recommended as some programs expect this line to be declared (a real number).

The remainder of the file lists the raster values for each cell, starting at the upper-left corner. These real numbers (with optional decimal point, if needed) and are delimited using a single space character.

File Name Extensions

.ASC, .AIG, .AGR, .GRD

Import Method

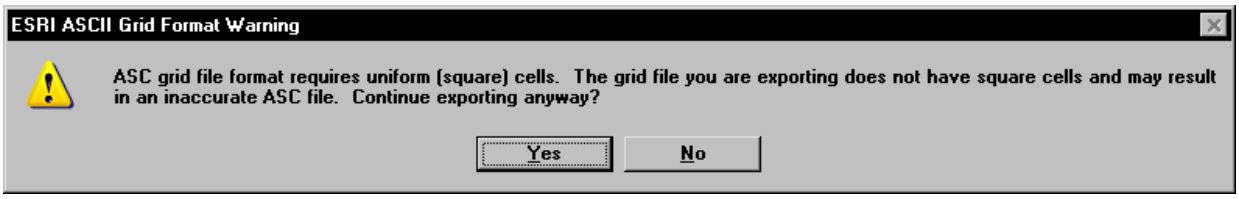
Choose the **File | Import** command.

Export Method

Select a lattice or image module and choose the **File | Save Data** command.

ESRII ASCII Grid Format Warning Dialog

When exporting, the **ESRII ASCII Grid Format Warning** dialog may appear.



The ESRII ASCII Grid Format Warning dialog

Click the *Yes* button to continue export. Click the *No* button to abort the export.

Amira Stacked Images .ASI File Description

The STACKED filter imports bitmap images in the Amira Stacked Slices format.

File Description

Amira stacked images allow a stack of individual image files to be read, with optional z-values for each slice. One or more bitmap images are imported from separate files as specified by a description file. The description file can be created by the user or by another application.

File Format

The description file is an ordinary text file. Any lines in the file that begin with a pound (#) character are assumed to be comments and are ignored. Any other lines must adhere to the format described below.

Typically the file begins with a comment line that reads

Amira Stacked Slices

Next comes the optional "pathname" line. If present, this line specifies the pathname of the directory where the image files are located. If the pathname line is not present, the software assumes the image files are located in the same directory as the description file. The image files to which the description file refers may include any of the bitmap image file formats supported by the other **Voxler** filters, such as .JPEG, .BMP, .PNG, .TGA, and so on.

Example 1

pathname c:\MyPictures

Next comes the optional "pixelsize" line. If present, this line specifies the spacing of the lattice nodes for each pixel in application units.

Example 2

pixelsize 0.1. 0.1

If the pixelsize line is not present, the default size is one unit per pixel.

Next comes the image list consisting of one or more lines, each of which specifies the file name of an image file to load. Each of these lines can also optionally specify an application Z position for each image slice.

Example 3

picture1.jpg 10.0

picture2.bmp 11.0

picture3.jpg 12.0

picture4.tif 15.0

The above example loads four image slices positioned at 10, 11, 12, and 15 units along the Z axis.

If the z positions are not given with the image filenames, the default z positioning, which uses the larger of the x pixel size and y pixel size as the amount to space between each slice on the z axis, is used.

If the z positions are uniformly spaced, the imported lattice is of the "uniform" lattice type.

Finally, the file ends with a line containing the word "end."

Example 4: Description File

Amira Stacked Slices

Directory where image files live

pathname c:\MyPictures

Pixel Size

pixelsize 0.5 0.5

Image List

slice1.jpg 1.6

slice2.jpg 1.8

slice3.jpg 2.0

Mark the end of the description file

end

The image files to which the description file refers may include any of the image file formats supported by the other filters, such as .JPG, .BMP, .PNG, .TGA, and so on.

File Name Extensions

The recommended (though not mandatory) file name extension is .ASI.

Format(s) Supported for Import

- 3D uniform lattice; integer data
- 3D rectilinear lattice; integer data

Import Method

Choose the **File | Import** command.

Import Restrictions/Limitations

The images referenced by any particular description file must all be of the same width and height (in pixels) and have the same color depth (in bits per pixel). If any referenced image differs in dimensions or color depth from the others, an error message displays and the import does not complete.

Band Interleaved .BIL, .BIP, .BSQ File Description

The Band Interleaved filter imports and exports lattice data as .BIL, .BIP, and .BSQ raster data sets.

File Description

Band Interleaved by line .BIL format is a common format for distributing satellite image data and organizing image data for multiband images. Band Interleaved by Pixel .BIP, and Band Sequential .BSQ. The .BIL, .BIP, and .BSQ are formats produced by remote-sensing systems. The primary difference among them is the technique used to store brightness values captured simultaneously in each of several colors or spectral bands. The data values are stored in a simple binary raster.

The .BIL, .BIP, and .BSQ files are not in themselves image formats but are schemes for storing the actual pixel values of an image in a file. These files support the display of single and multiband images. The .BIL file is an uncompressed file containing the actual pixel values of an image. It can handle black and white, grayscale, pseudocolor, true color, and multi-spectural image data. BIL data stores pixel information in separate bands within the file, so that the user can choose to display just one specific band in a multi-band image.

File Format

The .BIL, .BIP, and .BSQ files are binary files, and they must have an associated ASCII header .HDR file. This header file must have the same filename as the .BIL, .BIP or .BSQ, and only its file extension differs. This header file contains data about the image such as the number of rows and columns in the image and latitude and longitude. There are no header or trailer bytes imbedded in the image. The data are stored in row major (all the data for row 1, followed by all the data for row 2, etc.). A header file is an ASCII text file containing the size and coordinate information for the raster data.

There are four image description files (ASCII text file format) that can accompany the BIL data: a header .HDR file, a statistics .STX file, a resolution file .BLW, and a color file .CLR

.

The header file .hdr describes the nature of the image data, through the use of keywords and values.

The statistics file .stx is an optional file that describes the image statistics for each spectral band. It records the minimum and maximum pixel values, the mean, the standard deviation, and the two linear contrast stretch parameters.

The resolution file .blw describes the height and width of each cell and the coordinate position of the top left cell of the data. It gives the Resolution information with the following lines:

resolution in x-dimension
rotation factor (always zero for GPW)
rotation factor (always zero for GPW)
resolution in y-dimension
x-coordinate of the center of the upper-left cell
y-coordinate of the center of the upper-left cell

The color file (.clr) is an optional file that describes the image colormap (this file type is not used in the GPW data).

File Name Extensions

.BIL, .BIP, and .BSQ

Import Method

Choose the **File | Import** command.

Import Options

Specify the import options in the **Lattice Import Options** dialog.

Export Method

Select a lattice or image module and choose the **File | Save Data** command.

Golden Software Blanking .BLN File Description

The Golden Software Blanking filter imports Golden Software Blanking Files .BLN. **Voxler** honors Z values when importing a 3D BLN file.

File Description

Golden Software Blanking File .BLN is an ASCII format file used to store geographic information including areas, curves, and points. Spatial information is only concerned with the location of objects in space (i.e., their coordinates) and not with their attributes (i.e. line or fill style, marker symbol used, text labels, etc.). Even though the primary use of GS Blanking files is to indicate regions to be "blanked-out", they can also be used for simple boundaries and decorative illustrations.

File Format

The general format of the file is:

length,flag "Pname 1"

x1,y1,z1

x2,y2,z2

...
xn,yn,zn
length,flag "Pname 2"
x1,y1,z1
x2,y2,z2
...
xn,yn,zn

Where:

Length

The *length* value is an integer which indicates the number of X, Y coordinate pairs that follow.

Flag

The *flag* value is 1 if the region inside areas is to be blanked and 0 if the region outside areas is to be blanked.

Pname

Pname is optional and is the name of a primary ID to be associated with the object. The primary ID is used to link the object to external data.

X, Y, and Z Coordinates

Following lines contain the actual X, Y coordinate pairs (or X, Y, Z coordinate triads) that make up the object. These can be integers or real numbers, and are stored 1 pair/triad per line.

Type of Object

The type of object is determined as follows:

- If the *type*/*length* field is 1, the object is considered a point. One coordinate pair follows.
- If the *type*/*length* field is greater than 1 and the first and last coordinate pairs are equal, the object is considered a simple closed area. Otherwise, the object is considered a curve.

File Name Extensions

.BLN

Import Method

Choose the **File | Import** command to import the BLN file into the **Network Manager** as a *Geometry Source* module. Choose **File | Open** to open the BLN file in the worksheet.

Export Method

Choose the **File | Save Data** command for a module in a **Voxler** project or the **File | Save As** command for data in the worksheet.

Export Options

See *BLN Golden Software Blanking Export Options Dialog*

Bitmap .BMP Files

The Bitmap filter imports and exports Windows bitmap images. When loading bitmap files with coordinates, the coordinate information comes from a geotiff (internal information) or external files.

File Description

In computer graphics, a bitmap is a type of memory organization or image file format used to store digital images. The term *bitmap* comes from the computer programming terminology, meaning a *map of bits*, a spatially mapped array of bits.

Used as the standard bitmap storage format in the Microsoft Windows environment. Although it is based on Windows internal bitmap data structures, it is supported by many non-Windows and non-PC applications.

File Name Extensions

.BMP

Format(s) Supported for Import

device-independent bitmap; 1, 4, 8, 24 bit per pixel

Format(s) Supported for Export

device-independent bitmap; 8, 24 bit per pixel

Import Method

Choose the **File | Import** command.

Export Method

Choose the **File | Export** command or select an image module and choose the **File | Save Data** command.

Export Options

Specify the export options in the **Export Options** dialog, **Size and Color** page.

Atlas Boundary .BNA File Description

Voxler imports and exports Atlas Boundary files .BNA.

The Atlas Boundary File .BNA is an ASCII format file used to store spatial information including areas, curves, ellipses and points. Spatial information is only concerned with the location of objects in space (i.e., their coordinates) and not with their attributes (such as line or fill style, marker symbol used, text labels, etc.).

File Format

The general format of the file is:

"Pname 1", "Sname 1", type/length
x1,y1
x2,y2
...
xn,yn
"Pname 2", "Sname 2", type/length
x1,y1
x2,y2
...
xn,yn

Pname

Pname is the name of the primary ID. The primary ID is used to link the object to external data.

Sname

Sname is the name of the secondary ID. The secondary ID is optional.

Type/Length

The *type/length* is an integer which identifies the object as an area, curve, ellipse or point.

Following the *type/length* are the actual X,Y coordinate pairs that make up the object. These can be integers or real numbers, and are stored 1 pair per line.

The *type/length* field indicates the number of coordinate pairs to follow and also indicates the type of object as follows:

- Areas have a *type/length* value greater than 2. The value indicates the number of coordinate pairs to follow. Islands and lakes are concatenated to the coordinate list.
- Curves have a *type/length* value less than -1. The absolute value is the number of coordinates to follow for the curve.
- Ellipses have a *type/length* value 2. The first pair of coordinates describe the center of the ellipse. The major and minor radii are stored in the second pair of coordinates. If the minor radius is 0, the ellipse is a circle.
- If the *type/length* field is 1, the object is considered a point. One coordinate pair follows.

Simple and Compound Areas

Two kinds of areas exist, simple and compound. A simple area contains a starting point, a series of points specifying the area's boundary and a closing point with the same coordinate as the starting point. A compound area contains one or more subareas, such as islands or lakes. Atlas Boundary files use a special technique to specify the subareas comprising compound areas.

Example

There is a compound area consisting of a closed outer area "A" and two islands "B" and "C". Here is how the coordinates should be specified in an Atlas Boundary file:

AX1,AY1 Starting point of area "A"

AX2,AY2	Points specifying boundary of area "A"
.	
.	
AXn,AYn	Ending point of area "A"
BX1,BY1	Starting point of subarea "B"
BX2,BY2	Points specifying boundary of subarea "B"
.	
.	
BXn,BYn	Ending point of subarea "B"
AX1,AY1	Staring point of area "A" (Flag Point)
CX1,CY1	Starting point of subarea "C"
CX2,CY2	Points specifying boundary of subarea "C"
.	
.	
CXn,CYn	Ending point of subarea "C"
AX1,AY1	Staring point of area "A" (Flag Point)

Each area's ending point must have the same coordinate as its starting point. The staring point of area "A" is used as a marker (called a Flag Point) to indicate the end of each subarea. This means the first area point's coordinate must be unique and cannot appear as a coordinate within any subarea.

Import Options Dialog

No import dialog is displayed.

Import Automation Options

See *BNA Import Options*

Export Options Dialog

No export dialog is displayed.

Export Automation Options

See *BNA Export Options*

Color Spectrum CLR File Format

Color spectrum files .CLR are used to define a spectrum, or continuous gradation of colors. This is specified by a series of anchor points with associated colors. The colors between anchor points are interpolated from the nearest anchor points.

The basic format consists of an ASCII file with header information on the first line. Subsequent lines specify anchors points, one anchor point per line.

The header consists of the following space-delimited fields:

Element	Description
Id	case-sensitive string "ColorMap" without the quotes
Version	format version number, this should be set to 1 or 2
InterpMethod	interpolation method between anchors, set to 0
ColorNodes	the number of color nodes
OpacityNodes	the number of opacity nodes

Subsequent lines define the anchor points, one per line. Each line in the Color section has the following space-delimited fields:

Element	Description
Position	The position is the floating point value proportion of the colormap (from 0 to 1). Positions must be specified in increasing order from 0 to 1, and the 0 and 1 positions must be specified in the file.
Red	red color component (0 to 255)
Green	green color component (0 to 255)
Blue	blue color component (0 to 255)

Each line in the Opacity section has the following space-delimited fields:

Element	Description
Position	The position is the floating point value proportion of the colormap (from 0 to 1). Positions must be specified in increasing order from 0 to 1, and the 0 and 1 positions must be specified in the file.
Opacity	The opacity is the floating point value corresponding to the opacity at the node (from 0 to 1)

Examples

In the following example, the anchor points are at 0, 50, and 100. The zero position is a slightly transparent blue, the 50 percent anchor is mostly transparent green, and the 100 percent position is fully opaque yellow.

```
ColorMap 2 0 3 3
0    0     0     255
.5   0     255   0
1    255   255   0
0                    0.78431372549020
.5                   0.07843137254902
1                    1
```

It is also possible to have coincident anchor points in a color file. Anchors and colors are interpreted in order from 0 percent to 100 percent. In the case of coincident points, you can create maps with distinct boundaries, similar to the example shown here. This colormap has a fixed 0.2 Opacity.

```
ColorMap 2 0 4 2
0     255    0      0
50    255    255    0
50    0      0      255
100   255    255    255
0                          0.2
1                          0.2
```

The following example shows the format of the Rainbow colormap with a fixed 1.0 Opacity:

```
ColorMap 2 0 6 2
0     153    102    255
.2    0      0      255
.4    0      255    0
.6    255    255    0
.8    255    102    0
1     255    0      0
0                          1
1                          1
```

CPS-3 Grid .ADX, .DAT, .GRD, .CPS File Description

The CPS-3 Grid filter imports and exports CPS-3 ASCII grid files.

File Description
The CPS-3 Grid is an industry standard ASCII format.

File Format
The data are written a column at a time from the left side to the right side of the grid.

File Name Extensions
The format has several file extensions: .ASC, .DAT, .GRD, .CPS, and .CPS3.

Import Method
Choose the **File | Import** command.

Import Options
Specify import options in the **Lattice Import Options** dialog.

Import Restrictions/Limitations
This import filter only supports the GRID type of the CPS formats.

Export Method

Select a lattice or image module and choose the **File | Save Data** command.

DAT .CSV, . DAT, . TXT File Description

The DAT filter imports and exports tabular data from delimited text files.

These file formats are assumed to have one record per line in which each record contains a fixed number of numeric data fields. The method of delimiting the fields within each record is specified through the controls in the **Data Import Options** and **Data Export Options** dialog. The comma separated value .CSV, data .DAT, and text .TXT files are used for the digital storage of data structured in a table of lists. A comma separated value .CSV file is a delimited data format that has fields/columns separated by the comma character and records/rows separated by newlines.
The .TXT is a filename extension for files consisting of text usually contain very little formatting (ex: no **bolding** or *italics*). Files with the .TXT extension can easily be read or opened by any program that reads text and, for that reason, are considered universal (or platform independent).

Worksheet Formatting

ASCII files do not contain any worksheet formatting information such as row height, column width, or cell formatting. When ASCII files are loaded into the worksheet, the default column formatting parameters are applied to the data. This does not result in any change to data, but might result in rounding of values in the data display. There is no limitation on the number of rows or columns in an ASCII format. ASCII formats save and load slowly because there is a conversion from binary numbers to character representation.

Format

There are some distinctions in formatting of ASCII files. Here are some brief notes that outline the usefulness of the ASCII file features.

- *Delimiters* control the separation between cell entries in a file. Spaces, tabs, semi-colons, or commas can be used to separate cells. If cell entries contain spaces in text, the comma or semi-colon delimiters are necessary if quotes are not used to qualify the text. Otherwise, the text string would be interpreted as two cell entries rather than a single entry.
- *Placing Quotes Around Text* - There are two types of entries in an ASCII file, values and text. Values are actual numbers, while text can be any type of character, including numbers and text characters. Single or double quotes can be placed around text strings. If a number should be interpreted as text, surround it with double quotes. When text strings contain spaces, it is recommended to use single or double quotes around text cell entries.
- *Using Commas or Semicolons in Addition to Quotes* - Although double quotes are not required around text strings, they are useful when creating a space-delimited file that contains text. Often there are text strings that contain spaces, as in a date containing month name, day and year. With space delimited files this single entry is interpreted as more than one cell when loading this file into the worksheet. The safest way to eliminate this problem is to place double quotes around all text strings and use comma delimiting between variables.

Comma Separated Variables

Comma separated variable .CSV files are comma delimited with double-quotes around text strings (non-numeric or mixed alpha numeric).

When the computer's locale setting has the *Decimal separator* as *comma*, .CSV files are imported with commas as the decimal separators and semi-colons as column separators. If the CSV delimiter and the computer's locale delimiter are not the same the CSV will import incorrectly. Use the following process to correctly import the data into **Voxler** for use in the worksheet or project.

1. Open the CSV in a text editor application, such as Windows Notepad.
2. Press CTRL+A to select all the text in the file.
3. Copy the selected text to the clipboard. Press CTRL+C or use the text editor's **Copy** command.
4. In **Voxler**, click the **File | New | Worksheet** command.
5. Select the top-left cell in which the data should be pasted, usually A1.
6. Click the **Edit | Paste Special** command.
7. In the **Paste Special** dialog, select the *Unicode Text [Clipboard]* or *Text [Clipboard]* option, and make sure the *Show Import Options* box is checked.
8. Click *OK* in the **Paste Special** dialog.
9. In the **Data Import Options** dialog, change the selection in the *Delimiters* section and the *Use comma as decimal symbol* check box until the data are displayed correctly in the *Preview* section.
10. Click *OK* in the **Data Import Options** dialog.
11. Click the **File | Save As** command to save the reformatted data to a new data file.

Alternatively you can change the computer's locale setting to match that of the CSV file, and then open the file in a new instance of **Voxler**.

ASCII Text

ASCII text files .TXT are normally tab delimited ASCII text files with no quotes around the text strings. After selecting .TXT as the format, the **Data Export Options** dialog is displayed.

Golden Software DAT Files

ASCII .DAT files are ASCII files with no set format. Any delimiter or text qualifier can be set. When a file is saved in the .DAT format, the **Data Export Options** dialog is displayed.

File Name Extensions

.CSV, . DAT, . TXT

Format(s) Supported for Import

tabular data; double

Format(s) Supported for Export

tabular data; double

Import Method

Choose the **File | Import** command.

Import Options

Specify import options in the **Data Import Options** dialog.

Export Method

Select a data or lattice module and choose the command to save point and tabular data to an ASCII data format.

Export Options

These file formats are assumed to have one record per line in which each record contains a fixed number of numeric data fields. The method of adding delimiters to the fields within each record is specified through the controls in the **Data Export Options** dialog. The **Data Export Options** dialog allows you to specify *Comma*, *Tab*, *Space*, or *Semicolon* as the character to use to delimit fields.

ASCII Database .DBF File Description

The ASCII Database filter imports data from dBase/xBase database .DBF files.

File Description

Xbase is a complex of data files .DBF, indexes .NDX, .MDX, .CDX, etc. and eventually note files .DBT for storing large amounts of formatted data in a structured form.

File Format

DBase's database system was one of the first to provide a header section for describing the structure of the data in the file. This meant that the program no longer required advance knowledge of the data structure, but rather could ask the data file how it was structured. Note that there are several variations on the .DBFfile structure, and not all dBase-related products and .DBF file structures are necessarily compatible.

A second filetype is the .DBT file format for memo fields. While character fields are limited to 254 characters each, a memo field is a 10-byte pointer into a .DBT file which can include a much larger text field. DBase was very limited in its ability to process memo fields, but some other xBase languages treat memo fields as strings just like character fields for all purposes except permanent storage.

DBase uses .NDX files for indexes. Some xBase languages include compatibility with .NDX files while others use different file formats.

File Name Extensions

.DBF, .NDX, .MDX, .CDX, .DBT

Import Options

Choose the **File | Import** command.

SDTS DEM .DDF Files

The SDTS filter imports uniform lattices from several two-dimensional elevation grid file formats.

There are two separate types of SDTS DDF files. The Topological Vector Profile SDTS files contain boundary information. The Raster Profile SDTS files contain gridded elevation information.

File Description

The Spatial Data Transfer Standard, or SDTS, is a robust way of transferring earth-referenced spatial data between dissimilar computer systems with the potential for no information loss. It is a transfer standard that embraces the philosophy of self-contained transfers, i.e. spatial data, attribute, georeferencing, data quality report, data dictionary, and other supporting metadata all included in the transfer.

File Name Extensions

SDTS DEM .DDF, .TAR, .TAR.GZ, .ZIP, or .TGZ

Voxler can read the information in the various files directly from the compressed file. You do not need to unzip the compressed file.

If you do unzip the .TAR.GZ, .TAR, .ZIP, or .TGZ file containing the DDF files, there is an option in WinZip (or other unzipping software) that needs to be disabled. Use the *WinZip settings* to disable this option. All .DDF files must be extracted into the same directory.

1. In WinZip, use the **Options | Configuration** command to open the **Configuration** dialog.
2. Click on the *Miscellaneous* tab.
3. In the Other category, uncheck TAR file smart CR/LF conversion.
4. Click the *OK* button.
5. The Tar.GZ file will now properly unzip the files.

Format(s) Supported for Import

2D uniform lattice; 8-, 16-, 32-bit integer, float, double

Import Method

Choose the **File | Import** command.

Import Options

Specify import options in the **Lattice Import Options** dialog.

Other Notes

SDTS data sets can contain elevation data or several other types of data that are not applicable to **Voxler**. If you attempt to import an SDTS data set that does not contain elevation data, an error message displays.

References

http://mcmcweb.er.usgs.gov/sdts/whatsdts.html

SDTS Topological Vector Profile and Raster Profile .TVP, .DDF File Description

The SDTS Topological Vector Profile and Raster Profile filter imports USGS SDTS Topological Vector Profile .TVP or .DDF data sets.

File Description

The Spatial Data Transfer Standard, or SDTS, is a robust way of transferring earth-referenced spatial data between dissimilar computer systems with the potential for no information loss. It is a transfer standard that embraces the philosophy of self-contained transfers, i.e. spatial data, attribute, georeferencing, data quality report, data dictionary, and other supporting metadata all included in the transfer.

The purpose of the SDTS is to promote and facilitate the transfer of digital spatial data between dissimilar computer systems, while preserving information meaning and minimizing the need for information external to the transfer. Implementation of SDTS is of significant interest to users and producers of digital spatial data because of the potential for increased access to and sharing of spatial data, the reduction of information loss in data exchange, the elimination of the duplication of data acquisition, and the increase in the quality and integrity of spatial data. SDTS is neutral, modular, growth-oriented, extensible, and flexible--all characteristics of an "open systems" standard.

The SDTS provides a solution to the problem of spatial data transfer from the conceptual level to the details of physical file encoding. Transfer of spatial data involves modeling spatial data concepts, data structures, and logical and physical file structures. To be useful, the data to be transferred must also be meaningful in terms of data content and data quality. SDTS addresses all of these aspects for both vector and raster data structures.

File Format

There are two separate types of SDTS DEM files: topological vector profile SDTS and raster profile SDTS files.

The topological vector profile SDTS .TVP files contain boundary line information and can be used as a base map.

The raster profile SDTS .DDF files contain gridded elevation information and can be used to directly create a grid-based map or to perform grid based functions. An uncompressed SDTS contains several files with the .DDF extension. All of the .DDF files are necessary to produce a map (i.e. you cannot copy just one .DDF file and create a map from it). You can choose any of the .DDF files when importing a base map with the Topological Vector Profile SDTS files.

File Name Extensions

.TVP, .DDF, .TAR, .TAR.GZ, .ZIP, or .TGZ

Voxler can read the information in the various files directly from the compressed file. You do not need to unzip the compressed file.

If you do unzip the .TAR.GZ, .TAR, .ZIP, or .TGZ file containing the DDF files, there is an option in WinZip (or other unzipping software) that needs to be disabled. Use the *WinZip settings* to disable this option. All .DDF files must be extracted into the same directory.

1. In WinZip, use the **Options | Configuration** command to open the **Configuration** dialog.
2. Click on the *Miscellaneous* tab.
3. In the Other category, uncheck TAR file smart CR/LF conversion.
4. Click the *OK* button.
5. The Tar.GZ file will now properly unzip the files.

Import Method

Choose the **File | Import** command.

Import Options Dialog

Specify the import options in the **SDTS TVP Import Options** dialog.

SDTS Digital Elevation Model .DEM File Description

The SDTS Digital Elevation Model filter imports elevation data in the form of a uniform lattice from SDTS Digital Elevation Model .DEM data sets.

File Description

There are two separate types of SDTS DEM files. The Topological Vector Profile SDTS files contain boundary line information.

The Raster Profile SDTS files contain gridded elevation information and can be used to directly create a grid-based map such as a contour map. An uncompressed SDTS contains several files with the .DDF extension. All of the .DDF files are necessary to produce a map (i.e. you cannot copy just one .DDF file and create a map from it). You can choose any of the .DDF files when importing a base map with the Topological Vector Profile SDTS files.

File Name Extensions

.DDF

Format(s) Supported for Import

2D uniform lattice; 8-, 16-, 32-bit integer, float, double

Import Method

Choose the **File | Import** command.

Remarks

SDTS data sets can contain elevation data or several other types of data that are not applicable. If you attempt to import an SDTS data set that does not contain elevation data, an error message displays.

USGS Digital Elevation Model .DEM File Description

The USGS Digital Elevation Model filter imports and exports uniform lattices from several two-dimensional elevation grid file formats.

File Description

The .DEM is a grid file and can be used to create grid based maps such as contour maps, 3D surface maps, etc.

The USGS DEM files have been replaced with the USGS SDTS files. Older file type .DEM files containing arc-second XY coordinates are automatically converted to latitude/longitude decimal degrees.

The USGS produced two DEM file types: the 7.5 minute DEM data and the 1:250,000-scale DEM data.

- The 7.5 minute DEMs correspond to the standard 1:24,000 scale quadrangles. Elevation points are provided along north-south profile lines, and use a 30 meter spacing between adjacent data points and adjacent profile lines. Point locations are referenced to the UTM (Universal Transverse Mercator) coordinate system. The 7.5 minute quadrangles are usually not oriented exactly north-south. The DEM polygon contained within a 7.5 minute quadrangle consists of profiles that do not always have the same number of data points because of the variable angle between the UTM coordinate system and true north. This results in the potential for a regular, stair-stepped arrangement of the data points along the edges of the map which can translate to blanked grid nodes along the edges of grid files produced. This can lead to discontinuous contour lines along the edges of contour maps, or blanked edges on surface plots.
- The 1:250,000 DEMs cover a 1 degree by 1 degree block representing one-half of a 1 degree by 2 degree 1:250,000-scale map. Point locations are based on the latitude/longitude coordinate system. Elevation points are provided along north-south profile lines, and use a 3 arc-second spacing between each data point and between profile lines. Three arc seconds correspond to approximately 90 meters in the north-south direction and variable spacing in the east-west direction due to convergence of meridians as latitude increases (approximately 90 meters at the equator and approximately 60 meters at 50 degrees latitude).

File Name Extensions

.DEM

Format(s) Supported for Import

2D uniform lattice; 8-, 16-, 32-bit integer, float, double

Import Method

Choose the **File | Import** command.

Import Options

Specify import options in the **Lattice Import Options** dialog.

Export Method

Select a lattice or image module and choose the **File | Save Data** command.

Export Options Dialog

Specify export options in the **USGS Digital Elevation Model .DEM Export Options** dialog.

Remarks

The USGS header may contain some fields that are not supported.

The elevation data contained in USGS DEM files are regularly spaced, but the direction of the grid lines may not exactly coincide with the grid edges. This is due to the variable angle between the UTM coordinate system (used in some DEM files) and true north. This results in the potential for a

regular, stair-stepped arrangement of the grid nodes along the edges of the map which can translate to blanked grid nodes along the edges of maps produced. If the DEM file contains three-second arc coordinates, the coordinates are automatically converted to latitude/longitude coordinates.

SDTS data sets can contain elevation data or several other types of data that are not applicable to **Voxler**. If you attempt to import an SDTS data set that does not contain elevation data, an error message displays.

DICOM Medical Image .DIC, .DCM, .AN1, .AN2 Files

The DICOM filter imports bitmap images and uniform lattices from DICOM 3 medical image data sets. This filter is also able to read some files written in the obsolete ACR-NEMA format (from which the DICOM format was derived); however, Golden Software does not officially support the ACR-NEMA format.

File Description

Digital Imaging and Communications in Medicine (DICOM) is a standard for handling, storing, printing, and transmitting information in medical imaging. It includes a file format definition and a network communications protocol. The communication protocol is an application protocol that uses TCP/IP to communicate between systems. DICOM files can be exchanged between two entities that are capable of receiving image and patient data in DICOM format. The National Electrical Manufacturers Association (NEMA) holds the copyright to this standard. It was developed by the *DICOM Standards Committee*, whose members are also partly members of NEMA.

DICOM enables the integration of scanners, servers, workstations, printers, and network hardware from multiple manufacturers into a picture archiving and communication system (PACS). The different devices come with DICOM conformance statements which clearly state the DICOM classes they support. DICOM has been widely adopted by hospitals and is making inroads in smaller applications like dentists' and doctors' offices.

File Format

DICOM differs from other data formats in that it groups information into data sets. That means that a file of a chest X-Ray image, for example, actually contains the patient ID within the file, so that the image can never be separated from this information by mistake.

A DICOM data object consists of a number of attributes, including items such as name, ID, etc., and also one special attribute containing the image pixel data (i.e. logically, the main object has no "header" as such - merely a list of attributes, including the pixel data). A single DICOM object can only contain one attribute containing pixel data. For many modalities, this corresponds to a single image. But note that the attribute may contain multiple "frames", allowing storage of cine loops or other multi-frame data.

DICOM uses three different Data Element encoding schemes. With Explicit VR Data Elements, for VRs that are not OB, OW, OF, SQ, UT, or UN, the format for each Data Element is: GROUP (2 bytes) ELEMENT (2 bytes) VR (2 bytes) LengthInByte (2 bytes) Data (variable length). For the other Explicit Data Elements or Implicit Data Elements, see section 7.1 of Part 5 of the DICOM Standard.

The same basic format is used for all applications, including network and file usage, but when written to a file, usually a true "header" (containing copies of a few key attributes and details of the application which wrote it) is added.

File Name Extensions

.DIC; DICOM .DCM; and ACR-NEMA .AN1, AN2

Format(s) Supported for Import

- device-independent bitmap; 8, 24, 32 bit per pixel
- uniform lattice; 8-, 16-, 32-bit integer, float, double

Import Method

Choose the **File | Import** command.

Import Options Dialog

Specify import options in the **DICOM Import Options** dialog and the **Lattice Import Options** dialog.

Import Restrictions/Limitations

The DICOM specification allows an unusually wide variety of different formats and encodings within the same file format. While this software can read most of the common variants of DICOM, it would not be practical to develop software to read every possible variant. Some of the known deficiencies in this implementation include:

- DICOM images that contain bit per pixel counts other than 8, 12, 16, 24 or 32 may not be readable depending on the encoding and alignment of the data.
- DICOM images that are encoded with photometric interpretation models other than RGB, grayscale, or monochrome may not be readable. In particular, some YUV encodings cannot be imported.
- Some lossless JPEG images embedded in DICOM data sets do not import. In particular, images encoded with the "Cornell" JPEG codec are not always readable.
- Some of the obscure compression algorithms allowable under the DICOM specification are not supported by this software.
- Some ACR-NEMA files do not import. Golden Software does not officially support the obsolete ACR-NEMA file formats; however, **Voxler** does import many ACR-NEMA files successfully.

USGS Digital Line Graph .DLG File Description

The USGS Digital Line Graph filter imports .DLG files.

File Description

The United States Geological Survey (USGS) provides digitized base map data in "line graph" form. It is available in two formats ("standard" and "optional") on either 9-track magnetic tape or CD-ROM. The CD-ROM with 1:2,000,000-scale DLG files contains data in both standard and optional formats, as well as a "graphic" format. The definitive guide to these file formats is the USGS document: "Digital Line Graphs from 1:24,000-Scale Maps: Data Users Guide 1", "Digital Line Graphs from 1:100,000-Scale Maps: Data Users Guide 2" and "Digital Line Graphs from 1:2,000,000-Scale Maps: Data Users Guide 3".

File Format

Imported DLG files are read in the "standard" and "optional" formats, and provides point, area, and curve objects.

For the 1:2,000,000-scale files, the USGS has divided the U.S. into 21 sections. On the CD-ROM, the files associated with each section are stored in a separate subdirectory.

The 21 subdirectories use the following naming convention:

SECT00

Each subdirectory starts with the letters "SECT" followed by the section number (01 to 21). The sections correspond to the following regions:

SECT01	Northeastern States
SECT02	Middle Atlantic States
SECT03	Southeastern States
SECT04	Florida
SECT05	Southern Mississippi Valley States
SECT06	Central Mississippi Valley States
SECT07	Northern Great Lakes States
SECT08	Southern Texas
SECT09	Southern Plains States
SECT10	Central Plains States
SECT11	Northern Plains States
SECT12	Arizona and New Mexico
SECT13	Southern California
SECT14	Central Pacific States
SECT15	Northwestern States
SECT16	Southeastern Alaska
SECT17	Central Alaska
SECT18	Northern Alaska
SECT19	Southwestern Alaska
SECT20	Aleutian Islands
SECT21	Hawaiian Islands

Each section has one or more data files associated with it. The file naming convention used is as follows:

S00_XX.YYY

Each file starts with the letter "S", followed by the section number (01 to 21) and an underscore (_). The XX stands for the feature code (sometimes referred to as Overlay or Base Category). Feature codes are:

PB = Political Boundaries CF = Cultural Features

AB = Administrative Boundaries ST = Streams and Rivers

RD = Roads and Trails WB = Water Bodies

RR = Railroads HP = Hypsography (Continental Divide Only)

The YYY indicates the data format:

LGS = Line Graph Standard Format

LGO = Line Graph Optional Format

GRF = Graphic Format

If a section has more than one file for a feature, an underscore (_) and a letter are used to uniquely name the files. Examples:

S01_WB_A.LGO

S01_WB_B.LGO

LGO File

An .LGO file contains 15 records of general "header" information, followed by a series of Node, Area and Line entries, in that order. Each Node is assigned a unique ID number (1,2,...). A Node entry contains the node's ID, its coordinate (all coordinates in "optional" format files are in UTM or Albers Equal Area Ellipsoid projection) and the IDs of each line segment that begins at, ends at, or passes through the node. A "free-standing" node is one that has no line segments associated with it (i.e., the node is an isolated point).

Area ID Number

Each Area is assigned a unique ID number (1,2,...) called an area ID. An Area entry contains the area's ID, the coordinate of its "reference point", a list of the IDs of each line segment that forms the area's boundary (including islands and lakes) and a list of attribute codes assigned to the area. An area's reference point is that point on a map where a textual identifier for the area was placed (such as the name of a county).

Line ID Number

Each Line is also assigned a unique ID number (1,2,...). A Line entry contains the line's ID, the Node ID of the node it starts at, the Node ID of the node it ends at, the Area ID of the area to the left of the line, the Area ID of the area to the right of the line, a list of coordinates of the line vertices and a list of attribute codes assigned to the line segment. (Left and right are relative to the line's direction. The line was digitized from the start point to the end point) . A "free-standing" line is one that is not part of an area boundary.

Attribute Codes

Attribute codes are assigned to areas and line segments for the purpose of identifying and/or grouping them. An attribute code consists of two positive integers, a Major code value and a Minor code value. For example, USGS Section 15 contains data for Washington, Oregon, Idaho and part of Montana. In the Political Boundaries file (S15_PB.LGO), each county boundary area contains two attribute codes. One indicates which state the county was associated with (Major code = 91, Minor code = two-digit FIPS code for the state) and the other indicates which county it is (Major code = 92, Minor code = three-digit FIPS code for the county). For a list of attribute codes, see Major and Minor Attributes.

File Name Extensions

.DLG

Import Method

Choose the **File | Import** command.

Import Options Dialog

Specify the import options in the **USGS Digital Line Graph Boundary Import Options** dialog.

USGS ETOPO5 .DOS, .DAT File Description

The USGS ETOPO5 filter imports United States Geological Survey (USGS) ETOPO5 files.

File Description

ETOPO5 is a product of a digital data base of land and sea- floor elevations on a 5-minute latitude/longitude grid.

The ETOPO5 data files are commonly available in two different formats.

- An ETOPO5 file with the .DAT filename extension is typically stored in native or "big endian" byte order, as used by Macintosh, Sun, and some other workstations.
- An ETOPO5 file with the .DOS filename extension is typically stored in converted IBM-PC/DEC-VAX (or "little endian") byte order.

Data Formats

The data file is formatted as 16-bit BINARY INTEGERS in two byte orders: ETOPO5.DOS is in IBM-PC/DEC-VAX "swapped," lo-byte-first order. The file ETOPO5.DAT is in "normal," or hi-byte-first order, as used by Macintosh, Sun, and some other workstations. In both files, there are 2160x4320 data values, one for each five minutes of latitude and longitude, for a total of 9,331,200 points or 18,662,400 bytes.

Resolution and Accuracy

The resolution of the gridded data varies from true 5-minute for the ocean floors, the USA., Europe, Japan,and Australia to 1 degree in data-deficient parts of Asia, South America, northern Canada, and Africa.

Data values are in whole meters, representing the elevation of the CENTER of each cell.

Accuracy of the data set is hard to define, due to the disparate sources of the data. In general, the data sets for the USA, Western Europe, Korea/Japan, Australia and New Zealand are the most precise, having a horizontal resolution of five minutes of latitude and longitude, and vertical resolution of 1 meter. Data for Africa, Asia, and South America vary in resolution from +/- a few meters to only representing every 150m (500 fet), depending on the available source data. Very little detail is contained in the oceanic data shallower than 200m; the interpolation algorithm used by the US Navy to create the oceanic grid from contour charts was set to an arbitrary cutoff of -10m wherever the algorithm would have "overshot" and marked points as above sea level. An example of such an area is off Argentina, near 45S, 60W. All oceanic data are coded at least -1 m; land data are at 0 or greater, except where lake bottoms or other landlocked features go below sea level (Dead Sea, Death Valley, and in central Australia).

Data Order in the Files

The file may be thought of as having a logical record size of 8640 bytes. The data start at the North Pole (90 deg N, 0 deg 0' E) and are arranged in bands of 360 degrees x 12 points/degree = 4320 values (8640 bytes) ranging eastward from 0 deg 0' East longitude to 359 deg 55' East longitude (since it represents the North Pole, all possible longitudes still refer to a single point, thus the first band has 4320 identical values of -4290 m). The 8641st starts the latitude band for 89 deg 55' N, and so on. There is NO record for the South Pole (elevation 2810 m.)

Data sources are as follows: Ocean Areas: US Naval Oceanographic Office; USA., W. Europe, Japan/Korea: US Defense Mapping Agency; Australia: Bureau of Mineral Resources, Australia; New Zealand: Department of Industrial and Scientific Research, New Zealand; balance of world land masses: US Navy Fleet Numerical Oceanographic Center. These various data bases were originally assembled in 1988 into the worldwide 5-minute grid by Margo Edwards, then at Washington University, St. Louis, MO.

File Name Extensions

.DAT, .DOS

Import Method

Choose the **File | Import** command.

Import Options

Specify import options in the **Lattice Import Options** dialog.

DTED .DT0, .DT1, .DT2 Files

The DTED filter imports uniform lattices from several two-dimensional elevation grid file formats, also known as digital terrain elevation data or DTED data sets.

File Description

Digital Terrain Elevation Data .DTED are grids originally produced by the United States National Imaging and Mapping Agency (NIMA). The files contain latitude/longitude coordinates with the Z units in meters. The .DTED format is a standard of digital datasets which consists of a matrix of terrain elevation values. The standard was originally developed to support aircraft radar simulation and predication.

DTED supports many applications, including line-of-sight analyses, terrain profiling, 3D terrain visualization, mission planning/rehearsal, and modeling and simulation. DTED is a standard National Geospatial-Intelligence Agency (NGA) product that provides medium resolution, quantitative data in a digital format for military system applications that require terrain elevation.

File Format

There are multiple "levels" of DTED files. The DTED format for level 0, 1 and 2 is described in *U.S. Military Specification Digital Terrain Elevation Data (DTED) MIL-PRF-89020B*, and amongst other parameters describes the resolution for each level. DTED data is stored in a big endian format where negative numbers are signed magnitude.

Level 0 files .DT0 are 1-degree square blocks with 30 arc-second spacing in latitude direction, similar to GTOPO 30 files.

Level 1 files .DT1 use three arc-second spacing. Level 1 has a post spacing of ca. 90 meters which results in the following table:

Zone	Latitude	Interval Latitude (arc sec)	Interval Longitude (arc sec)
I	0-50 degrees (North-South)	3	3
II	50-70 degrees (North-South)	3	6
III	70-75 degrees (North-South)	3	9
IV	75-80 degrees (North-South)	3	12
V	80-90 degrees (North-South)	3	18

Level 2 files .DT2 use one arc-second spacing.

In addition three more levels (3, 4 and 5) at increasing resolution have been proposed, but not yet standardized.

File Name Extensions

DTED .DT0, .DT1, .DT2

Format(s) Supported for Import

2D uniform lattice; 8-, 16-, 32-bit integer, float, double

Import Method

Choose the **File | Import** command.

AutoCAD DXF File Description

Voxler imports and exports 2D and 3D DXF files.

File Description

AutoCAD DXF files are ASCII files (i.e., they can be edited with a text editor or word processor) or binary files (cannot be edited) containing records indicating graphical entities and their attributes.

They provide a medium of exchange with AutoDesk's AutoCAD program. DXF files are commonly used for storage and exchange of CAD and vector information.

File Format

The format of DXF files is complex and a detailed discussion is beyond the scope of a help file. Many books describing the DXF file format are widely available.

Graphics

Graphical information may be stored in the AutoCAD Drawing Exchange Format (DXF). Many programs, including AutoDesk Inc.'s AutoCAD (Computer Aided Design) program can import DXF files, allowing one to display and/or further manipulate the images. MTEXT (multi-line text block) background color in DXF import filter is supported.

3DSOLID is a proprietary format, and 3DSOLID entities are not supported in **Voxler**. If the .DXF file contains 3DSOLID objects, the 3DSOLID objects will be omitted during the import process.

ASCII or Binary Format

DXF files can be stored in either ASCII or Binary format. ASCII DXF files are the most versatile, since they can be displayed, edited, printed and transported to non-IBM machines (such as mainframes, minicomputers or Macintosh). However, they are somewhat bigger and take longer to read back into another application.

Table and Entities Sections

DXF files have two important sections.

- The *Tables* section contains definitions of the various line styles and other attributes.
- The *Entities* section contains specific information about each graphical entity (line, polygon, etc.) including coordinates and references to the attributes in the defined in the Tables section. All exported graphical entities are assigned to a layer named GSLAYER.

Filled Graphics

The DXF file format doesn't allow graphical entities (such as rectangles, polygons, etc.) to be filled. Therefore, only the edges of objects are exported. Their interior is always transparent. DXF is a difficult format, mainly because it can contain so many different types of data. The format is controlled and defined by Autodesk for use in its CAD program AutoCAD. The most common form of DXF is 7-bit text, but there are also two related binary formats, one that also uses the DXF extension and another that uses the DXB extension.

Line Styles

Lines styles are exported with equivalent AutoCAD-compatible line types. The document's internal line styles are assigned the following AutoCAD line type names:

Document	AutoCAD
Solid	CONTINUOUS
Internal Dash	GSDASHED
Internal Dot	GSDOTTED
Internal Dash-Dot	GSDASHDOT

Internal Dash-Dot-Dot	GSDASHDOTDOT

Custom line styles in the document are assigned AutoCAD line type names of the form GSCUSTOM0, GSCUSTOM1, GSCUSTOM2, etc.

Color Numbers

Indexed DXF color numbers are assigned to each entity. Color numbers (1,2,3,…,255) are indices into AutoCAD's internal color table. By convention, the first 7 color numbers are guaranteed to have known colors assigned to them by AutoCAD. They are:

1	Red
2	Yellow
3	Green
4	Cyan
5	Magenta
7	Black (Default for GSLAYER)

AutoCAD has a default association of colors to color numbers, but the AutoCAD user is free to change the colors associated with color numbers 8 through 255. When exporting to DXF format, the color number of the color from the default AutoCAD color table closest to the actual color of the object is assigned to the entity in the DXF file. Unless you use only the seven colors listed above, the color of objects inside AutoCAD may be different than those in the application document.

AutoCAD 2004 and later versions support true colors. When user chooses *AutoCAD 2004 or later* in the **Export Options** dialog, true colors are written to the export file.

File Name Extensions

.DXF

Import Method

Choose the **File | Import** command.

Import Options Dialog

Specify import options in the **AutoCAD DXF Import Options** dialog.

Import Restrictions/Limitations

The fill and text in a .DXF will not appear when loaded into Voxler. Varying line widths are imported into Voxler and scaled from 0 to 4 points where *size (points) = width (inches) / 72*. However line widths greater than or equal to 0.05555556 inches will be rendered in Voxler at the same width (i.e. 4 points).

3DSOLID is a proprietary format, and 3DSOLID entities are not supported in **Voxler**. If the .DXF file contains 3DSOLID objects, the 3DSOLID objects will be omitted during the import process.

Export Method

Select a module and click the **File | Save Data** command.

Export Options Dialog

Specify export options in the **AutoCAD DXF Export Options** dialog.

Esri Format .E00 Grid File Description

Esri Format .E00 grid files are ASCII files containing elevation data. They provide a medium of exchange between Esri application programs on different hardware and operating system platforms (Windows, UNIX, etc.).

Import Options

Select lattice options in the **Lattice Import Options** dialog. No E00 specific import options dialog is displayed.

Import Automation Options

No import options are available.

Export

Voxler does not currently export .E00 grid files.

Disclaimer

The E00 file format is not publicly documented by Esri. Although Golden Software has tested this E00 import filter software with a number of publicly available E00 grid files, it may not be compatible with all E00 files created by all versions of Esri application programs. Golden Software is not affiliated with Esri, and this import filter software is not a product of, nor endorsed by, Esri.

Esri ArcInfo Export Format .E00 File Description

The Esri ArcInfo Export Format filter imports .E00 files.

File Description

Esri ArcInfo Export Format .E00 files are ASCII files containing topological entities and their attributes. They provide a medium of exchange between Esri application programs on different hardware and operating system platforms (Windows, UNIX, etc.). E00 is essentially an interchange data format developed to enable users to move data into and out of ARC/INFO.

File Format

Data in .E00 is an ARC/INFO interchange file format. this is a simple ASCII file which can be opened and viewed using any common text editor (wordpad, notepad, textpad).

Example

```
EXP  0
/GISWK/FTP/PUB/TMP20/CNTYS.E00                              ARC  2       1     18114       2
     1       2       1       2 -1.9764960E+06 3.1712590E+06-1.9802236E+06
3.1723502E+06       2    18115       1       2       2       1       4 -1.9802236E+06
3.1723502E+06-1.9810328E+06 3.1690918E+06 -1.9762238E+06 3.1684802E+06-
1.9764960E+06 3.1712590E+06
```

File Name Extensions

.E00

Import Method

Choose the **File | Import** command.

Import Options

Specify import options in the **Esri ArcInfo Export Format .E00 Import Options** dialog.

Disclaimer

The E00 file format is not publicly documented by Esri. Although Golden Software has tested this E00 import filter software with a number of publicly available E00 files, it may not be compatible with all E00 files created by all versions of Esri application programs. Golden Software is not affiliated with Esri, and this import filter software is not a product of, nor endorsed by, Esri.

ER Mapper .ECW File Description

Voxler imports ERMapper .ECW files.

Enhanced Compression Wavelet .ECW is an open standard wavelet compression image format developed by Earth Resource Mapping. The file format is optimized for aerial and satellite imagery, and efficiently compresses very large images with fine, alternating contrast. This is a lossy compression format. A region can be specified for import in the **.ECW Image Import Options** dialog, or the image can be imported as a read-only image. Each of these options decreases the import time for large ECW files.

The .ECW file format has the following properties:

- Embeds map projection information
- Fast compression (about 1.5 MB of compressed file per second on 1 GHz processor)
- Typical compression ratios between 1:10 and 1:100
- Possible decompression of selected regions without the need to decompress the whole file
- Data flow compression allows for compression of big files with small RAM requirements

Import Options

See **.ECW Image Import Options Dialog**.

Import Automation Options

See **ER Mapper .ECW Import Automation Options**.

Export Options

Voxler does not currently support .ECW export.

Encapsulated PostScript .EPS File Description

The Encapsulated PostScript filter exports Encapsulated PostScript files .EPS files.

File Description

An encapsulated postscript file is more-or-less a self-contained, reasonably predictable PostScript document that describe an image or drawing, that can be placed within another PostScript document. EPS files also frequently include a preview picture of the content, for on-screen display. The idea is to allow a simple preview of the final output in any application that can draw a bitmap. Without this preview the applications would have to directly render the PostScript (PS) data inside the EPS, which was beyond the capabilities of most machines until recently. EPS files are commonly used for illustration and DTP applications, bitmap and vector data interchange.

File Format

At a minimum, an EPS file contains a *BoundingBox DSC comment*, describing the rectangle containing the image described by the EPS file. Applications can use this information to lay out the page, even if they are unable to directly render the PostScript inside.

Identifying EPS files

Because of the different ways in which EPS previews are handled, there is no one way to identify an EPS file.

- A Windows-format EPS file containing a TIFF or WMF preview must start with the four bytes containing, in hexadecimal, C5 D0 D3 C6. Bear in mind these files are widespread on all platforms.
- In all other cases an EPS file must start with a line %!PS-Adobe-a.b EPSF-c.d where a, b, c and d are all single digit numbers.
- A Mac-format EPS file is accompanied by a resource fork. The preview is a PICT resource with ID 256. An EPS file on the Mac is expected to have a file type code of "EPSF", whether or not it has a preview.
- An EPSI file will contain a line starting %%BeginPreview: in the DSC prolog.
- In many cases no preview is present at all.

File Name Extensions

.EPS

Export Method

Choose the **File | Export** command or select an image module and choose the **File | Save Data** command.

Export Options

Specify the export options in the **Export Options** dialog and **Encapsulated PostScript .EPS Export Options** dialog. The **Size and Color** page is not available when the **File | Save Data** command is used to export the file.

ER Mapper Grid .ERS File Description

The ER Mapper Grid filter imports and exports ER Mapper grid format .ERS files.

File Description

The ER Mapper grid format consists of two files: a binary data file (no file suffix) and an ASCII header file .ERS.

The ER mapper grid file does not have a file extension. It should have the same format as a .BIL file, and it must have an associated ASCII header file with the same filename but with .ERS file extension. This header file contains data about the image such as the number of rows and columns in the image and latitude and longitude.

These files can be read directly by the ER Mapper software or by other packages such as ERDAS IMAGINE

File Name Extensions

.ERS

Import Method

Choose the **File | Import** command.

Import Options

Specify import options in the **Lattice Import Options** dialog.

Export Method

Select a lattice or image module and choose the **File | Save Data** command.

AVS Field .FLD Files

The AVS filter imports and exports several different types of data from the AVS Field data sets.

File Description

The AVS Field data set file format originated with the AVS5 and AVS/Express applications.

File Description

The data set consists of a file with the filename extension .FLD, which describes the data set; plus possibly one or more additional data files referenced if the data is not contained within the .FLD file itself.

File Name Extensions

.FLD, which describes the data set; plus possibly one or more additional data files referenced if the data is not contained within the .FLD

Format(s) Supported for Import

- 3D uniform lattice; 8-, 16-, 32-bit integer, float, double

- 3D rectilinear lattice; 8-, 16-, 32-bit integer, float, double
- 3D curvilinear lattice; 8-, 16-, 32-bit integer, float, double

Format(s) Supported for Export

- 3D uniform lattice; 8-, 16-, 32-bit integer, float, double
- 3D rectilinear lattice; 8-, 16-, 32-bit integer, float, double
- 3D curvilinear lattice; 8-, 16-, 32-bit integer, float, double

Import Methods

Choose the **File | Import** command.

Export Methods

Select a data, lattice, or image module and choose the **File | Save Data** command.

Export Options

Specify export options in the Export Blanking Options dialog or the **AVS .FLD Export Options** dialog.

Import Restrictions/Limitations

During import, any external data files referenced from within the .FLD file are assumed to reside in the same directory where the .FLD file resides.

Export Restrictions/Limitations

Exported AVS data sets always place the data values inside the .FLD, not in external data files.

Esri Binary Float Grid .FLT, .HDR File Description

The Esri Binary Float Grid filter imports and exports Esri floating point grid files in Esri Spatial Analyst .FLT, .HDR format.

File Description

A grid is a raster data storage format native to Esri. There are two types of grids: integer and floating point. Use integer grids to represent discrete data and floating-point grids to represent continuous data.

Attributes for an integer grid are stored in a value attribute table (VAT). A VAT has one record for each unique value in the grid. The record stores the unique value (VALUE is an integer that represents a particular class or grouping of cells) and the number of cells (COUNT) in the grid represented by that value. For example, if 50 cells have a value of 1 representing a forest, then the VAT would show a VALUE = 1 and COUNT = 50 for each of the 50 cells. Floating-point grids do not have a VAT because the cells in the grid can assume any value within a given range of values. The cells in this type of grid do not fall neatly into discrete categories. The cell value itself is the attribute that describes the location. For example, in a grid that represents elevation data in meters above sea level, a cell with a value of 10.1662 indicates that the location is about 10 meters above sea level.

The HDR file is a binary file. Information stored in the file includes the cell size, type of grid (integer or floating point), compression technique, blocking factor, and tile information.

The coordinate system of a grid is the same as that of other geographic data. The rows and columns are parallel to the x- and y-axes of the coordinate system. Since each cell within a grid has the same dimension as other cells, the location and area covered by any cell is easily determined by its row and column. The coordinate system of a grid is thus defined by the cell size, the number of rows and columns, and the x,y coordinate of the upper left corner. Grids also carry additional information, such as the coordinate system associated with the grid.

File Format

The Esri Binary Float Grid format consists of 2 files: a binary .FLT image file and ASCII .HDR header file with the same filename but different file extension. For example, Test.FLT and Test.HDR.

Grids are implemented using a tiled raster data structure in which the basic unit of data storage is a rectangular block of cells. Blocks are stored on disk in compressed form in a variable-length file structure referred to as a tile. Each block is stored as one variable-length record.

The size of the tile for a grid is based on the number of rows and columns in the grid at the time of creation. The upper limit on the size of a tile is set by the application and is very large (currently set at 4,000,000 x 4,000,000 cells). As a result, most grids used for GIS applications are automatically stored in a single tile. The spatial data for a grid is automatically split across multiple tiles if the size of the grid at the time of creation is larger than the upper limit for the size of a tile.

File Name Extensions

.FLT, .HDR

Import Method

Choose the **File | Import** command.

Import Options

Specify import options in the **Lattice Import Options** dialog.

Export Method

Select a lattice or image module and choose the **File | Save Data** command.

Graphics Interchange Format .GIF File Description

The GIF filter imports and exports bitmap images from CompuServe Graphics Interchange Files .GIF.

File Description

The Graphics Interchange Format .GIF is an image format that was introduced by CompuServe in 1987 and has since come into widespread usage on the World Wide Web due to its wide support and portability. Graphics interchange files are suitable for sharp-edged line art (such as logos) with a limited number of colors. This takes advantage of the format's lossless compression, which favors flat areas of uniform color with well defined edges (in contrast to JPEG, which favors smooth gradients and softer images). The 256 image palette color limitation makes the GIF format

unsuitable for reproducing color photographs and other images with continuous color, but it is well-suited for simpler images such as graphics or logos with solid areas of color. The GIF format was originally designed to facilitate image transfer and online storage for use by CompuServe and its customers, GIF is primarily an exchange and storage format. GIF is a well-defined, well-documented format in wide use, which is quick, easy to read, and reasonably easy to uncompress. It lacks, however, support for the storage of deep-pixel images.

GIF images are compressed using the Lempel-Ziv-Welch (LZW) lossless data compression technique to reduce the file size without degrading the visual quality. This compression technique was patented in 1985. Controversy over the licensing agreement between the patent holder, Unisys, and CompuServe in 1994 inspired the development of the Portable Network Graphics (PNG) standard; since then all the relevant patents have expired.

Usage

- Sharp-edged line art (such as logos) with a limited number of colors. This takes advantage of the format's lossless compression, which favors flat areas of uniform color with well defined edges (in contrast to JPEG, which favors smooth gradients and softer images).
- Used to store low-color sprite data for games.
- Used for small animations and low-resolution film clips.
- In view of the general limitation on the GIF image palette to 256 colors, it is not usually used as a format for digital photography. Digital photographers use image file formats capable of reproducing a greater range of colors, such as TIFF, RAW or the lossy JPEG, which is more suitable for compressing photographs.
- The PNG format is a popular alternative to GIF images since it uses better compression techniques and does not have a limit of 256 colors, but PNGs do not support animations.

File Format

The format supports up to 8 bits per pixel, allowing a single image to reference a palette of up to 256 distinct colors chosen from the 24-bit RGB color space. It also supports animations and allows a separate palette of 256 colors for each frame. The color limitation makes the GIF format unsuitable for reproducing color photographs and other images with continuous color, but it is well-suited for simpler images such as graphics or logos with solid areas of color.

GIF is different from many other common bitmap formats in the sense that it is stream-based. It consists of a series of data packets, called *blocks*, along with additional protocol information. Because of this arrangement, GIF files must be read as if they are a continuous stream of data. The various blocks and sub-blocks of data defined by GIF may be found almost anywhere within the file. This uncertainty makes it difficult to encapsulate every possible arrangement of GIF data in the form of C structures.

There are a number of different data block categories, and each of the various defined blocks falls into one of these categories. In GIF terminology, a Graphics Control Extension block is a type of Graphics Control block, for instance. In like manner, Plain Text Extension blocks and the Local Image Descriptor are types of Graphic Rendering blocks. The bitmap data is an Image Data block. Comment Extension and Application Extension blocks are types of Special Purpose blocks.

Blocks, in addition to storing fields of information, can also contain sub-blocks. Each data sub-block begins with a single count byte, which can be in the range of 1 to 255 and indicates the number of data bytes that follow the count byte. Multiple sub-blocks may occur in a contiguous grouping (count byte, data bytes, count byte, data bytes, and so on). A sequence of one or more data sub-blocks is terminated by a count byte with a value of zero.

The GIF format is capable of storing bitmap data with pixel depths of 1 to 8 bits. Images are always stored using the RGB color model and palette data. GIF is also capable of storing multiple images per file, but this capability is rarely utilized, and the vast majority of GIF files contain only a single image. Most GIF file viewers do not, in fact, support the display of multiple image GIF files or may display only the first image stored in the file. For these reasons, we recommend not creating applications that rely on multiple images per file, even though the specification allows this.

File Name Extensions

.GIF

Import Method

Choose the **File | Import** command.

Export Method

Choose the **File | Export** command or select an image module and choose the **File | Save Data** command.

Export Options

See *Export Options - Size and Color* page.

See *GIF Export Options* dialog.

Export Restrictions/Limitations

The .GIF file format is limited to paletted images with 8 or fewer bits per pixel. If the export source contains more than 8 bits per pixel, it is quantized down to 8 bits per pixel during the export procedure. The .GIF file format is set to 72 *Pixels per inch* for all .GIF images and cannot be changed. GIF images are always 72 DPI by definition. For higher quality images, it is suggested that PNG, TIFF, or BMP is used instead of GIF.

GRIB Weather Data Grid File Description

Voxler imports and GRB weather data grid files.

A GRIB grid is a binary file format developed for use in the weather and meteorological industries to transport and manipulate weather data and is the foundation of weather forecasts.

File Format

The file format is maintained by the World Meteorological Organization. The format is described in detail at http://www.wmo.int/pages/prog/www/WDM/Guides/Guide-binary-2.html.

File name extensions

.GRB

Import Options Dialog

See *GRIB Import Options*

Import Automation Options

See *GRIB Import Automation Options*.

Loading Files

Use the **File | Import** command to load a GRIB Weather Data Grid file.

Export Options

Voxler does not currently export .GRB files.

ASCII Grid File Format

File Description

ASCII grid files .GRD contain five header lines that provide information about the size and limits of the grid, followed by a list of Z values. The fields within ASCII grid files must be space delimited.

The listing of Z values follows the header information in the file. The Z values are stored in row-major order starting with the minimum Y coordinate. The first Z value in the grid file corresponds to the lower left corner of the map. This can also be thought of as the southwest corner of the map or, more specifically, the grid node of minimum X and minimum Y. The second Z value is the next adjacent grid node in the same row (the same Y coordinate but the next higher X coordinate). When the maximum X value is reached in the row, the list of Z values continues with the next higher row until all of the rows of Z values have been included.

The general format of an ASCII grid file is:

id	The identification string DSAA that identifies the file as an ASCII grid file.
nx ny	nx is the integer number of grid lines along the X axis (columns) ny is the integer number of grid lines along the Y axis (rows)
xlo xhi	xlo is the minimum X value of the grid xhi is the maximum X value of the grid
ylo yhi	ylo is the minimum Y value of the grid yhi is the maximum Y value of the grid
zlo zhi	zlo is the minimum Z value of the grid zhi is the maximum Z value of the grid
grid row 1 grid row 2 grid row 3 ...	These are the rows of Z values of the grid, organized in row order. Each row has a constant Y coordinate. Grid row 1 corresponds to ylo; the last grid row corresponds to yhi. Within each row, the Z values are arranged from xlo to xhi.

Example

The following example grid file is ten rows high by ten columns wide. The first five lines of the file contain header information. X ranges from 0 to 9; Y ranges from 0 to 7; and Z ranges from 25 to 97.19. The first Z value shown corresponds to the lower left corner of the map and the following values correspond to the increasing X positions along the bottom row of the grid file. This file has a total of 100 Z values.

```
DSAA
10 10
0.0 9.0
0.0 7.0
25.00 97.19
91.03 77.21 60.55 46.67 52.73 64.05 41.19 54.99 44.30 25.00
96.04 81.10 62.38 48.74 57.50 63.27 48.67 60.81 51.78 33.63
92.10 85.05 65.09 53.01 64.44 65.64 52.53 66.54 59.29 41.33
94.04 85.63 65.56 55.32 73.18 70.88 55.35 76.27 67.20 45.78
97.19 82.00 64.21 61.97 82.99 80.34 58.55 86.28 75.02 48.75
91.36 78.73 64.05 65.60 82.58 81.37 61.16 89.09 81.36 54.87
86.31 77.58 67.71 68.50 73.37 74.84 65.35 95.55 85.92 55.76
80.88 75.56 74.35 72.47 66.93 75.49 86.39 92.10 84.41 55.00
74.77 66.02 70.29 75.16 60.56 65.56 85.07 89.81 74.53 51.69
70.00 54.19 62.27 74.51 55.95 55.42 71.21 74.63 63.14 44.99
```

Geosoft Binary Grid .GRD File Description

The Geosoft Grid filter imports Geosoft .GRD or .GGF grid version 2 format in compressed and uncompressed format and exports in .GRD or .GGF format.

File Description

Geosoft grid is a binary format for storing raster data typically used for geophysical and elevation data. Two files describe each grid (.GRD and .GI). Technical information on this file format is reportedly available in the manual for Geosoft's Montaj software application.

File Format

Sometimes there is an associated .GI projection file that comes with the binary .GRD file.

File Name Extensions

.GI, .GRD, .GGF

The .GI file contains information about the coordinate reference system.

Import Method

Choose the **File | Import** command.

Export Method

Select a lattice or image module and choose the **File | Save Data** command to export .GRD or .GGF files.

Surfer Grid File .GRD Files

The Golden Software Grid filter imports and exports two-dimensional uniform lattices (elevation grids) in the **Surfer** Grid file format. Both ASCII and binary variants are supported.

File Description

The Golden Software Grid filter imports and exports two-dimensional uniform lattices (elevation grids) in the **Surfer** Grid file format. Both ASCII and binary variants are supported

Files with the .GRD extension are **Surfer** grid files. To preserve faulting information and to use double precision values, be sure to save the grid in the **Surfer 7** format. If you need to use a grid in **Surfer 5** or **Surfer 6**, save the grid as GS ASCII .GRD or GS Binary .GRD. Keep in mind that these two formats do not retain fault information or double precision values.

File Name Extensions

.GRD

Format(s) Supported for Import

2D uniform lattice; float, double

Format(s) Supported for Export

2D uniform lattice; float, double

Import Method

Choose the **File | Import** command.

Import Options

Specify import options in the **Lattice Import Options** dialog.

Export Method

Select a lattice or image module and choose the **File | Save Data** command.

Select one of three encodings for the export file:

- ASCII Grid
- Surfer 6 Binary Grid
- Surfer 7 Binary Grid

Export Restrictions/Limitations

Choose the **File | Save Data** command to export a uniform lattice to the **Surfer** Grid format. The exported lattice has a "z" dimension of one, i.e., the lattice is two-dimensional. If an input lattice with a z dimension of greater than one is exported, then **Voxler** prompts the user to save multiple slices into separate .GRD files.

If a multi-slice uniform lattice is output to a .GRD file format, it is output as a series of slices in the Z direction. The **Select Slices** dialog appears and the user is prompted for the range of slices to output and a file name template that adds the slice number to each generated file name.

Surfer 6 Grid File Format

Surfer 6 grid files .GRD use a layout similar to the ASCII grid format. The only difference is in the identification string and that **Surfer 6** grid files are binary. Data types used in **Surfer 6** grid files include:

Type	Description
char	single byte
short	16-bit signed integer
float	32-bit single precision floating point value
double	64-bit double precision floating point value

The **Surfer 6** format has the following layout:

Element	Type	Description
id	char	4-byte identification string 'DSBB' which identifies the file as a **Surfer 6** binary grid file
nx	short	number of grid lines along the X axis (columns)
ny	short	number of grid lines along the Y axis (rows)
xlo	double	minimum X value of the grid
xhi	double	maximum X value of the grid
ylo	double	minimum Y value of the grid
zlo	double	minimum Z value of the grid
zhi	double	maximum Z value of the grid
z11, z12, ...	float	first row of the grid. Each row has a constant Y coordinate. The first row corresponds to ylo; the last row corresponds to yhi.
z21, z22, ...	float	second row of the grid
z31, z32, ...	float	third row of the grid
...	float	all other rows of the grid up to yhi

Surfer 7 Grid File Format

Surfer 7 grid files .GRD use a tag-based binary format to allow for future enhancements. Each section is preceded by a tag structure which indicates the type and size of the following data. If a program does not understand or want a particular type of data, it can read the associated tag and quickly skip to the next section. In general, sections can appear in any order except for the first, which must be a Header section.

Data types used in **Surfer 7** grid files:

Type	Description
long	32-bit signed integer
double	64-bit double precision floating point value

Each section is preceded by a tag structure with the following format:

Element	Type	Description
Id	long	The type of data in the following section. See the next table for a list of valid values.
Size	long	The number of bytes in the section (not including this tag). Skipping this many bytes after reading the tag aligns the file pointer on the next tag.

Tag Id values. The 0x prefix indicates a hexadecimal value:

Id	Description
0x42525344	*Header* section - must be the first section within the file.
0x44495247	*Grid* section - describes a 2D matrix of Z values.
0x41544144	*Data* section - contains a variable amount of data. The size of the data section is given by the *Size* field in the tag structure.
0x49544c46	*Fault Into* section - describes the fault traces used when creating the grid.

The *Grid* section consists of a header that describes a 2D matrix of values followed by the matrix itself. This section encapsulates all of the data that was traditionally referred to as a grid.

Element	Type	Description
nRow	long	number of rows in the grid
nCol	long	number of columns in the grid
xLL	double	X coordinate of the lower left corner of the grid
yLL	double	Y coordinate of the lower left corner of the grid
xSize	double	spacing between adjacent nodes in the X direction (between columns)
ySize	double	spacing between adjacent nodes in the Y direction (between rows)

zMin	double	minimum Z value within the grid
zMax	double	maximum Z value within the grid
Rotation	double	not currently used
BlankValue	double	nodes are blanked if greater or equal to this value

A *Data* section containing the 2D matrix of values (doubles) must immediately follow a *Grid* section. The grid is stored within the *Data* section in row-major order, with the lowest row (minimum Y) first.

A *Fault Info* section describes the fault geometry used to create the grid. *Fault Info* sections have the following format:

Element	Type	Description
nTraces	long	number of fault traces (polylines)
nVertices	long	total number of vertices in all the traces
data section		variable-sized data block consisting of an array of *Traces* immediately followed by an array of *Vertices*

A *Data* section containing an array of *Trace* structures and an array of *Vertex* structures must immediately follow a *Fault Info* section. The number of *Trace* structures in the array is nTraces, and the number of *Vertex* structures is nVertices.

Trace structure:

Element	Type	Description
iFirst	long	0-based index into the vertex array for the first vertex of this trace
nPts	long	number of vertices in this trace

Vertex structure:

Element	Type	Description
x	double	X coordinate of the vertex
y	double	Y coordinate of the vertex

Example

The following example illustrates the layout for a 5-row by 10-column grid:

Element	Type	Description
0x42525344	long	Tag: Id for Header section
4	long	Tag: Size of Header section
1	long	Header Section: Version
0x44495247	long	Tag: ID indicating a grid section
72	long	Tag: Length in bytes of the grid section
5	long	Grid Section: nRow
10	long	Grid Section: nCol

0.0	double	Grid Section: xLL
0.0	double	Grid Section: yLL
1.0	double	Grid Section: xSize
1.75	double	Grid Section: ySize
25.0	double	Grid Section: zMin
101.6	double	Grid Section: zMax
0.0	double	Grid Section: Rotation
1.70141e38	double	Grid Section: BlankValue
0x41544144	long	Tag: ID indicating a data section
400	long	Tag: Length in bytes of the data section (5 rows x 10 columns x 8 bytes per double)
Z11, Z12, ...	double	Data Section: First (lowest) row of matrix, 10 doubles
Z21, Z22, ...	double	Data Section: Second row of matrix, 10 doubles
Z31, Z32, ...	double	Data Section: Third row of matrix, 10 doubles
Z41, Z42, ...	double	Data Section: Fourth row of matrix, 10 doubles
Z51, Z52, ...	double	Data Section: Fifth row of matrix, 10 doubles

Golden Software Boundary .GSB File Description

The Golden Software Boundary filter imports .GSB files.

The .GSB format is a proprietary Golden Software file format. There are several different versions of GSB files, so older Golden Software applications may not be able to read .GSB files exported from newer applications.

File Description

Golden Software Boundary files contain boundary objects including areas, curves and points. Primary and Secondary IDs are usually associated with each object. The objects have no attributes (such as color or line style) associated with them.

File Format

Golden Software Boundary files are binary files (i.e., they can't be created or modified with a text editor or word processor) that are usually used as base maps. Information indicating the type of projection used (if any) is also stored in the file.

File Name Extensions

.GSB

Import Method

Choose the **File | Import** command.

Golden Software Interchange .GSI File Description

The Golden Software Interchange filter imports .GSI files.

File Description

Golden Software Interchange .GSI files are binary files containing records indicating graphical entities and their attributes. They provide a medium of exchange between Golden Software application programs.

File Name Extensions

.GSI

Import Methods

Choose the **File | Import** command.

Golden Software Reference Files

If you have looked in a folder that contains data files created by **Voxler**, **Surfer**, or another Golden Software application you may have noticed files with a .GSR2 extension in the folder. These Golden Software Reference files are created when you save a projected data file from **Voxler**. For example if you create a data file, use the **Data | Assign Coordinate System** command, and save the data file as a *Golden Software .DAT* file, you will see both the filename.dat and filename.dat.gsr2 in the directory.

The Golden Software Reference files contain the projection settings used to project the data in **Surfer**, **Grapher**, **MapViewer**, **Strater**, or **Didger**.. Projection, datum, and georeference information are stored in the .GSR2 file. When you create a map from data or a grid that has an associated .GSR2 file into **Surfer**, the projection information is used when displaying the map. If the Golden Software Reference file is deleted, the boundary can be imported but it will not be projected properly.

Grid Exchange .GXF File Description

The Grid Exchange filter imports and exports elevation data in Grid Exchange .GXF file format.

File Description

The Grid eXchange File .GXF is an open standard ASCII file format for exchanging gridded data among different software systems. Software that supports the .GXF standard will be able to import properly formatted .GXF files and export grids in .GXF format.

File Format

A GXF file is an ASCII file made up of a number of labeled data objects and comments. Each labeled data object has a label line followed by one or more data lines. A label line is identified by a '#' character in the first column followed immediately by an upper-case label. The data associated with that label are found on one or more lines that follow that label.

Any lines that are not part of a labeled data object are ignored and can be used to place comments with a .GXF file. Programs that read .GXF files ignore such comment lines while searching for the next GXF data object.

All lines in a .GXF file must be less than or equal to 80 characters in length. If the last non-white space character on a line is a '\', the next line is assumed to be a continuation of the current line (except for #GRID data lines). Spaces and tab characters are white space characters. Note that continuation lines are a feature of GXF revision 3, and they should only be required for defining projection parameters that require more than 80 characters. They are not required and cannot be used for writing grid data to the GXF (in the #GRID object). This allows GSF readers written for GXF revision 2 and earlier to read GXF revision 3 files.

Any name or string parameters that are part of a data object and which themselves contain spaces, must be enclosed in double quotes. Parameters on data lines may be separated by a space or comma.

For example, the following file satisfies this format:

This is a very small 4 x 5 point grid.

#POINTS
5
#ROWS
4

These lines are skipped because they are not part of a data object.

#GRID
135.28 122.21 119.64 163.25 199.15
145.38 132.45 120.32 121.42 205.18
140.13 151.48 132.91 119.12 219.67
132.67 150.56 140.45 102.89 218.41

This file contains two comment lines at the beginning, followed by two labeled data objects, another comment, and finally a #GRID data object. The #GRID object is always the last object and indicates the beginning of the gridded data. This example contains the minimum information required to define a grid, and other grid information, such as location, storage, sense, sampling separations, etc, assume default values. The next section documents all data objects defined in this revision. The only required data objects are #POINTS, #ROWS, and #GRID. All other GXF objects will assume default value. However, a recommended minimum set of defined data objects would also include #PTSEPARATION, #RWSEPARATION, #XORIGIN, and #YORIGIN, since these objects serve to define the minimum amount of information needed to locate the grid in an assumed coordinate system.

File Name Extensions
.GXF

Import Method

Choose the **File | Import** command.

Import Options

Specify import options in the **Lattice Import Options** dialog.

Export Method

Select a lattice or image module and choose the **File | Save Data** command.

Export Options

Specify export options in the **Grid Exchange .GXF Export Options** dialog.

References

SEG Gravity and Magnetic Committee, GXF Grid eXchange File, May 2, 2001.
http://www.geosoft.com/resources/technotes/pdfs/gxfr3d9_1.pdf

Hierarchical Data Format .HDF File Description

The Hierarchical Data Format filter imports and exports data sets in Hierarchical Data Format (HDF) Version 4.

File Description

Hierarchical Data Format, commonly abbreviated HDF, HDF4, or HDF5 is a library and multi-object file format for the transfer of graphical and numerical data between computers. It was created by the NCSA, but is currently maintained by The HDF Group. The freely available HDF distribution consists of the library, command-line utilities, test suite source, Java interface, and the Java-based HDF Viewer (HDFView).

HDF supports several different data models, including multidimensional arrays, raster images, and tables. Each defines a specific aggregate data type and provides an API for reading, writing, and organizing the data and metadata. New data models can be added by the HDF developers or users.

HDF is self-describing, allowing an application to interpret the structure and contents of a file without any outside information. One HDF file can hold a mixture of related objects which can be accessed as a group or as individual objects. Users can create their own grouping structures called "vgroups."

File Name Extensions

.HDF

Format(s) Supported for Import

- 3D uniform lattice; 8-, 16-, 32-bit integer, float, double
- 3D rectilinear lattice; 8-, 16-, 32-bit integer, float, double
- 3D curvilinear lattice; 8-, 16-, 32-bit integer, float, double
- point data

Format(s) Supported for Export

- 3D uniform lattice; 8-, 16-, 32-bit integer, float, double
- 3D rectilinear lattice; 8-, 16-, 32-bit integer, float, double
- 3D curvilinear lattice; 8-, 16-, 32-bit integer, float, double
- point data

Import Method

Choose the **File | Import** command.

Import Options

Specify import options in the **Lattice Import Options** dialog.

Export Method

Choose the **File | Save Data** command.

Export Options

Specify export options in the **Hierarchical Data Format .HDF Export Options** dialog ifthe lattice contains one or more blanked nodes. See Export Blanking Options.

Analyze 7.5 Medical Image .HDR, .IMG Files

The Analyze 7.5 Medical Image filter imports and exports Analyze 7.5 medical image data sets as uniform lattices.

File Description

Analyze 7.5 Medical Image .HDR, .IMG files allow medical images (such as MRI's) to be displayed. An ANALYZE image database consists of at least two files; an image file .IMG and a header file .HDR. The files have the same name being distinguished by the extensions.

Analyze 7.5 format can contain multiple volumes (i.e., time points), each volume with multiple slices (i.e., Z dimensions), each slice with X and Y dimensions and each pixel (i.e., node) with possible multiple components. Each volume should have the same X, Y and Z dimension and component info.

File Format

The format of the image file is simple and typically contains uncompressed pixel data for the images. The header file typically describes the dimensions and history of the pixel data. The header structure consists of three substructures; the header_key describes the header, the image_dimension describes the image sizes, and the data_history is optional.

The header format is flexible and can be extended for new user-defined data types. The essential structures of the header are the header_key and the image_dimension. The data_history substructure is not required, but the orient field is used to indicate individual slice orientation.

File Name Extensions

.HDR, .IMG

.HDR, which describes the data set; and .IMG, which contains the image data.

Format(s) Supported for Import

3D uniform lattice; 8-, 16-, 32-bit integer, float, double

Format(s) Supported for Export

3D uniform lattice; 8-, 16-, 32-bit integer, float, double

Import Method

Choose the **File | Import** command.

Export Method

Select a lattice or image module and choose the **File | Save Data** command.

Export Options

See Export Blanking Options. See *Analyze 7.5 Medical Image .HDR, .IMG Export Options Dialog*.

GTopo30 .HDR, .STX, .DEM File Description

The GTOPO-30 filter imports uniform lattices from several two-dimensional elevation grid file formats.

File Description

GTOPO30 is a global digital elevation model (DEM) resulting from a collaborative effort led by the staff at the U.S. Geological Survey's EROS Data Center in Sioux Falls, South Dakota. Elevations in GTOPO30 are regularly spaced at 30-arc seconds (approximately 1 kilometer). GTOPO30 was developed to meet the needs of the geospatial data user community for regional and continental scale topographic data.

GTOPO30 is a global data set covering the full extent of latitude from 90 degrees south to 90 degrees north, and the full extent of longitude from 180 degrees west to 180 degrees east. The horizontal grid spacing is 30-arc seconds (0.008333333333333 degrees), resulting in a DEM having dimensions of 21,600 rows and 43,200 columns. The horizontal coordinate system is decimal degrees of latitude and longitude referenced to WGS84. The vertical units represent elevation in meters above mean sea level. The elevation values range from -407 to 8,752 meters. In the DEM, ocean areas have been masked as "no data" and have been assigned a value of -9999. Low lying coastal areas have an elevation of at least 1 meter, so in the event that a user reassigns the ocean value from -9999 to 0 the land boundary portrayal will be maintained. Due to the nature of the raster structure of the DEM, small islands in the ocean less than approximately 1 square kilometer will not be represented.

File Format

To facilitate electronic distribution, GTOPO30 has been divided into 33 smaller pieces, or tiles. The area from 60 degrees south latitude to 90 degrees north latitude and from 180 degrees west longitude to 180 degrees east longitude is covered by 27 tiles, with each tile covering 50 degrees of latitude and 40 degrees of longitude. Antarctica (90 degrees south latitude to 60 degrees south latitude and 180 degrees west longitude to 180 degrees east longitude) is covered by 6 tiles, with each tile covering 30 degrees of latitude and 60 degrees of longitude. The tiles names refer to the

longitude and latitude of the upper-left (northwest) corner of the tile. For example, the coordinates of the upper-left corner of tile E020N40 are 20 degrees east longitude and 40 degrees north latitude. There is one additional tile that covers all of Antarctica with data in a polar stereographic projection.

File Name Extensions

GTOPO-30 .HDR, .STX, .DEM

A GTopo30 data set consists of multiple files. The following files must exist in the same directory:

.HDR contains header information

.STX the statistics file, used to obtain zmin, zmax

.DEM contains the actual grid nodes

Format(s) Supported for Import

2D uniform lattice; 8-, 16-, 32-bit integer, float, double

Import Method

Choose the **File | Import** command.

Remarks

GTopo30 files are extremely large. The files can be read, though approximately 220 MB of RAM is needed for this operation due to the grid size. You must have the associated .DEM, .HDR, and .STX files to use GTopo30 files. These files are necessary for computing the X, Y, and Z limits. Some GTopo30 files contain a "no data" value of -9999 in the file. The no data value is set to zero elevation.

HGT NASA SRTM Grid Data File Description

Voxler can import HGT binary grid files. **Voxler** can import HGT binary grid files directly from a compressed (zipped) file, if desired. HGT Files are produced by the Shuttle Radar Topography Mission, of NASA.

The SRTM data sets result from a collaboration between NASA, National Geospatial-Intelligence Agency, German space agency, and the Italian space agency. The files contain near-global DEM files of the Earth using radar interferometry. The SRTM instrument consisted of the Spaceborne Imaging Radar-C (SIR-C) hardware set modified with a Space Station-derived mast and additional antennae to form an interferometer with a 60 meter long baseline.

The SRTM data have undergone a sequence of processing steps resulting in several data versions having slightly different characteristics. In addition, the different naming conventions used by the NGA and NASA can lead to some confusion. In the first step raw SRTM radar echo data were processed in a systematic fashion using the SRTM Ground Data Processing System (GDPS) supercomputer system at the Jet Propulsion Laboratory. This processor transformed the radar echoes into strips of digital elevation data, one strip for each of the 1000 or so data swaths. These strips were then mosaicked into just less than 15,000 one degree by one degree cells and formatted according to the Digital Terrain Elevation Data (DTED) specification for delivery to NGA, who are using it to update and extend their DTED products. The DTED specification can be found in MIL-PDF-89020b.pdf on this server.

The data were processed on a continent-by-continent basis beginning with North America and proceeding through South America, Eurasia, Africa, Australia and Islands, with data from each continent undergoing a "block adjustment" to reduce residual errors. These data were also reformatted into the SRTM format, detailed in Section 3 below, and placed on this server as Version 1.0.

In the next step NGA applied several post-processing procedures to these data including editing, spike and well removal, water body leveling and coastline definition as described in the document SRTM_Edit_Rules.doc on this server. Following these "finishing" steps data were returned to NASA for distribution to the scientific and civil user communities as well as the public. These data were also reformatted into the SRTM format and are referred to as Version 2. The figure below shows a portion of cell N34W119.hgt, demonstrating the difference between the edited and unedited data.

Data Formats

The names of individual data tiles refer to the longitude and latitude of the lower-left (southwest) corner of the tile (this follows the DTED convention). For example, the coordinates of the lower-left corner of tile N40W118 are 40 degrees north latitude and 118 degrees west longitude. To be more exact, these coordinates refer to the geometric center of the lower left sample, which in the case of SRTM3 data will be about 90 meters in extent.

SRTM1 data are sampled at one arc-second of latitude and longitude and each file contains 3601 lines and 3601 samples. The rows at the north and south edges as well as the columns at the east and west edges of each cell overlap and are identical to the edge rows and columns in the adjacent cell.

SRTM3 data are sampled at three arc-seconds and contain 1201 lines and 1201 samples with similar overlapping rows and columns. This organization also follows the DTED convention. Unlike DTED, however, 3 arc-second data are generated in each case by 3x3 averaging of the 1 arc-second data - thus 9 samples are combined in each 3 arc-second data point. Since the primary error source in the elevation data has the characteristics of random noise this reduces that error by roughly a factor of three.

This sampling scheme is sometimes called a "geographic projection", but of course it is not actually a projection in the mapping sense. It does not possess any of the characteristics usually present in true map projections, for example it is not conformal, so that if it is displayed as an image geographic features will be distorted. However it is quite easy to handle mathematically, can be easily imported into most image processing and GIS software packages, and multiple cells can be assembled easily into a larger mosaic (unlike the pesky UTM projection, for example.)

Refer to NASA/JPL SRTM web site for additional information.

Import Options Dialog

Specify grid import options in the **Lattice Import Options** dialog. There are no HGT specific import options.

Export Options Dialog

Voxler does not currently export this file format.

Image (Bitmap) File Description

A bitmap is an image displayed as an array of dots or "bits."

Device-independent bitmaps and and BMP file format

A typical BMP file usually contains the following blocks of data:

BMP File Header	Stores general information about the BMP file
Bitmap Information (DIB header)	Stores detailed information about the bitmap image.
Color Palette	Stores the definition of the colros being used for indexed color bitmaps.
Bitmap Data	Stores the actual image, pixel by pixel

The following sections discuss the data stored in the BMP file or DIB in details. This is the standard BMP file format.[2] Some bitmap images may be stored using a slightly different format, depending on the application that creates it. Also, not all fields are used; a value of 0 will be found in these unused fields.

DIBs in memory

A BMP file is loaded into memory as a DIB data structure, an important component of the Windows GDI API. The DIB data structure is the same as the BMP file format, but without the 14-byte BMP header.

BMP file header

This block of bytes is at the start of the file and is used to identify the file. A typical application reads this block first to ensure that the file is actually a BMP file and that it is not damaged. Note that the first two bytes of the BMP file format (thus the BMP header) are stored in big-endian order. This is the magic number 'BM'. All of the other integer values are stored in little-endian format (i.e. least-significant byte first).

Offset#	Size	Purpose
0	2	the magic number used to identify the BMP file: 0x42 0x4D (Hex code points for **B** and **M**)
2	4	the size of the BMP file in bytes
6	2	reserved; actual value depends on the application that creates the image
8	2	reserved; actual value depends on the application that creates the image
10	4	the offset, i.e. starting address, of the byte where the bitmap data can be found.

Bitmap information (DIB header)

This block of bytes tells the application detailed information about the image, which will be used to display the image on the screen. The block also matches the header used internally by Windows and OS/2 and has several different variants. All of them contain a dword field, specifying their size, so that an application can easily determine which header is used in the image. The reason that there are different headers is that Microsoft extended the DIB format several times. The new extended headers can be used with some GDI functions instead of the older ones, providing more functionality. Since the GDI supports a function for loading bitmap files, typical Windows applications use that functionality. One consequence of this is that for such applications, the BMP

formats that they support match the formats supported by the Windows version being run. See the table below for more information.

Size	Header	Identified by	Supported by the GDI of
40	Windows V3	BITMAPINFOHEADER	all Windows versions since Windows 3.0
12	OS/2 V1	BITMAPCOREHEADER	OS/2 and also all Windows versions since Windows 3.0
64	OS/2 V2		
108	Windows V4	BITMAPV4HEADER	all Windows versions since WIndows 95/NT4
124	Windows V5	BITMAPV5HEADER	WIndows 98/2000 and newer

For compatibility reasons, most applications use the older DIB headers for saving files. With OS/2 being obsolete, for now the only common format is the V3 header.

Color Palette

The palette occurs in the BMP file directly after the BMP header and the DIB header. Therefore, its offset is the size of the BMP header plus the size of the DIB header.

The palette is a block of bytes (a table) listing the colors available for use in a particular indexed-color image. Each pixel in the image is described by a number of bits (1, 4, or 8) which index a single color in this table. The purpose of the color palette in indexed-color bitmaps is to tell the application the actual color that each of these index values corresponds to.

A DIB always uses the RGB color model. In this model, a color is terms of different intensities (from 0 to 255) of the additive primary colors red (R), green (G), and blue (B). A color is thus defined using the 3 values for R, G and B (though stored in backwards order in each palette entry).

The number of entries in the palette is either 2n or a smaller number specified in the header (in the OS/2 V1 format, only the full-size palette is supported).[2][4] Each entry contains four bytes, except in the case of the OS/2 V1 versions, in which case there are only three bytes per entry.[4] The first (and only for OS/2 V1) three bytes store the values for blue, green, and red, respectively,[2] while the last one is unused and is filled with 0 by most applications.

As mentioned above, the color palette is not used when the bitmap is 16-bit or higher; there are no palette bytes in those BMP files.

Bitmap Data

This block of bytes describes the image, pixel by pixel. Pixels are stored "upside-down" with respect to normal image raster scan order, starting in the lower left corner, going from left to right, and then row by row from the bottom to the top of the image.[2] Uncompressed Windows bitmaps can also be stored from the top row to the bottom, if the image height value is negative.

RGB color (24-bit) pixel values are stored with bytes in the same order (blue, green, red) as in the color table.[2]

If the number of bytes matching a row (scanline) in the image is not divisible by 4, the line is padded with one to three additional bytes of unspecified value (not necessarily 0) so that the next row will start on a multiple of 4 byte location in memory or in the file. (the total number of bytes in a row can be calculated as the image size/bitmap height in pixels) Following these rules there are

several ways to store the pixel data depending on the color depth and the compression type of the bitmap.

File Formats

The term image (bitmap) includes the following file formats:

- BMP Bitmap .BMP
- EPS Encapsulated Postcript .EPS
- GIF Graphics Interchange Format .GIF
- JPEG Compressed Bitmap .JPG
- JP2 JPEG-2000 Compressed Bitmap .JP2
- PCX ZSoft Paintbrush .PCX
- PNG Portable Network Graphics .PNG
- PNM Portable Any Map Image .PNM, .PPM, .PGM, .PBM
- RAS Sun Raster Image .RAS, .SUN
- RGB Silicon Graphics RGB Image .RGB, .RGBA, .BW
- SID LizardTech MrSID Image .SID
- TGA TrueVision Targa .TGA
- TIF Tagged Image File Format .TIF, .TIFF

Import Method

See the file format pages for import methods.

Export Method

See the file format pages for export methods.

ERDAS Imagine .IMG File Description

Voxler imports ERDAS Imagine IMG files. ERDAS Imagine uses .img files to store raster data with the ERDAS Imagine Hierarchal File Format (HFA) structure. The contents of an ERDAS Imagine .IMG file is not fixed and can be an image or a grid.

Import Options Dialog

If the ERDAS Imagine IMG file is an image file, no import options dialog is displayed. If the IMG file contains grid data, the **Lattice Import Options** dialog is displayed.

Import Automation Options

No import options are available.

Leica Confocal Scanning .INFO Files

The Leica Confocal Scanning filter imports image data from Leica Confocal Scanning (LEICA) or LCS Microscope data sets. Each data set consists of one or more image slices.

File Description

There are several different data set formats generated by the various Leica Laser Confocal Scanning .LCS Microscopes. The format supported by the import filter is the microscope's NATIVE (or "raw") storage format. This format uses an .INFO file to describe a series of images slices stored in a raw pixel array format. The base name of the .INFO file is also used for each of the images, e.g., if "mydata.info" is the header, then the images might appear in "mydata.12" or "mydata.13." The numbering of the image files is specified in the range parameters in the .INFO file. The first image is not always image number zero.

The microscope's software also has the option of storing image slices as TIFF files, which are not supported by this particular filter; however, Golden Software does provide a TIFF import filter for reading those.

File Format

The base name of the .INFO file is also used for each of the images, e.g., if " mydata.info" is the header, then the images might appear in "mydata.12" or "mydata.13." The numbering of the image files is specified in the range parameters in the .INFO file. The first image is not always image number zero.

The microscope's software also has the option of storing image slices as .TIFF files, which are not supported by this particular filter; however, Golden Software does provide a .TIFF import filter that works fine for those.

File Name Extensions

.INFO

Format(s) Supported for Import

3D uniform lattice; 8-bit integer

Import Method

Choose the **File | Import** command.

Inventor .IV Files

The IV filter imports and exports geometric figures from SGI Open Inventor files.

File Description

Inventor is the underlying code used to display the three-dimensional objects. **Voxler** uses Inventor code to display objects in the **Viewer** window. Originally from Silicon Graphics, it has undergone a number of customizations and changes, including IRIS and Coin. **Voxler** uses Inventor code to display objects in the **Viewer** window.

File Name Extensions

.IV

Format(s) Supported for Import

Voxler geometry object (Inventor scene graph)

Format(s) Supported for Export

Voxler geometry object (Inventor scene graph)

Import Method

Choose the **File | Import** command.

Export Method

Choose the **File | Export** command.

Other Notes

This filter uses the Open Inventor file read/write functionality provided by COIN, a third-party library that is compatible with Open Inventor.

JP2 JPG 2000 File Interchange Format .JP2, J2K, .JPC, .JPT, .JPEG2000, .J2000 File Description

Voxler imports and exports JP2 JPEG 2000 raster image files. The JPEG 2000 File Interchange Format, .JP2, is a image file format standard with additional wavelet compression techniques. It is a format for exchanging JP2 encoded files compliant with the JPEG Interchange Format standard. This format is based on the ISO standard (ISO 15444/6). JPEG2000 offers both lossless and lossy compression and creates smaller file sizes than JPG exports. The image is also better quality than the traditional JPG format.

Import Options Dialog

No import options dialog is displayed.

Import Automation Options

See *JP2 Import Options* for additional information.

Export Options Dialog

See *Size and Color* and *JPEG-2000 Options*

Export Automation Options

See *JP2 Export Options* for additional information.

JPEG Compressed Bitmap .JPG, .JPEG Files

The JPEG filter imports and exports bitmap images.

File Description

In computing, JPEG (named after the Joint Photographic Experts Group who created the standard) is a commonly used method of lossy compression for photographic images. The degree of compression can be adjusted, allowing a selectable tradeoff between storage size and image

quality. JPEG typically achieves 10:1 compression with little perceptible loss in image quality. JPEG compression is used in a number of image file formats. JPEG/Exif is the most common image format used by digital cameras and other photographic image capture devices; along with JPEG/JFIF, it is the most common format for storing and transmitting photographic images on the World Wide Web. These format variations are often not distinguished, and are simply called JPEG.

File Name Extensions

.JPG, .JPEG

Format(s) Supported for Import

device-independent bitmap; 1, 4, 8, 24 bit per pixel

Format(s) Supported for Export

device-independent bitmap; 8, 24 bit per pixel

Import Method

Choose the **File | Import** command.

Export Method

Choose the **File | Export** command or select an image module and choose the **File | Save Data** command.

Export Options

Specify the export options in the **Export Options** dialog and the **JPEG Image Export Options** dialog.

Google Earth Keyhole Markup .KML and. KMZ File Description

Voxler can import .KML and .KMZ files. KML and KMZ files are imported into **Voxler** as a *geometry source* module.

.KML and .KMZ files are used by Google Earth to display information, such as contour maps or point locations, on Google Earth maps. Both .KML and .KMZ files contain the same information. The difference is that .KMZ files are compressed versions of .KML files.

Coordinate System

.KML and .KMZ files require that the coordinates be in latitude and longitude degrees.

Remarks

Voxler does not import marker symbols, labels, icons, or text in KML and KMZ files. If the marker symbol can be exported to the KML or KMZ file as rendered curves, as when exporting a plot in Golden Software's **Surfer**, then the lines representing the marker symbol will be imported into **Voxler**.

When an imported KML or KMZ file contains point locations as marker symbols or icons, as when exported from Google Earth, a *point source* module is also created upon import. The point locations can be represented by connecting a *ScatterPlot* module to the point source module.

Import Options Dialog

No import options are displayed.

LAS File Description

Log ASCII Standard (.LAS) files are a type of well data file that is common to many logging operations.

File Description

The file format is maintained by the Canadian Well Logging Society. and comes in various versions. Each file can contain different logs, different header information, and different parameter information. **Voxler** reads the well name and log information from the .LAS file. Specific logs can be imported in the **LAS Import Options Dialog**. Trajectory information and collar data is not imported from the .LAS file. This data will need to be imported from a data file.

File Format

The top section of a .LAS file contains version information. This determines the .LAS file format compatibility. **Voxler** imports version 1.2, 2.x, and most 3.x LAS files. The next section contains well information. The starting and ending depth values, and the location information is in this section. Although this information is not read by **Voxler**, the file can be opened in a text editor and copied into the *Set well top* section in the **LAS Import Options** dialog. The next section contains curve names and units. The curve names are used to name the logs in **Voxler**. The final section contains the log data values. These are imported into the various logs for the well.

File Name Extensions

.LAS

Import Method

Choose the **File | Import** command.

Import Options

Specify the import options in the **LAS Import Options** dialog.

LAS LiDAR Binary File Description

The ASPRS LiDAR LAS binary file format stores 3D point data. LiDAR stands for Light Detection and Ranging data. This data is generally created by software which combines GPS, IMU, and laser pulse range data to produce X, Y, and Z point data. To use binary LiDAR data, click **File | Open** and select the LAS or click **File | Import**.

File Format

The format contains binary data consisting of a header block, variable length records, and point data. All data is in little-endian format. The header block consists of a public block followed by variable length records. The public block contains generic data such as point numbers and

coordinate bounds. The variable length records contain variable types of data including projection information, metadata, and user application data.

Currently, **Voxler** imports 1.0, 1.1, 1.2, and 1.3 format LAS files. These can contain up to 32 codes.

Classification Number	Meaning
0	Created, never classified
1	Unclassified
2	Ground
3	Low Vegetation
4	Medium Vegetation
5	High Vegetation
6	Building
7	Low Point (noise)
8	Model Key-point
9	Water
10	Rail
11	Road Surface
12	Reserved
13	Wire - Guard (Shield)
14	Wire - Conductor (Phase)
15	Transmission Tower
16	Wire - Structure Connector (e.g. Insulator)
17	Bridge Deck
18	High Noise
19-32	Reserved

Import Options Dialog

See *LiDAR Import Filtering Dialog*

Import Automation Options

See *LiDAR Import Automation Options*

Export Options Dialog

Voxler does not currently export LiDAR LAS data

Export Automation Options

Voxler does not currently export LiDAR LAS data

Iris Explorer .LAT Files

The LAT filter imports and exports IRIS Explorer data sets.

File Description

IRIS Explorer is an Open Inventor based product. Open Inventor provides a *de facto* standard for 3D scene definition and was the basis of the VRML (Virtual Reality Modeling Language) extension to HTML and the World Wide Web.

The IRIS Explorer *Lattice* data type is a generalised multi-dimensional array. It differs from arrays in other languages by being able to store both node-based values and coordinate information for each of the nodes. Being extremely flexible, it is the most commonly used data type in IRIS Explorer, used almost exclusively to store data and pass it between modules.

The *Data* part of a lattice can be multi-dimensional. If required, multiple data values can be stored at each node.

The *Coordinate* part of a lattice can have a varying number of dimensions, dependent on the number of dimensions that the data space occupies. There are three different types of coordinate information: uniform, perimeter, curvilinear.

File Name Extensions

.LAT

Format(s) Supported for Import

- 3D uniform lattice
- 3D rectilinear lattice
- 3D curvilinear lattice

Format(s) Supported for Export

- 3D uniform lattice
- 3D rectilinear lattice
- 3D curvilinear lattice

Import Method

Choose the **File | Import** command.

Export Method

Select a data, lattice, or image module and choose the **File | Save Data** command.

Export Options

See *Iris Explorer .LAT Export Options Dialog*. See Export Blanking Options.

NetCDF .NC File Description

Voxler can import and export NetCDF .NC lattice file formats. These files are uniform grids.

NetCDF is a set of software libraries and self-describing, machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data. It is also a community standard for sharing scientific data. NetCDF was developed and is maintained at Unidata, part of the University Corporation for Atmospheric Research (UCAR) Community Programs (UCP). Unidata is funded primarily by the National Science Foundation.

A NetCDF file may contain multiple variables. An import options dialog shows all importable variables available in the file. Only one variable may be imported at a time, and only the selected variable will be imported.

Import Options Dialog

NetCDF Import Options Dialog

Import Automation Options

See *NetCDF Import Automation Options*

Loading Files

Use the **File | Import** command to load a NetCDF .NC file.

Export Options

There are no NetCDF export options.

MapInfo Interchange Format .MIF File Description

The MapInfo Interchange Format imports .MIF files.

File Description

MapInfo Interchange Format .MIF files contain boundary objects including areas, curves and points. The objects optionally have attributes (such as color or line style) associated with them.

File Name Extensions

.MIF, .MID

Each MIF file is usually accompanied by a file with the same name, but with the .MID filename extension. The MID file contains attribute information about the objects in the map. This information is imported in the objects' Primary ID and Secondary ID.

Import Method

Choose the **File | Import** command.

Microsoft Access .ACCDB and .MDB File Description

The Microsoft Access filter imports .MDB and .ACCDB files.

File Description

Microsoft Access .MDB is a binary database file format used by pre-2007 versions of Microsoft Access. The .ACCDB format is a binary database file format used by Microsoft Access 2007 and 2010.

File Name Extensions

.ACCDB, .MBD

Import Method

Choose the **File | Import** command.

Import Options

See *Microsoft Access .ACCDB .MBD Import Options* dialog.

64-Bit Access Driver

In order to import Microsoft Access Database (*.mdb, *.accdb) files, you must have the Microsoft Access Database driver installed on your machine. It's shipped as part of the Microsoft Office suite and comes in 32-bit and 64-bit versions. Installing the 32-bit Microsoft Office suite will install the 32-bit Access Database driver. Installing the 64-bit Microsoft Office suite will install the 64-bit Access Database driver. Unfortunately, Microsoft doesn't allow BOTH to be installed simultaneously on a 64-bit Windows platform. If you need to import data from Microsoft Access Database files into Golden Software products, you must install the 32-bit version of our product if you have a 32-bit Access Database driver. You must install the 64-bit version of our product if you have a 64-bit Access Database driver. If you don't need to import Access Database data with our product, you may install either version on a 64-bit Windows platform.

ZSoft Paintbrush .PCX File Description

The ZSoft Paintbrush filter imports and exports ZSoft Paintbrush bitmaps.

File Description

The .PCX is an image file format developed by the ZSoft Corporation of Georgia, USA. It was the native file format for PC Paintbrush (PCX = "PC Paintbrush Exchange") and became one of the first widely accepted DOS image standards, although its use has since been succeeded by more sophisticated image formats such as GIF, JPEG, and PNG. PCX is used in Microsoft Windows and Windows-based products but has found wide acceptance mainly in the MS-DOS world. It is mainly an exchange and storage format. PCX is a format in wide use, which is quick and easy to read and decompress. It lacks, however, a superior compression scheme, making it unsuitable for the storage of deep-pixel images.

The .PCX file is a device-independent raster image format; the file header stores information about the display hardware (screen resolution, color depth and palette information, and bit planes)

separately from the actual image information, allowing the image to be properly transferred and dsiplayed on computer systems with different hardware. PCX files commonly store palette-indexed images ranging from 2 or 4 colors to 16 and 256 colors, although the format has been extended to record true-color (24-bit) images as well.

File Header

PCX files are organized into three major sections: the header, the image data, and the color palette. The color palette normally contains entries for 256 colors and is associated with the VGA display adapter. This VGA color palette is only found in later versions of the PCX image file format.

The PCX file header contains an identifier byte (value 10), a version number, image dimensions, a 16 color palette, number color planes and the bit depth of each plane. While the file header could describe a wide variety of image formats, many are not of practical use. Convention has limited the supported combinations of plane count and bit depth to match specific PC display hardware. Many image editing programs will not read PCX files which do not match one of these conventions. Table A shows the most commonly supported combinations.

The header also contains a value for compression method. All PCX files are written with the same compression scheme and this value is always 1. No other values have been defined and there are no uncompressed PCX files.

PCX version numbers range from 0 to 5, though the file format does not change between versions.

The header is always 128 bytes long though only 74 bytes are used. The rest of the 128 bytes is padded and the image data always begins 128 bytes after the start of the file.

File Name Extensions

.PCX

Format(s) Supported for Import

device-independent bitmap; 1, 4, 8, 24 bit per pixel

Format(s) Supported for Export

device-independent bitmap; 8, 24 bit per pixel

Import Method

Choose the **File | Import** command.

Export Method

Choose the **File | Export** command or select an image module and choose the **File | Save Data** command.

Export Options

Specify the export options in the **Export Options** dialog.

Adobe Portable Document Format .PDF File Description

The Adobe Portable Document filter exports .PDF files.

File Description

Portable Document Format .PDF is a file format used for document exchange. Portable document format is used for representing two-dimensional documents. PDF was created by Adobe Systems.

File Format

A .PDF file consists primarily of eight object types.

- *Boolean* values, representing true or false
- *Numbers*
- *Strings*
- *Names*
- *Arrays*, ordered collections of objects
- *Dictionaries*, collections of objects indexed by *Names*
- *Streams*, usually containing large amounts of data
- The *Null* object

File Name Extensions

.PDF

GeoPDF

A GeoPDF file is a PDF file with a georeferenced coordinate system. **Voxler** can import a GeoPDF file and will honor the defined coordinate system.

Import Options

Use **File | Import** to import a *PDF (Raster) (*.PDF)* or *PDF (Vector) (*.PDF)* file. The **PDF Import Options** dialog is displayed.

Export Options

Use the **File | Export** command to export as a *PDF (Raster) (*.PDF)* file.

Export Method

Choose the **File | Export** command or choose the **File | Save Data** command to save as a *PDF (Raster) (*.PDF)* file. Alternately, use the **File | Print** command to print to a PDF driver if you have one installed.

Golden Software PlotCall .PLT File Description

The Golden Software PlotCall filter imports Golden Software PlotCall .PLT files.

File Description

PlotCall files .PLT contain line graphics designed to be output on pen plotters. The curve, point and text objects of a PlotCall file dan be imported. There is no capability to export PlotCall files.

File Format

In the PlotCall file, each pen used is assigned a number. There can be up to 16 pens used in a PlotCall file.

PlotCall files can be either ASCII files (i.e., they can be edited with a text editor or word processor) or binary files (can't be edited) containing commands. Each command occupies one record and begins with a two-letter operation code (op-code) to determine its function. The currently supported op-codes are:

Op-code	Function
MA	Move Absolute
PA	Plot Absolute
TR	Translate
SC	Scale
PS	Plot String
SS	Set Symbol Set
RO	Rotate
PI	Pivot
SP	Select Pen

The general format of most commands is "op-code X,Y" where X and Y are coordinate values measured in inches. See Golden Software's PlotCall manual for a detailed description of each command. PlotCall files are usually produced by Golden Software's DOS applications, or by special user-written programs on PCs or mainframes.

File Name Extensions

.PLT

Import Method

Choose the **File | Import** command.

Import Options

No import options are available.

Stanford Polygon .PLY Files

The Standard Polygon filter imports geometric data from Stanford Polygon .PLY data sets.

File Description

PLY is a computer file format known as the Polygon File Format or the Stanford Triangle Format. The format was designed to store three-dimensional data from 3D scanners. It supports a relatively simple description of a single object as a list of nominally flat polygons. A variety of properties can be stored including: color and transparency, surface normals, texture coordinates and data confidence values. The format permits one to have different properties for the front and back of a polygon.

File Format

Files are organised as a header, that specifies the elements of a mesh and their types, followed by the list of elements itself, usually vertices and faces eventually other entities such as edges, samples of range maps, and triangle strips can be encountered.

The header of both ASCII and binary files is ASCII text. Only the numerical data that follows the header is different between the two versions.

The header always starts with the line 'ply'. The header helps to identify this as a genuine 'PLY' file.

```
ply
```

The second line indicates which variation of the PLY format this is. It should be one of:

```
format ascii 1.0
format binary_little_endian 1.0
format binary_big_endian 1.0
```

Future versions of the standard will change the revision number at the end - but 1.0 is the only version currently in use.

Comments may be placed in the header by using the word 'comment' at the start of the line. Everything from there until the end of the line should then be ignored. eg:

```
comment This is a comment!
```

The 'element' keyword introduces a description of how some particular data element is stored and how many of them there are. Hence, in a file where there are 12 vertices, each represented as a floating point (X,Y,Z) triple, one would expect to see:

```
element vertex 12
property float x
property float y
property float z
```

Other 'property' lines might indicate that colours or other data items are stored at each vertex and indicate the data type of that information. Regarding the data type there are two variants, depending on the source of the ply file, the type can be specified with one of *char uchar short ushort int uint float double*, or one of *int8 uint8 int16 uint16 int32 uint32 float32 float64*. For an object with ten polygonal faces, one might see:

```
element face 10
property list uchar int vertex_index
```

The word 'list' indicates that the data is a list of values - the first of which is the number of entries in the list (represented as a 'uchar' in this case) and each list entry is (in this case) represented as an 'int'.

At the end of the header, there must always be the line:

end_header

In the ASCII version of the format, the vertices and faces are each described one to a line with the numbers separated by white space. In the binary version, the data is simply packed closely together at the 'endianness' specified in the header and with the data types given in the 'property' records. For the common "property list..." representation for polygons, the first number for that element is the number of vertices that the polygon has and the remaining numbers are the indices of those vertices in the preceding vertex list.

File Name Extensions

.PLY

Format(s) Supported for Import

Voxler geometry object (Inventor scene graph)

Import Method

Choose the **File | Import** command.

References

This implementation is based on the file format described in Greg Turk's 1998 "The PLY Polygon File Format" document, which can be found in numerous places on the Internet.

Portable Network Graphic .PNG File Description

The Portable Network Graphic filter imports and exports Portable Network Graphic .PNG files.

File Description

Portable Network Graphics .PNG is an image format that employs lossless data compression. PNG was created to improve upon and replace GIF (Graphics Interchange Format) as an image-file format not requiring a patent license.

The .PNG format is one of the few image formats with complete support for alpha channels (transparency). PNG supports palette-based, greyscale, or RGB images. PNG does not support CMYK color spaces.

File Name Extensions

.PNG

Format(s) Supported for Import

device-independent bitmap; 1, 4, 8, 24 bit per pixel

Format(s) Supported for Export

device-independent bitmap; 8, 24 bit per pixel

Import Method

Choose the **File | Import** command.

Export Method

Choose the **File | Export** command or choose the **File | Save Data** command.

Export Options

See *Export Options - Size and Color* page.

See *PNG Export Options* dialog.

Portable Any Map .PNM File Description

The Portable Any Map filter imports and exports Portable Any Map .PNM, Portable Pixel Map .PPM, Portable Gray Map .PGM, and Portable Bitmap .PBM files.

File Description

Although there are both binary and ASCII variants of this file format, the current version of the filter always writes binary files when exporting.

The portable pixmap file format .PPM, the portable graymap file format .PGM and the portable bitmap file format .PBM specify rules for exchanging graphics files. They provide very basic functionality and serve as a least-common-denominator for converting pixmap, graymap, or image files between different platforms. Several applications refer to them collectively as the PNM format (portable anymap).

The .PGM and .PPM formats (both ASCII and binary versions) have an additional parameter for the maximum value in a line between the X and Y dimensions and the actual pixel data.

File Name Extensions

.PNM, .PGM, .PPM

Import Method

Choose the **File | Import** command.

Export Method

Choose the **File | Export** command or select an image module and choose the **File | Save Data** command.

Export Options

Specify export options in the **Export Options** dialog, on the **Size and Color Page**.

PLOT-3D .P3D, .XYZ Files

The PLOT-3D filter imports Plot-3D data sets.

File Description

A Plot-3D data set consists of a coordinate file and (optionally) a solution file. The coordinate file contains three-dimensional X,Y,Z coordinates in several possible formats. The solution file (if present) contains several columns of data values associated with each of the coordinates. P3D is a system that originated at Carnegie Mellon University's Pittsburgh Supercomputing Center, which retains the copyright for the system. The P3D format used by the P3D system was intended for the storage of 3D models and was designed to be portable, flexible, compact, and extensible. The authors wished to create a format that would be compatible with applications, renderers in particular, on a number of platforms.

Since a Plot-3D data set does not contain any header information to indicate the specific format of the file, the user is required to provide some information about the file format in the **Import Options** dialog. A detailed discussion of the various format options for Plot-3D files is beyond the scope of this document.

File Format

A P3D file consists of a number of ASCII lines that are usually Common Lisp statements. Extensions to the language are mainly in an idiosyncratic terminology that is unfortunately at odds with most of the rest of the computer graphics world.

File Name Extensions

coordinate file .P3D, .XYZ; optional solution file .Q

Format(s) Supported for Import

3D curvilinear lattice; float

Format(s) Supported for Export

3D curvilinear lattice; float

Import Options

Several import options appear in the **PLOT3D Import Options** dialog.

Export Options

Several options appear in the **PLOT3D Export Options** dialog.

Export Restrictions/Limitations

If doubles are exported, they are truncated to floats in order to be compatible with the PLOT-3D file format. When this occurs, a warning is displayed.

Sun Raster Image .RAS, .SUN Files

The Sun Raster Image filter imports and exports Sun Raster .RAS and .SUN image files.

File Description

The Sun Raster .RAS .SUN file format originated at Sun Microsystems and is a common file format for storing bitmap images on UNIX and Solaris workstations.

File Name Extensions

.RAS, .SUN

Format(s) Supported for Import

device-independent bitmap; 1, 4, 8, 24 bit per pixel

Format(s) Supported for Export

device-independent bitmap; 8, 24 bit per pixel

Import Method

Choose the **File | Import** command.

Export Method

Choose the **File | Export** command or choose the **File | Save Data** command.

Export Options

Specify the export options in the **Export Options** dialog.

RAW Binary Lattice .RAW, .BIN File Description

The RAW Binary Lattice filter imports and exports uniform lattices from raw (untyped) binary files.

File Description

The RAW file is a simple headerless binary file consisting of lattice nodes arranged according to the export options that were selected when the file was created. No specific filename extension is required, but .RAW and .BIN are commonly used for these files.

File Format

The lattice dimensions and several other aspects of the file format can be controlled by the user.

File Name Extensions

.RAW, .BIN

Since this filter works with the headerless/untyped files, it is important to knows what options are appropriate for the data in question.

Format(s) Supported for Import

3D uniform lattice; 8-, 16-, 32-bit integer, float, double

Format(s) Supported for Export

3D uniform lattice; 8-, 16-, 32-bit integer, float, double

Import Method

Choose the **File | Import** command.

Import Options

Specify import options in the **Lattice Import Options** dialog and **RAW Import Options** dialog.

Export Method

Select a lattice or image module and choose the **File | Save Data** command.

Export Options

See the **RAW Export Options** and Export Blanking Options.

Silicon Graphics (SGI) RGB Image .RGB, .RGBA, .BW Files

The Silicon Graphics Image filter imports and exports Silicon Graphics Image .RGB, .RGBA, and .BW files.

File Description

The .RGB, .RGBA, .BW formats originated on Silicon Graphics workstations and is/was used in a variety of high-end imaging applications, both Unix- and Windows-based.

File Name Extensions

.RGB, .RGBA, .BW

The filename extension is sometimes used to indicate the format of the image contained in the file, but is not required to do so. The extensions are typically .BW for black and white images, .RGB for 24-bit color images, and .RGBA for 32-bit color images with an alpha channel.

Format(s) Supported for Import

device-independent bitmap; 24, 32 bit per pixel color and 8-bit grayscale

Format(s) Supported for Export

device-independent bitmap; 24, 32 bit per pixel color and 8-bit grayscale

Import Method

Choose the **File | Import** command.

Export Method

Choose the **File | Export** command or select an image module and choose the **File | Save Data** command.

Export Options

Specify the export options in the **Export Options** dialog.

Import Restrictions/Limitations

Although the industry convention is for the file name extension .RGB to be used for 24-bit images, .RGBA to be used for 32-bit images, and .BW to be used for 8-bit grayscale images, the filter examines the contents of the file header to determine the pixel format and ignores the file name extension.

Idrisi Raster Image .RST, .IMG File Description

The Idrisi Raster Image filter imports the Idrisi Raster Image .RST format. It requires the associated .RCS file to be present in the same directory. Sometimes there is also an associated .REF file, depending on the refsystem member of the IDR_RDC struct inside the file.

File Description

Idrisi grids have either a .RST or .IMG extension. Every data set come with a set of files including a .DOC or .RDC file. The Idrisi Raster Image filter supports all Idrisi grid types except the RGB type.

The associated .RDC file must be present in the same directory as the .RST. Sometimes there is also an associated .REF file, depending on the refsystem member of the IDR_RDC struct.

- The old IDrisi data sets consist of paired .IMG/.DOC.
- The New IDrisi data sets consist of paired .RST/.RDC.
- Allowable .RST/.IMG formats are ASCII, Binary and Packed Binary forms. However, most routines in the IDRISI application expect the binary format only.

File Format

IDRISI starts in the upper-left corner (row 0/column 0), then advances column by column and row by row.

File Name Extensions

.RST/.RDC, .IMG/.DOC

Import Method

Choose the **File | Import** command.

Import Options

Specify import options in the **Lattice Import Options** dialog

.SP1 SEG Standard Data Exchange File Description

The .SP1 SEG standard data exchange file format is a format widely used in the geophysical industry to exchange data for shotpoint locations for seismic surveying. **Voxler** currently imports version SP1 and SEG file formats. Shotpoint locations are usually those computed locations which are the best estimates of where actual data points are located in the field. The locations are derived from a variety of complex field data. This format is applicable for both land and marine locations

and has sufficient flexibility for use in 3D seismic surveys. This can include gravity, magnetic data, or other data about each shotpoint. For additional information on specifics about each file format, refer to the Society for Exploration Geophysicists.

SP1 files can be opened by clicking the **File | Open** command or by clicking the **File | Import** command in a worksheet window.

The SEG and SP1 file formats are fairly flexible and can include data in latitude and longitude or in easting, northing, and depth/elevation formats. When loading latitude/longitude data, the WGS84 datum is assumed.

Import Options Dialog

See *SEG SP1 Import Options*

SEG-Y Seismic Data Log .SGY, .SEGY Files

The SEGY filter imports SEG-Y seismic data log files as point sets or two-dimensional lattices.

File Description

Each SEG-Y file contains one or more "traces," which are digital representations of amplitude samples as recorded by a paper tape seismic recorder (or modern paperless equivalent).

File Name Extensions

.SGY, .SEGY

Format(s) Supported for Import

- 2D uniform lattice; 8-,16-bit integer, float
- point set; 8-,16-bit integer, float

Import Method

Choose the **File | Import** command.

Import Options

Several options appear in the **SEGY Import Options** dialog. Specify import options in the **Lattice Import Options** dialog

Remarks

It is possible for SEG-Y files to have zeros or nonsensical values in the trace coordinates. When this is the case, coordinates from 0 to N will be automatically generated upon import. The coordinate scaling information from the trace headers will be applied to the generated coordinates.

Import Restrictions/Limitations

Since many SEG-Y files in the real world deviate from the SEG-Y specification, the user may need to know exactly what format the SEG-Y data is encoded in and set the **Import Options** dialog sections appropriately; otherwise, the data may not import successfully.

Esri Shapefile .SHP File Description

The Esri Shapefile filter imports .SHP files. **Voxler** can import shapefiles in compressed (.TAR, .TAR.GZ, .ZIP, .TGZ) folders. However if the folder contains multiple shapefiles, only the first shapefile will be imported. The shapefiles must be extracted if you wish to import the subsequent shapefiles in the compressed folder.

File Description

The Esri Shapefile or simply a shapefile is a popular geospatial vector data format for geographic information systems software. It is developed and regulated by Esri as a (mostly) open specification for data interoperability among Esri and other software products. A shapefile commonly refers to a collection of files with .SHP, .SHX, .DBF, and other extensions on a common prefix name (e.g., "state"). The actual shapefile relates specifically to files with the .SHP extension, however this file alone is incomplete for distribution, as the other supporting files are required.

Shapefiles spatially describe geometries: points, polylines, and polygons. These, for example, could represent water wells, rivers, and lakes, respectively. Each item may also have attributes that describe the items, such as the *name* or *temperature*.

File Format

Esri Shapefiles are in a binary file format (i.e., they can't be created or modified with a text editor or word processor) that is compatible with Arc/Info, Arc/View, and other Esri application programs. This format is used to store spatial information including boundary objects such as areas, curves, and points. Spatial information is only concerned with the location of objects in space (i.e., their coordinates) and not with their attributes (such as line or fill style, marker symbol used, text labels, etc.).

Three types of files are produced with each export:

Filename Extension	Extension Format	Description
.SHP	Shape format	Contains the coordinates of each object in the drawing.
.SHX	Shape index format	Contains the file offset of each object in the .SHP file.
.DBF	Attribute format	Contains the attribute text associated with each object in the .SHP file.

In each of the .SHP, .SHX, and .DBF files, the shapes in each file correspond to each other in sequence. That is, the first record in the .SHP file corresponds to the first record in the .SHX and .DBF files, and so on. The .SHP and .SHX files have various fields with different endianness, so as an implementor of the file formats you must be very careful to respect the endianness of each field and treat it properly.

Mandatory files

.SHP - shape format; the feature geometry itself

.SHX - shape index format; a positional index of the feature geometry to allow seeking forwards and backwards quickly

.DBF - attribute format; columnar attributes for each shape, in dBase III format

Optional files

.PRJ - projection format; the coordinate system and projection information, a plain text file describing the projection using well-known text format

.SBN and .SBX - a spatial index of the features

.FBN and .FBX - a spatial index of the features for shapefiles that are read-only

.AIN and .AIH - an attribute index of the active fields in a table or a theme's attribute table

.IXS - a geocoding index for read-write shapefiles

.MXS - a geocoding index for read-write shapefiles (ODB format)

.ATX - an attribute index for the .dbf file in the form of shapefile.columnname.atx (ArcGIS 8 and later)

.SHP.XML- metadata in XML format

Shapefile shape format .SHP

The main file .SHP contains the primary geographic reference data in the shapefile. The file consists of a single fixed length header followed by one or more variable length records. Each of the variable length records includes a record header component and a record contents component. A detailed description of the file format is given in the Esri Shapefile Technical Description.[1] This format should not be confused with the AutoCAD shape font source format, which shares the .shp extension.

The main file header is fixed at 100 bytes in length and contains 17 fields; nine 4-byte (32-bit signed integer or int32) integer fields followed by eight 8-byte (double) signed floating point fields:

Bytes	Type	Endianness	Usage
0-3	int32	big	File code (always hex value 0x0000270a)
4-23	int32	big	Unused; five uint32
24-27	int32	big	File length (in 16-bit words, including the header)
28-31	int32	little	Version
32-35	int32	little	Shape type (see reference below)
36-67	double	little	Minimum bounding rectangle (MBR) of all shapes contained within the shapefile; four doubles in the following order: min X, min Y, max X, max Y
68-83	double	little	Range of Z; two doubles in the following order: min Z, max Z
84-99	double	little	Range of M; two doubles in the following order: min M, max M

The file then contains any number of variable-length records. Each record is prefixed with a record-header of 8 bytes:

Bytes	Type	Endianness	Usage
0-3	int32	big	Record number
4-7	int32	big	Record length (in 16-bit words)

Following the record header is the actual record:

Bytes	Type	Endianness	Usage
0-3	int32	big	Shape type (see reference below)
4-	-	-	Shape content

The variable length record contents depend on the shape type. The following are the possible shape types:

Value	Shape Type	Fields
0	Null shape	None
1	Point	X, Y
3	Polyline	MBR, Number of parts, Number of points, Parts, Points
5	Polygon	MBR, Number of parts, Number of points, Parts, Points
8	MultiPoint	MBR, Number of points, Points
11	PointZ	X, Y, Z, M
13	PolylineZ	*Mandatory*: MBR, Number of parts, Number of points, Parts, Points, Z range, Z array *Optional*: M range, M array
15	PolygonZ	*Mandatory*: MBR, Number of parts, Number of points, Parts, Points, Z range, Z array *Optional*: M range, M array
18	MultiPointZ	*Mandatory*: MBR, Number of points, Points, Z range, Z array *Optional*: M range, M array
21	PointM	X, Y, M
23	PolylineM	*Mandatory*: MBR, Number of parts, Number of points, Parts, Points *Optional*: M range, M array
25	PolygonM	*Mandatory*: MBR, Number of parts, Number of points, Parts, Points *Optional*: M range, M array
28	MultiPointM	*Mandatory*: MBR, Number of points, Points *Optional Fields*: M range, M array
31	MultiPatch	*Mandatory*: MBR, Number of parts, Number of points, Parts, Part types, Points, Z range, Z array *Optional*: M range, M array

In common use, shapefiles containing Point, Polyline, and Polygon are extremely popular. The "Z" types are three-dimensional. The "M" types contain a user-defined measurement which coincides with the point being referenced. Three-dimensional shapefiles are rather uncommon, and the measurement functionality has been largely superseded by more robust databases used in conjunction with the shapefile data.

Shapefile shape index format .SHX

The shapefile index contains the same 100-byte header as the .SHP file, followed by any number of 8-byte fixed-length records which consist of the following two fields:

Bytes	Type	Endianness	Usage
0-3	int32	big	Record offset (in 16-bit words)
4-7	int32	big	Record offset (in 16-bit words)

Using this index, it is possible to seek backwards in the shapefile by seeking backwards first in the shape index (which is possible because it uses fixed-length records), reading the record offset, and using that to seek to the correct position in the .SHP file. It is also possible to seek forwards an arbitrary number of records by using the same method.

Shapefile attribute format .DBF

Attributes for each shape are stored in the xBase (dBase) format, which has an open specification.

Shapefile projection format .PRJ

The projection information contained in the .PRJ file is critical in order to understand the data contained in the .SHP file correctly. Although it is technically optional, it is most often provided, as it is not necessarily possible to guess the projection of any given points. The file is stored in well-known text (WKT) format.

Some typical information contained in the .PRJ file is:

- Geographic coordinate system
- Datum (geodesy)
- Spheroid
- Prime meridian
- Map projection
- Units used
- Parameters necessary to use the map projection, for example:
- Latitude of origin
- Scale factor
- Central meridian
- False northing
- False easting
- Standard parallels

Shapefile spatial index format .SBN

This is a binary spatial index file, which is used only by Esri software. The format is not documented, and is not implemented by other vendors. The .SBN file is not strictly necessary, since the .SHP file contains all of the information necessary to successfully parse the spatial data.

Limitations

Topology and shapefiles

Shapefiles do not have the ability to store topological information. ArcInfo coverages and Personal/File/Enterprise Geodatabases do have the ability to store feature topology.

Spatial representation

The edges of a polyline or polygon are defined using points, which can give it a jagged edge at higher resolutions. Additional points are required to give smooth shapes, which requires storing quite a lot of data compared to, for example, bézier curves, which can capture complexity using smooth curves, without using as many points. Currently, none of the shapefile types support bézier curves.

Data storage

Unlike most databases, the database format is based on older xBASE standard, incapable of storing null values in its fields. This limitation can make the storage of data in the attributes less flexible. In ArcGIS products, values that should be null are instead replaced with a 0 (without warning), which can make the data misleading. This problem is addressed in ArcGIS products by using Esri's Personal Geodatabase offerings, one of which is based on Microsoft Access.

Mixing shape types

Each shape file can technically store a mix of different shape types, as the shape type precedes each record, but common use of the specification dictates that only shapes of a single type can be in a single file. For example, a shape file cannot contain both Polyline and Polygon data. Thus, well (point), river (polyline) and lake (polygon) data must be kept in three separate files.

Import Method

Choose the **File | Import** command.

LizardTech MrSID .SID File Description

The LizardTech MrSID filter imports .SID files.

File Description

The MrSID (pronounced Mister Sid) is an acronym that stands for *multiresolution seamless image database*. It is a file format developed and patented by LizardTech for encoding of georeferenced raster graphics. Unlike many other file formats, any rectangular region of the image can be quickly extracted from the file without decompressing the entire file. A region can be specified for import in the **MrSID Import Options** dialog, or the image can be imported as a read-only image. Each of these options decreases the import time for large MrSID files.

File Name Extensions

.SID

Import Method

Choose the **File | Import** command.

Import Options

Specify import options in the **SID Image Import Options** dialog.

Third Party Notice

Portions of this computer program are copyright (C) 1995-2008 Celartem, Inc., doing business as LizardTech. All rights reserved. MrSID is protected by U.S. Patent No. 5,710,835. Foreign Patents Pending.

Sylk Spreadsheet .SLK Files

The SYLK Spreadsheet filter imports SYLK format spreadsheet files as tabular data.

File Description

Symbolic Link SYLK is a Microsoft file format typically used to exchange data between applications, specifically spreadsheets. SYLK files conventionally have a .SLK suffix. From within a spreadsheet data can be exported in the SYLK format. Composed of only displayable ANSI characters, it can be easily created and processed by other applications, such as databases.

Microsoft does not publish a SYLK specification. Variants of the format are supported by Multiplan, Microsoft Excel, Microsoft Works, OpenOffice.org, and Gnumeric.

Note that even if a SYLK file is created by an application that supports Unicode (for example Microsoft Excel), the SYLK file will be encoded in the current system's ANSI code page, not in Unicode. If the application contained characters that were displayable in Unicode but have no codepoint in the current system's code page, they will be converted to question marks ('?') in the SYLK file.

File Format

As an example, the following SYLK code in a text file with the .SLK extension:

```
ID;P
C;Y1;X1;K"Row 1"
C;Y2;X1;K"Row 2"
C;Y3;X1;K"Total"
C;Y1;X2;K11
C;Y2;X2;K22
C;Y3;X2;K0;ER1C2+R2C2
E
```

would be displayed like this when read by an appropriate spreadsheet:

Row 1	11
Row 2	22
Total	33

File Name Extensions

.SLK

Format(s) Supported for Import

tabular data; double

Import Method

Choose the **File | Import** command.

Import Options

No import options dialog is displayed.

Metamorph .STK File Description

The Metamorph filter imports lattice data from Metamorph .STK data sets.

File Description

A Metamorph .STK data set contains one or more stacked bitmap images (slices). This file format has historically been used to record data from some makes of confocal scanning microscopes.

File Name Extensions

.STK

Format(s) Supported for Import

uniform lattice; 8-, 16-bit integer.

Import Method

Choose the **File | Import** command.

Truevision Targa .TGA File Description

The Truevision Targa filter imports and exports TrueVision Targa bitmap images.

File Description

TARGA is an acronym for Truevision Advanced Raster Graphics Adapter; TGA is an initialism for Truevision Graphics Adapter. Today, most people refer to the format as the "TARGA File Format".

File Format

Truevision's (now AVID) TGA file format, often referred to as TARGA file format, is a raster graphics file format. It was the native format of Truevision Inc.'s TARGA and VISTA boards, which were the first graphic cards for IBM-compatible PCs to support Highcolor/truecolor display. This family of graphic cards was intended for professional computer image synthesis and video editing with PCs; for this reason, usual resolutions of TGA image files match those of the NTSC and PAL video formats.

TGA files commonly have the extension .TGA on PC DOS/Windows systems. The format can store image data with 8, 16, 24, or 32 bits of precision per pixel, the maximum 24 bits of RGB and an extra 8-bit alpha channel. Color data can be color-mapped, or in direct color or truecolor format; optionally, a lossless PackBits RLE compression can be employed.

Uncompressed 24-bit TGA images are relatively simple compared to several other prominent 24-bit storage formats: A 24-bit TGA contains only an 18-byte header followed by the image data as packed RGB data. In contrast, BMP requires padding rows to 4-byte boundaries, TIFF and PNG are metadata containers that do not place the image data or attributes at a fixed location within the file.

File Name Extensions

.TGA

Format(s) Supported for Import

device-independent bitmap; 1, 4, 8, 24 bit per pixel

Format(s) Supported for Export

device-independent bitmap; 8, 24 bit per pixel

Import Method

Choose the **File | Import** command.

Export Method

Choose the **File | Export** command or select an image module and choose the **File | Save Data** command.

Export Options

Specify the export options in the **Export Options** dialog.

Tagged Image File Format .TIF, .TIF File Description

The Tagged Image filter imports and exports bitmap images and uniform lattices in Tagged Image File Format (TIFF). This implementation is based on the TIFF Specification version 6.

File Description

The tagged image file format .TIF or .TIFF is a file format for storing images, including photographs and line art. The TIFF format is widely supported by a variety of applications.

File Format

The TIFF specification allows an unusually wide variety of different formats and encodings within the same file format. While the filter can read most of the common variants of TIFF, it would be impractical to develop software to read every possible variant. The TIFF filter supports a wide variety of TIFF files that use the PlanarConfiguration 2 encodings.

TIFF images that contain bit per pixel counts other than 1, 4, 8, 16, 24, or 32 may not be readable depending on the encoding and alignment of the data.

TIFF images that are encoded with photometric interpretation models other than RGB, YCbCr, grayscale, or monochrome may not be readable. In particular, some YUV encodings cannot be imported.

Some of the compression algorithms allowable under the TIFF specification are not supported.

Voxler does support importing georeferenced TIF files with internal or external references. TIF images import into **Voxler** with units of image size and at the size specified in the .TIF file, when this information is stored in the file. Otherwise, images are displayed in pixels.

File Name Extensions

.TIF, .TIFF

Format(s) Supported for Import

- device-independent bitmap; 8, 24, 32 bit per pixel
- uniform lattice; 8-, 16-bit integer

Format(s) Supported for Export

- device-independent bitmap; 8, 24, 32 bit per pixel
- uniform lattice; 8-, 16-bit integer

Import Method

Choose the **File | Import** command.

Export Method

Choose the **File | Export** command or choose the **File | Save Data** command.

Export Options

Specify the export options in the **Export Options** dialog. Several options appear in the **TIFF Image Export Options** dialog.

Voxler Data .VDAT Files

The VDAT filter imports and exports **Voxler** data files.

File Description

Voxler data files can contain uniform lattices, rectilinear lattices, curvilinear lattices, or point data.

The VDAT filter maintains an exact representation of your data (no rounding off), so it is an ideal format to use.

File Name Extensions
.VDAT

Format(s) Supported for Import

- 2D uniform lattice; 8-, 16-, 32-bit integer, float, double
- 2D rectilinear lattice; 8-, 16-, 32-bit integer, float, double
- 2D curvilinear lattice; 8-, 16-, 32-bit integer, float, double
- 3D uniform lattice; 8-, 16-, 32-bit integer, float, double
- 3D rectilinear lattice; 8-, 16-, 32-bit integer, float, double
- 3D curvilinear lattice; 8-, 16-, 32-bit integer, float, double
- point data

Format(s) Supported for Export

n/a

Import Method

Choose the **File | Import** command.

Export Method

Select a module and choose the **File | Save Data** command.

Voxler Project .VOXB Files

Voxler project files contain modules, connections, and source data.

File Name Extensions
.VOXB

Format(s) Supported for Import

Project files with the .VOXB extension

Format(s) Supported for Export

Projects may be saved as .VOXB files

Import Method

Choose the **File | Open** command.

Export Method

Choose the **File | Save** command.

Visualization Toolkit .VTK Files

The Visual Tool Kit filter imports and exports Visualization Tool Kit .VTK data sets. This implementation is based on version 4.2 of the VTK File Format documentation from Kitware.

File Description

VTK data sets can contain several types of lattice data and/or geometric figures. The content of VTK files can be in ASCII text format or a mixed binary/ASCII format in which headers and parameters are in ASCII format but the data values are in binary format.

File Name Extensions

.VTK

Format(s) Supported for Import

- 3D uniform lattice; 8-, 16-, 32-bit integer, float, double
- 3D rectilinear lattice; 8-,16-, 32-bit integer, float, double
- 3D curvilinear lattice; 8-, 16-, 32-bit integer, float, double
- point set; double

Format(s) Supported for Export

- 3D uniform lattice; 8-, 16-, 32-bit integer, float, double
- 3D rectilinear lattice; 8-, 16-, 32-bit integer, float, double
- 3D curvilinear lattice; 8-, 16-, 32-bit integer, float, double

Import Method

Choose the **File | Import** command.

Export Method

Choose the **File | Save Data** command.

Export Options

See **Visual Tool Kit .VTK Export Options Dialog**. See Export Blanking Options.

Import Restrictions/Limitations

The implementation of the software does not directly support geometric figures in VTK data sets. Geometric figures are currently imported as **Voxler** point sets, with one point at each vertex in the geometric model.

Export Restrictions/Limitations

This implementation does not support exporting of geometric figures; only lattices can be exported as .VTK files.

AVSX .X, .XIMG Files

The AVSX filter imports and exports 32-bit per pixel color bitmap images from AVS X-Image .X and .XIMG files.

File Description

The AVS X-Image format originated on UNIX workstations and is typically used to store true-color images containing an alpha channel, a feature that most other image file formats lacked at the time this file format was developed.

File Format

An AVS X-Image file is a binary file containing a raster image with 8-bits each for the red, green, blue, and alpha channels (32 bits total per pixel).

File Name Extensions

.X, .XIMG

Format(s) Supported for Import

device-independent bitmap; 32 bit per pixel RGBA

Format(s) Supported for Export

device-independent bitmap; 32 bit per pixel RGBA

Import Method

Choose the **File | Import** command.

Export Method

Choose the **File | Export** command or select an image module and choose the **File | Save Data** command.

Excel Spreadsheet .XLS, .XLSM, and .XLSX Files

The XLS filter imports and exports tabular data from Microsoft Excel spreadsheet files.

File Description

Unlike the other spreadsheet file formats supported by **Voxler**, the Excel file format allows multiple pages or "worksheets" in a single Excel file. When you import an Excel file containing multiple worksheets, you are given the option to select which worksheet(s) to import.

File Name Extensions

.XLS, .XLSM, .XLSX

Format(s) Supported for Import

tabular data; double

Format(s) Supported for Export

tabular data; double

Import Method

Choose the **File | Import** command to import the file into a *Data Source* module. Choose the **File | Open** command to open the file in the worksheet.

Export Method

Click the **File | Save As** command in the worksheet. Alternatively, select a data module in the **Network Manager** and choose the **File | Save Data** command. Both .XLS and .XLSX files can be saved from **Voxler**. After typing a name for the .XLS format, the **XLS Export Options** dialog is displayed.

Other Notes

Excel .XLS file formats have a 65,536 row limit and a 256 column limit. Excel .XLSX file formats have a 1,048,576 row limit and a 16,384 column limit. Font color is maintained when opening XLSX and XLSM files in the worksheet. However XLS files are opened with black font, regardless of font color in the XLS file.

XYZ Grid .DAT File Description

The XYZ Grid filter exports to the XYZ grid file format.

File Description

The XYZ file is a simple encoding of 3D points into a text file. Each line of the file contains three numbers. Each triplet of three numbers comprises the X, Y, and Z coordinates, respectively, of a point in 3D space.

File Name Extensions

.DAT

Export Method

Select an image module and choose the **File | Save Data** command.

Z-MAP Plus Grid .ASC, .DAT, .GRD, .XYZ, .ZMAP, .ZYC, .ZYCOR File Description

The Z-MAP Plus Grid filter imports and exports Z-Map Plus ASCII grid files.

File Description

Z-Map Plus grid files are used by Geographix applications such as IsoMap to store gridded elevation data.

File Format

The data are written a column at a time from the left side to the right side of the grid.

File Name Extensions

The format has several file extensions: .ASC, .DAT, .GRD, .XYZ, .ZMAP, .ZYC, .ZYCOR

Import Method

Choose the **File | Import** command.

Import Options

Specify the import options in the **Lattice Import Options** dialog.

Import Restrictions/Limitations

This import filter only supports the GRID type of the Z-Map Plus formats.

Export Method

Select a lattice or image module and choose the **File | Save Data** command.

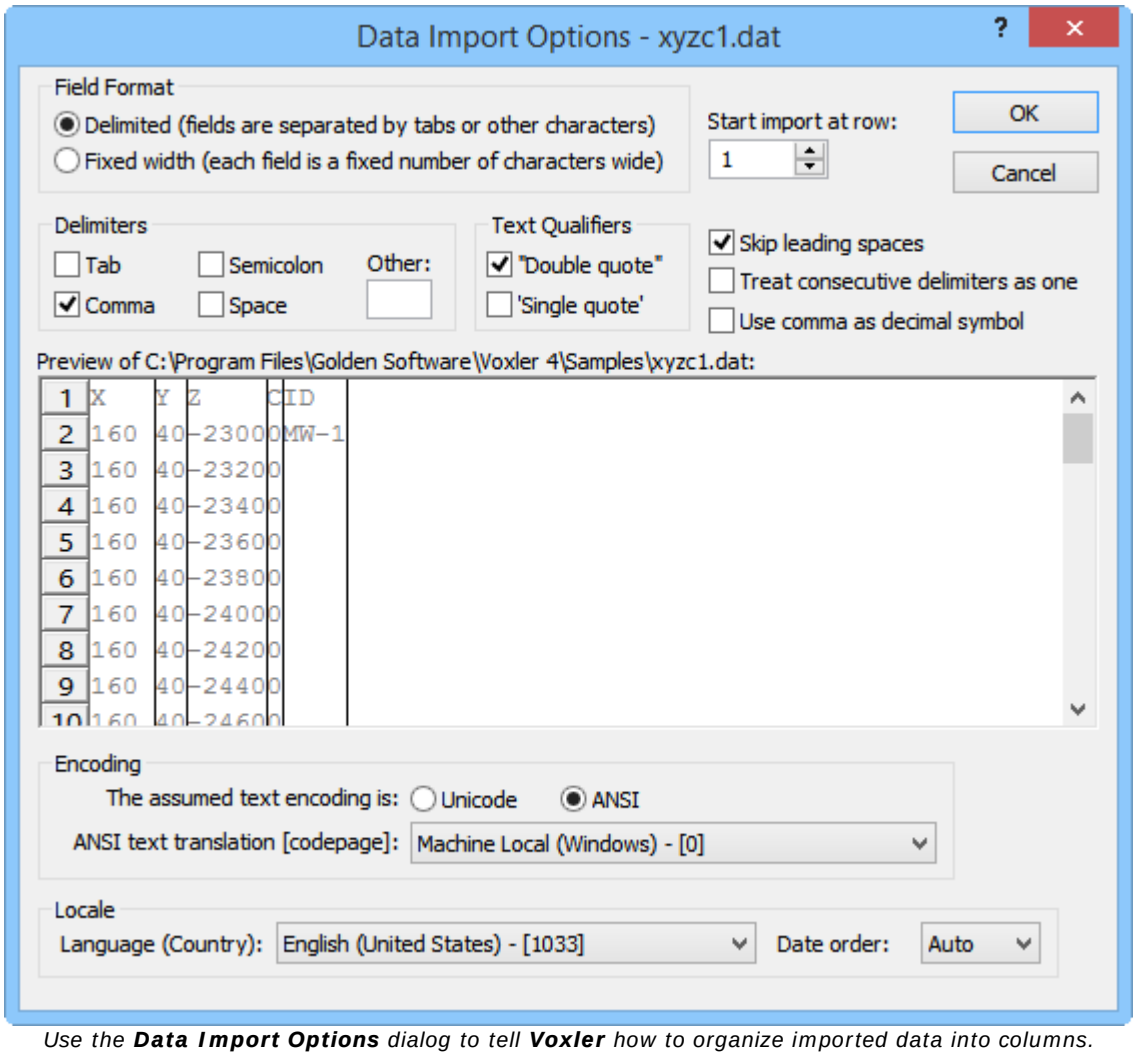
Import Options Dialogs

Data Import Options Dialog

If a file is in an ASCII text format with an unrecognized file extension, the **Data Import Options** dialog appears when opening the file.

Data Import Options Dialog

The **Data Import Options** dialog is used to organize tabular data into point databases organized by columns. The dialog appears when the **File | Import** command is used when importing tabular data from delimited text files (i.e. .TXT). The **Data Import Options** dialog is also displayed when using the worksheet **Import** command and when the *Show Import Options* box is checked in the **Paste Special** dialog. These file formats are assumed to have one record per line in which each record contains a fixed number of numeric data fields.



*Use the **Data Import Options** dialog to tell **Voxler** how to organize imported data into columns.*

Field Format

Specify the format of the input fields in the *Field Format* group. The *Field Format* group controls allow you to specify whether the fields in each record are of fixed width or are separated by delimiters. The options are *Delimited* or *Fixed Width*.

Delimited

Choose *Delimited (fields are separated by tabs or other characters)* if the imported data uses *Delimiters* (tab, semicolon, comma, space, other) to separate data fields.

Fixed Width

Choose *Fixed width (each field is a fixed number of characters wide)* if the imported data uses a fixed width to separate data fields.

Start Import at Row

Use the arrow buttons or type in a row number at which to start the data import in the *Start import at row* box. For example, a value of one will start the data import at row one of the data set. A value of five will start the data import at row five of the data set.

Delimiters

The *Delimiters* group is used to specify how the fields are separated if *Delimited (fields are separated by tabs or other characters)* is the selected *Field Format*. *Delimiters* controls allow you to specify what characters act as delimiters between fields in a record. Choose the desired delimiters to be used during the import process by checking the box next to *Tab*, *Comma*, *Semicolon*, or *Space*. You may also enter a custom delimiter in the *Other* box. More than one delimiter may be checked.

Text Qualifiers

Specify *"Double Quote"* or '*Single Quote'* in the *Text Qualifiers* group to indicate the correct qualifier to identify text values in the data file.

Double Quote

Check the box next to *"Double Quote"* or *'Single Quote'* to specify that everything between those marks should be interpreted as a single value, and any delimiter characters between any two quote characters are not treated as a delimiter.

For example, if *Space* is chosen as the delimiter and *Double Quote* is chosen as the text qualifier, the string "Aspen Park" is treated as a single data value due to the double quotes surrounding it, and the space delimiter between the words is treated as part of the value.

Single Quote

Check the box next to *'Single Quote'* to specify that everything between those marks should be interpreted as a single value, and any delimiter characters between any two quote characters are not treated as a delimiter.

For example, if *Space* is chosen as the delimiter and *Single Quote* is chosen as the text qualifier, the string 'Aspen Park' is treated as a single data value due to the single quotes surrounding it, and the space delimiter between the words is treated as part of the value.

Skip Leading Spaces

Check the box next to *Skip leading spaces* to tell the software to ignore spaces that appear before initial text (not treat spaces at the beginning of the line as delimiters).

Treat Consecutive Delimiters as One

Check the box next to *Treat consecutive delimiters as one* to instruct the software to interpret any consecutive delimiters into a single delimiter rather than breaking to a new column for each consecutive delimiter.

Check the *Treat consecutive delimiters as one* box to treat any sequence of more than one consecutive delimiter as if it were one delimiter. If this box is unchecked, each delimiter character marks a new field, i.e., consecutive delimiters act to define "empty" fields.

Use Comma as Decimal Symbol

Check the box next to *Use comma as decimal symbol* if commas are used to as the divisor symbol between whole numbers and fractions. If your data uses commas as the decimal symbol, it is highly recommend that you use some other character for the *Delimiters*. This option is more common in locations where the thousands separator is the period (.) and the comma is used as the decimal.

Preview

The parsed data are shown in the *Preview* section. The *Preview* field displays a preview of how the text in the import field is divided into fields to help you set the other dialog controls appropriately.

Encoding

The *Encoding* section allows the choice of *Unicode* data or *ANSI* data when importing or opening an ASCII data file. Unicode data is often referred to as international data. It would include character sets from Russia, Israel, China, Greece, Hungary, among others. After selecting *Unicode*, select the *ANSI text translation [codepage]* option that will read the data correctly. If the data does not appear correctly in the *Preview* window, the *Encoding* may be specified incorrectly.

ANSI encoding contains characters within the first 256 characters of a font. These are normally in English.

Locale

The locale section of the dialog contains options for determining date/time values. The *Language (Country)* setting determines which month names are interpreted as part of a date. For example, if *German (Germany)* is selected, "Oktober" will be recognized as a valid month name. English month names are always recognized as valid month names. The default *Language (Country)* is determined by the user locale set in the Windows Control Panel. To change the *Language (Country)*, click the current option and select a language from the list.

The *Date order* option specifies the order in which dates are written in the data file. The date 02/03/04 is ambiguous and could be Month-Day-Year, Day-Month-Year, Year-Month-Day, etc. The *Date order* option ensures dates in the data file are imported correctly into **Voxler**. The default *Date order* is *Auto*. The standard date order for the *Language (Country)* setting is used when *Date order* is set to *Auto*. Specify the *Date order* for the data file import by clicking the current *Date order* selection and then selecting the desired *Date order* from the list. All six combinations of Day (D), Month (M), and Year (Y) are included in the *Date order* list.

OK

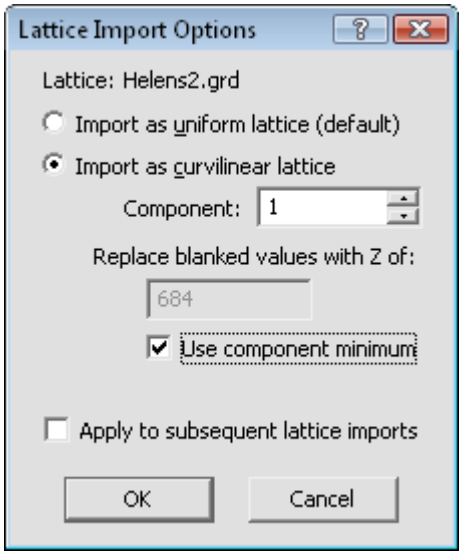
Click the *OK* button to proceed with the import process.

Cancel

Click the *Cancel* button to close the dialog without importing the data set.

Lattice Import Options Dialog

Load a lattice into **Voxler** using the **File | Import** command and the **Lattice Import Options** dialog is displayed.



Specify import options in the ***Lattice Import Options*** *dialog.*

Lattice Name

The lattice file being imported into **Voxler** is displayed next to *Lattice*. The displayed lattice name is read-only. The supported file formats that display this dialog include:

- ADF Arc/Info Binary Grid
- BIL Band Interleaved .BIL, .BIP, .BSQ
- CPS-3 Grid .ADX, .DAT, .GRD, .CPS
- DDF SDTS DEM .DDF
- DEM USGS Digital Elevation Model .DEM
- DOS USGS ETOPO5 .DOS, .DAT
- ERS ER Mapper Grid .ERS
- FLT Esri Binary Float Grid .FLT, .HDR
- GRD Surfer Grid .GRD
- GXF Grid Exchange .GXF
- HDF Hierarchical Data Format .HDF
- HGT NASA SRTM Data Format .HGT
- IMG ERDAS Imagine File Format .IMG
- RAW Binary Lattice .RAW, .BIN
- RST Idrisi Raster Image .RST, .IMG
- SEG-Y Seismic Data Log .SGY, .SEGY
- Z-MAP Plus Grid .ASC, .DAT, .GRD, .XYZ, .ZMAP, .ZYC, .ZYCOR

Import as Uniform Lattice

Select *Import as uniform lattice (default)* to import the grid file with the Z values as zeros for the entire grid. The grid values are imported as component information. This method imports all Z values data as zero. This means that some graphical output modules cannot be attached to the grid.

Import as Curvilinear Lattice

Select *Import as curvilinear lattice* to import the grid with a Z value for each grid node. Specify the Z value using the *Component* option. This method imports the component information as the Z data. For blanked values, a value can be specified or the component minimum value can be used.

Component

When the *Import as curvilinear lattice* is selected, specify the Z value for each grid node in the *Component* box. To change the component, highlight the existing value and type a new value in the box or click the button to increase or decrease the *Component* value. The number is controlled by the number of components in the grid file.

Blanked Z Values

When the original grid has a blank value, the program must use a new number in the curvilinear lattice. The *Replace blanked values with Z of* can be specified for the blanking value. To set the blank value to any numeric value, highlight the existing number and type a new value.

Use Component Minimum

Check the box next to the *Use component minimum* option to use the minimum component value when a blank value in the grid needs to be filled. This option is useful when using the *Transform* module on a *HeightField*. For example, when a grid file has X and Y scales that differ from the Z scale and a *Transform* module is added between the data and the *HeightField*.

Apply to Subsequent Uniform Lattices

Check the *Apply to subsequent lattice imports* box to apply the selections to future imported lattices.

OK or Cancel

Click the *OK* button to close the dialog and import the lattice file using the specified options. Click the *Cancel* button to close the dialog without saving changes.

Voxler Warning

If you load a file with multiple 2D slices, choose to import as a uniform lattice, and the *Apply to subsequent lattice imports* box is checked, a warning is issued. The warning is a reminder that **Voxler** is ignoring a lattice that contains no data and any subsequent objects that have no data will also be ignored. Click *OK* to accept the warning and continue.



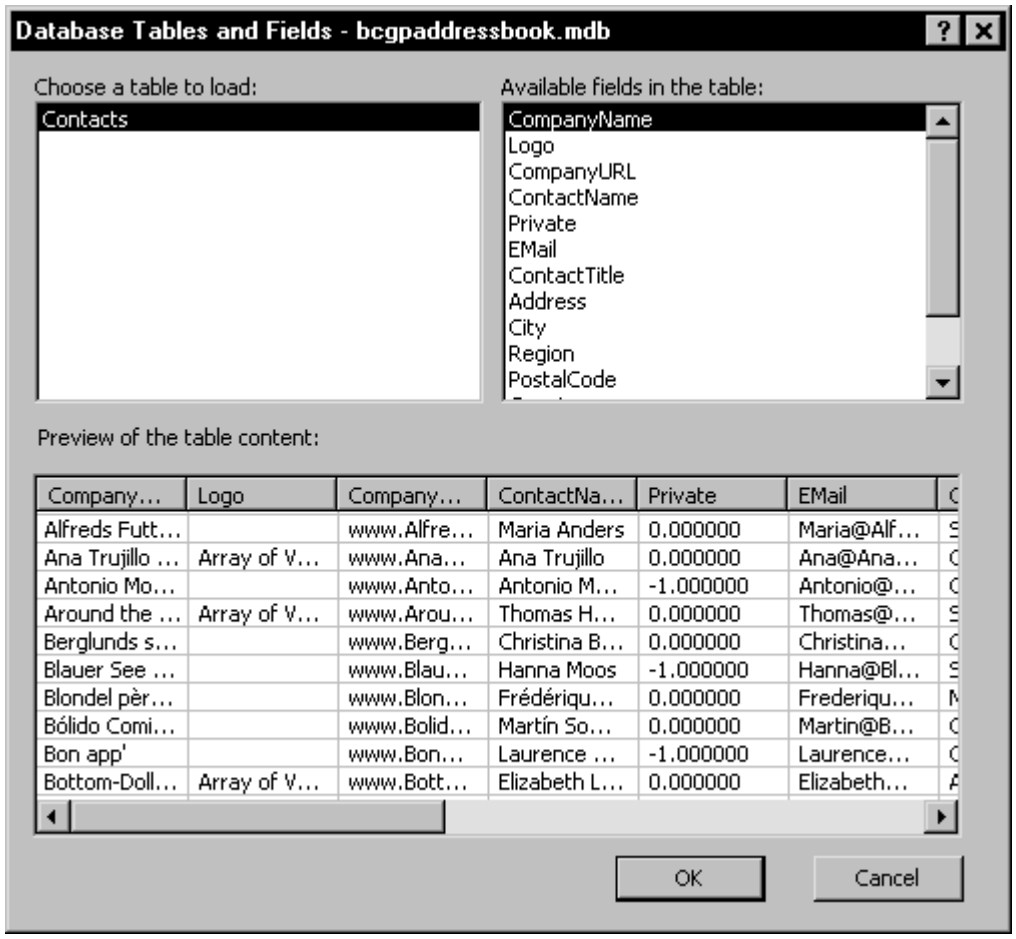
A warning message is issued to confirm that subsequent objects with no data will also be ignored.

Microsoft Access .ACCDB and .MDB Import Options Dialog

Microsoft Access .ACCDB and .MDB is a binary database file format. The .MDB file format was used by pre-2007 versions of Microsoft Access. The .ACCDB file format is used by Microsoft Access 2007 and 2010.

Database Tables and Fields Dialog

Click the **File | Import** command to load a .MDB file. The **Database Tables and Fields** dialog allows you to choose what table to load and preview the data that will be imported.



*Specify .MBD import options in the **Database Tables and Fields** dialog.*

Choose a Table to Load

If the .MDB file contains multiple tables, you can select which table to load in the *Choose Table To Load* list.

Available Fields in the Table

The *Available fields in the table* displays all of the available fields in the table.

Preview of the Table Content

A preview of the selected table content is shown in the *Preview of the table content* section.

OK

Click the *OK* button to proceed with the import process.

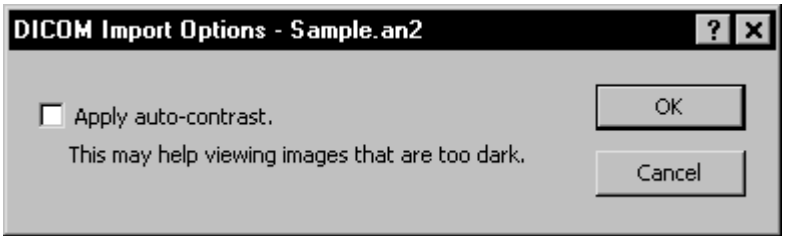
Cancel

Click the *Cancel* button to close the dialog without importing.

DICOM Import Options Dialog

The DICOM Import Options Dialog

Import a ACR-NEMA Medical Image file or a DICOM Medical Image file with the **File | Import** command and the **DICOM Import Options** dialog opens.



*Customize import options in the **DICOM Import Options** dialog.*

Apply Auto-Contrast

Check the *Apply auto-contrast* box to expand the dynamic range of the import data to fit the maximum extents of the import data type. This can improve visibility (contrast) on images recorded with low dynamic range.

OK

Click the *OK* button to proceed with the import process.

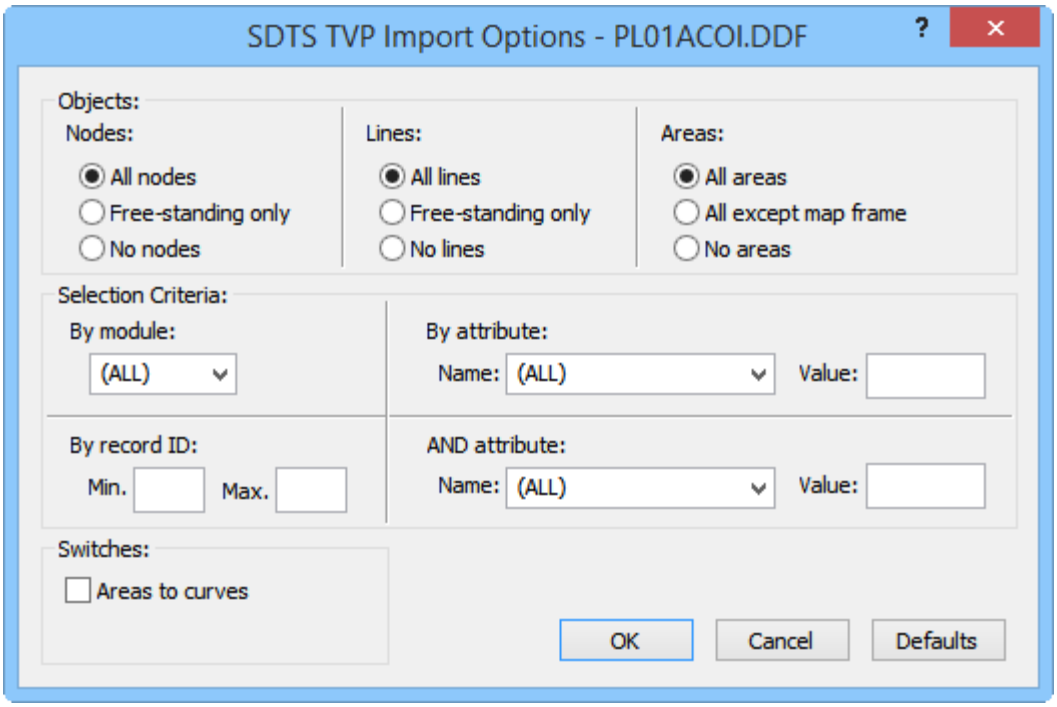
Cancel

Click the *Cancel* button to close the dialog without importing.

SDTS Topological Vector Profile .TVP Import Options Dialog

SDTS TVP Import Options Dialog

Select any .DDF file to open the **Import Options** dialog.



*Customize the import options in the **SDTS TVP Import Options** dialog.*

Nodes Options

These options determine which Node items are imported. If *All nodes* is selected, all nodes in the SDTS data set are imported. If *Free-standing only* is selected, only those nodes that are not associated with an area or line are imported. If *No nodes* is selected, none of the nodes are imported.

Lines Options

These options determine which Line items are imported. If *All lines* is selected, all lines in the SDTS data set are imported. If *Free-standing only* is selected, only those lines that are not associated with an area are imported. If *No lines* is selected, none of the lines are imported.

Areas Options

These options determine which Area items are imported. If *All areas* is selected, all areas in the SDTS data set are imported. If *All except map frame* is selected, the areas that make up the outline of the map will not be imported. If *No areas* is selected, none of the areas are imported.

Some SDTS data sets supplied by USGS contain a map frame that is expressed in the data set as a normal polygon instead of an "invisible" polygon (i.e. a "PC" entity instead of a "PW" or "PX" entity in SDTS terminology). In such cases, the *All Except Map Frame* control will have no effect.

By Module List Box

If the name of a specific module is selected, imported items will be limited to those that reside in the specified module. If *(ALL)* is selected, items will be imported from all modules in the SDTS data set.

By Record ID Edit Boxes

Each item in the SDTS data set has a unique record ID number. To import only those items within a specific range of record IDs, enter the lowest desired record ID number in the *Min* edit box, and the highest desired record ID number in the *Max* edit box.

By Attribute, AND Attribute Controls

Application-specific attributes are associated with some items in an SDTS data set. To import only those items that have a specific attribute, select the name of the desired attribute in the *Name* list box and enter the value of the desired attribute in the *Value* edit box. If *(ALL)* is selected in the list box, items will be imported without regard to attributes. If both *By Attribute* and *AND Attribute* are specified, only those items that have both of the specified attributes will be imported.

Areas to curves Check Box

If the *Areas to curves* box is checked, any areas in the data set will instead be imported as lines (curves) instead of polygonal areas.

Defaults Button

The *Defaults* button resets the Import Options to default values. The default options direct the Import Filter as follows: Import free-standing nodes, import free-standing lines, import all areas, don't limit by module, record ID, or attribute, don't import areas as curves, and don't synthesize Ids.

SDTS Topological Vector Profile Import Filter Messages

The following messages may appear while importing SDTS Topological Vector Profile files.

Expected SDTS TVP data. Found SDTS DEM or raster

This error message:

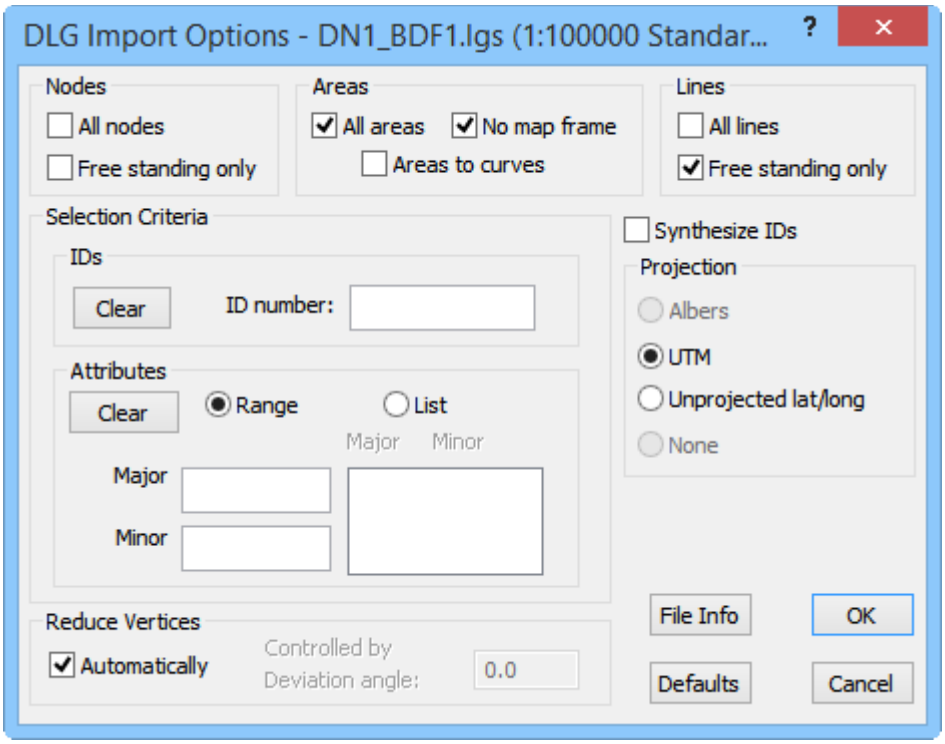
Expected SDTS Topological Vector Profile (TVP) format. Dataset appears to contain SDTS Raster (grid) Profile.

may appear if the selected .DDF file is in the incorrect format. If you receive this message after choosing **Map | New | Base Map**, you likely have a SDTS DEM file. Try using the **Map | New | Contour Map** command instead.

USGS Digital Line Graph .DLG, .LGO, .LGS Import Options Dialog

DLG Import Options Dialog

Choose the **File | Import** command and select a *DLG/LGO/LGS USGS Digital Line Graph (*.dlg)* file to open the **DLG Import Opitions** dialog.



*Customize import options in the **DLG Import Options** dialog.*

Nodes, Areas, Lines

Nodes - Most commonly, the point where two (or more) lines meet. Sometimes there are "free-standing" nodes, which are isolated points.

Areas - A closed, bounded region whose interior may be filled with a color or pattern.

Lines - A series of connected points which always begin at a node and end at a node.

Nodes

If *All Nodes* is checked, consider all nodes. If they meet the selection criteria, they will be passed to the application as points.

If *Free standing only* is checked, consider only free-standing nodes. If they meet the selection criteria, they will be passed to the application as points. The only free-standing nodes are the map reference points.

Areas

If *All areas* is checked, consider all area items. If they meet the selection criteria, they will be passed to the application.

If *No map frame* is checked, the area that makes up the map frame won't be imported.

If *Areas to curves* is checked, any selected areas passed to the application will be passed as a series of "curve" objects (lines), as opposed to the usual "area" objects (polygons).

Lines

If *All lines* is checked, consider all line items. If they meet the selection criteria, they will be passed to the application as "curve" objects.

If *Free standing only* is checked, consider only free-standing line segments. This is the normal case, since one usually wants line segments that are area boundaries to be passed to the application as part of one or more "area" objects.

Selection Criteria

The *Selection Criteria* edit boxes allow you to specify a value (like 140) or range of values (like 6001-6009 inclusive) which limit the items considered. Leave an edit box empty to place no limit.

ID Number

Only items with the specified ID or IDs within the specified range will be passed back to the application. This is useful for importing a single item (or group of items) of one type. For example, to import just Area 100, make sure no Nodes or Lines boxes are checked, the All Areas box is checked and 100 is entered in the ID Number edit box. The Import Filter considers no nodes or lines and, within areas, only those with ID 100, and passes back one area, Area 100.

Major and Minor Attribute

Range

If the *Range* radio button in the *Attributes* group is turned on, the edit controls for the Major and Minor attributes are enabled. Only items that have at least one attribute code whose Major attribute falls within the specified range and whose corresponding Minor attribute falls within its specified range will be passed to the application. For example, to import reservoirs only, open the appropriate water bodies file (S??_WB.LGO), check only All Areas, enter nothing in the ID Number edit box (means any ID), turn on the Range radio button and enter 40 in the Major Attribute edit box (Water Bodies) and 106 (Reservoir) in the **Minor Attribute** edit box.

List

If the *List* radio button in the *Attributes* group is turned on, the list box is enabled. Only items that have at least one attribute code whose *Major* and *Minor* attributes match one of the attribute pairs in the list box will be passed to the application. To add attribute pairs into the list box, click *File Info* button to bring up the **File Info** dialog, and double-click the attribute pairs in the Attributes group in the **File Info** dialog. To remove an item from the list box select that item and then click the *Clear* button. You may remove multiple items at one time by selecting multiple items in the list box.

Projections

The *Projection* radio buttons control how coordinates are returned to the application. DLG files have coordinates calculated using a Universal Transverse Mercator (UTM) or Albers Equal Area Ellipsoid projection. UTM is used in 1:24,000-scale maps and 1:100,000-scale maps, while Albers is used in 1:2,000.000-scale maps.

UTM

The native file coordinates of 1:24,000-scale maps or 1:100,000-scale maps are returned to the application, but the application is also given the parameters used in the UTM projection. Use of this option makes sense only if the application understands how to handle a UTM projection.

Albers

The native file coordinates of 1:2,000,000-scale maps are returned to the application, but the application is also given the parameters used in the Albers projection. Use of this option makes sense only if the application understands how to handle an Albers projection.

Unprojected Lat/Long

The file coordinates are converted from their native form to Lat/Long and the application is informed that it is receiving Lat/Long coordinates. This can lengthen the import time considerably, since substantial computation is involved.

None

The native file coordinates are returned to the application, but the application is told that the projection is unknown.

Reduce Vertices

DLG files typically have many vertices in each line segment, often more than are needed for many tasks. The DLG Import Filter provides two methods for reducing the number of vertices.

Automatically

When you check *Automatically*, the import filter applies an algorithm which requires no further input. This algorithm reduces the number of vertices on most DLG line segments by about 50-80%.

Use Deviation Angle

Achieve finer control over vertex reduction by unchecking the *Automatically* box and entering a Deviation Angle (in degrees) into the *Controlled by Deviation angle* field. Use small angles (5-10 degrees) to eliminate a few points, somewhat larger angles (15-25 degrees) to eliminate more points and use larger angles (30-60 degrees) to eliminate the greatest possible number of points.

No Text ID

USGS DLG files have no text ID items associated with Nodes, Areas or Lines. The line segments that make up the Pecos River, for example, are stored as free-standing line segments with river attribute codes, but there is no way to distinguish those line segments from any other line segments with river attributes. Sometimes, it is useful for investigation purposes to know the numeric ID of imported items. When the Synthesize IDs box is checked, the Import Filter synthesizes a Primary ID for each item using the item's type and numeric ID value. (Examples are "N14" for Node 14, "A237" for Area 237 and "L1067" for Line 1067.)

Defaults

The *Defaults* button resets the Import Options to default values. The default options direct the DLG Import Filter as follows: ignore all Nodes, consider all Areas (selected areas are returned as polygons), consider free-standing lines, place no limiting selection criteria, use UTM or Albers projection and do not synthesize text IDs.

File Info

When the *File Info* is clicked, the **File Info** dialog appears. The base file information is displayed, such as *Image Extents*, *Object IDs*, and *Major/Minor Attributes*.

OK

Click the *OK* button to proceed with the import process.

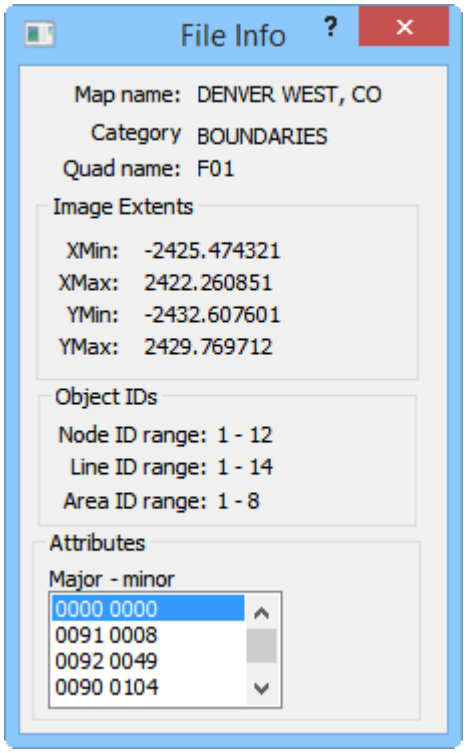
Cancel

Click the *Cancel* button to close the dialog without importing.

USGS Digital Line Graph .DLG, .LGO, .LGS File Info Dialog

File Info Dialog

Click the *File Info* button in the **DLG Import Options** dialog and the **File Info** dialog opens.



*The **File Info** dialog displays information about the DLG file.*

Display Information

The *Map name*, *Category*, and *Quad name* are displayed at the top of the dialog. The base file information is display only.

Image Extents

The *XMin*, *XMax*, *YMin*, and *YMax* values for the file are displayed.

Object IDs

The Node ID range, Line ID range, and Area ID range are displayed.

Attributes

The *Major - minor* attributes are displayed. Use the scroll bar to display additional attributes if necessary.

OK

Click the *OK* button to proceed with the import process.

Cancel

Click the *Cancel* button to close the dialog without importing.

AutoCAD DXF Import Options Dialog

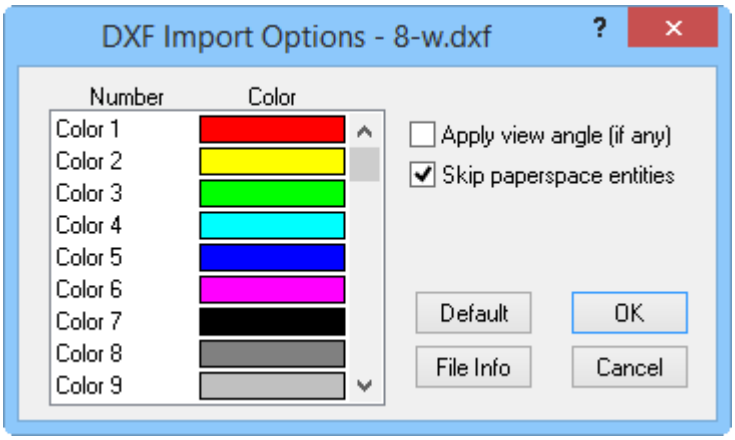
AutoCAD-compatible Drawing Exchange Format .DXF files contain information describing graphical objects, such as areas, curves, points and text. The DXF Import filter reads DXF files and structures the information in a form usable by the application.

Import Restrictions/Limitations

The fill and text in a .DXF will not appear when loaded into Voxler. Varying line widths are not imported into Voxler. Lines in the DXF file are displayed as the same width in Voxler. 3DSOLID is a proprietary format, and 3DSOLID entities are not supported in Voxler. If the .DXF file contains 3DSOLID objects, the 3DSOLID objects will be omitted during the import process.

DXF Import Options Dialog

Choose the **File | Import** command to load a .DXF file. The **DXF Import Options** dialog allows you to specify options which determine how information in the file is imported.



*Customize import options in the **DXF Import Options** dialog.*

Color Number

DXF files contain no direct color information, but use color numbers (1-255) instead. There is an adhoc standard association of colors with the first 7 color numbers: Red, Yellow, Green, Cyan, Blue, Magenta and Black. By double-clicking on items in the COLOR list box, you can change the color associated with a specific color number.

Default

Pressing the *Default* button will assign a default set of colors to each color number.

Apply View Angle

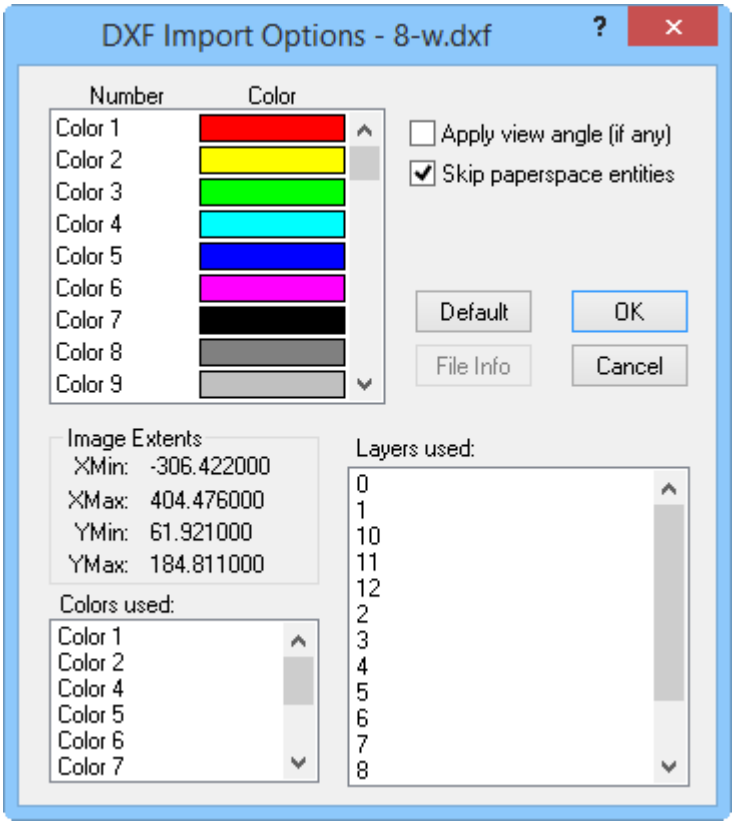
If any viewing angles have been applied in AutoCAD, check *Apply view angle (if any)* to preserve these settings. The unrotated coordinates will not be preserved if this box is checked.

Skip Paperspace Entities

To import only graphical entities from AutoCAD's 'modelspace' and skip importing entities from 'paperspace', check the *Skip paperspace entities* option. If this option is not selected, entities from both 'paperspace' and 'modelspace' are imported.

File Info

Click the *File Info* button to expand the **DXF Import Options** dialog and display file information concerning the *Image Extents*, *Colors used*, and *Layers used*.



Click the File Info button to expand the **DXF Import Options** dialog to display additional information.

Image Extents

Click the *File Info* button to expand the **DXF Import Options** dialog. The *Image Extents* section displays the *XMin*, *XMax*, *YMin*, and *YMax* values for the file being imported.

Colors Used

Click the *File Info* button to expand the **DXF Import Options** dialog. Selecting a color number displayed in the *Colors used* list box automatically selects that color number in the *Color* list box.

Layers Used

Click the *File Info* button to expand the **DXF Import Options** dialog. Double-clicking on a layer displayed in the *Layers used* list box displays the **Layer Name** dialog, showing the graphical entities present in the layer and a check box showing whether the layer is marked frozen (invisible) or not.

AutoCAD Entities

The point, line, polygon, and 3D AutoCAD entities are currently supported.

OK

Click the *OK* button to proceed with the import process.

Cancel

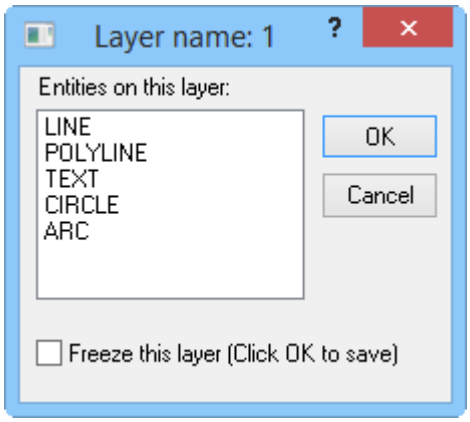
Click the *Cancel* button to close the dialog without importing.

AutoCAD DXF Import Options - Layer Name Dialog

The **Layer Name** dialog shows the graphical entities present in the layer and a check box showing whether the layer is marked frozen (invisible) or not.

Layer Name Dialog

To open the **Layer Name** dialog, first open the **DXF Import Options** dialog and click the *File Info* button to display the *Layers Used* section. Double-click on a layer displayed in the *Layers Used* list to display the **Layer Name** dialog.



Specify the layer name for the DXF import in the Layer Name dialog.

Entities on this Layer

Select a option from the *Entities on this layer* box. Left-click to select an option. The selected option will be highlighted. Only one entity can be selected at once.

Freeze this Layer

Check the *Freeze this layer (Click OK to save)* option to freeze/unfreeze layers prior to importing. Objects in layers that are marked frozen will not be passed on to the application.

OK

Click the *OK* button to proceed with the import process.

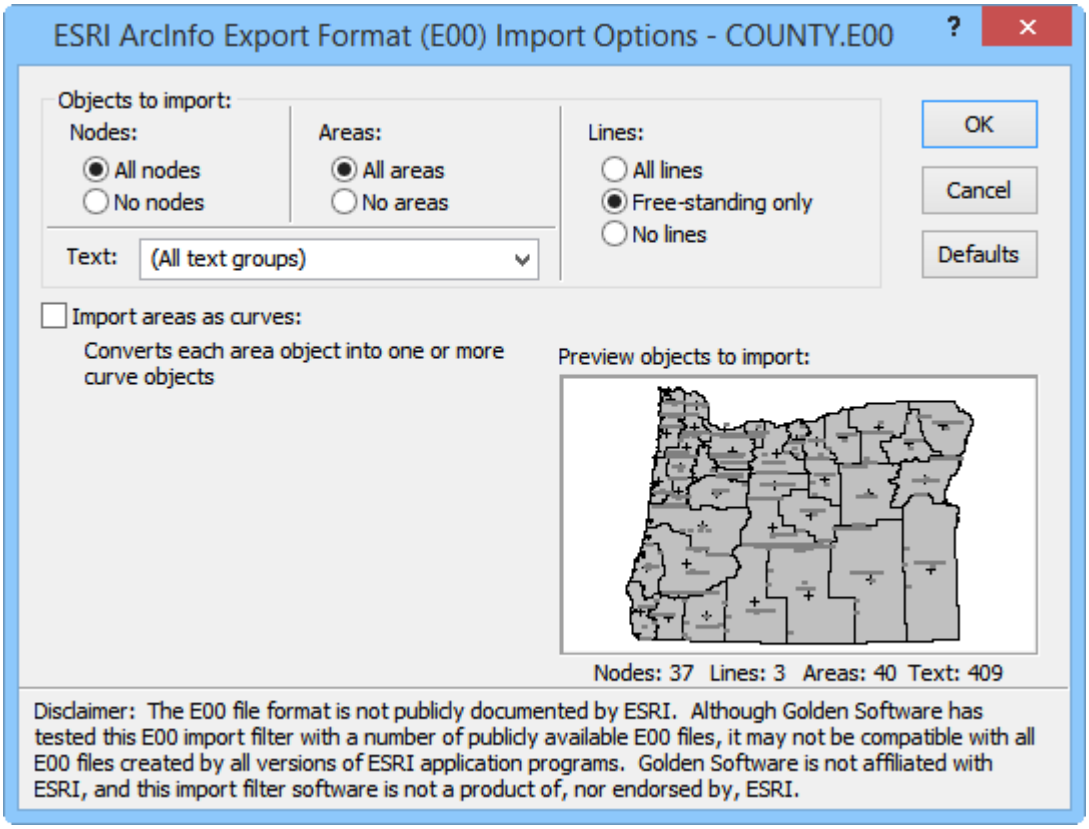
Cancel

Click the *Cancel* button to close the dialog without importing.

Esri ArcInfo Export Format .E00 Import Options Dialog

Esri ArcInfo Export Format (E00) Import Options Dialog

Choose the **File | Import** command to load an .E00 file. The **Esri ArcInfo Export Format (E00) Import Options** dialog allows you to specify options which determine how information in the file is imported. Each of the controls in this dialog is described below.



*Select the E00 import options in the **Import Options** dialog.*

Nodes

These radio buttons determine which *Node* items are imported.

- If *All nodes* is selected, all nodes in the drawing are imported.
- If *No nodes* is selected, none of the nodes are imported.

Areas

These radio buttons determine which *Area* items are imported.

- If *All areas* is selected, all areas in the file are imported.
- If *No areas* is selected, none of the areas are imported.

Lines

These radio buttons determine which *Line* items are imported.

- If *All lines* is selected, all lines in the file are imported.
- If *Free-standing only* is selected, only those lines that are not associated with an area are imported.
- If *No lines* is selected, none of the lines are imported.

Text List

The selection in this list determines which groups of text items are imported.

- If *All text groups* is selected, all text items from the import file are imported.
- If No text is selected, no text items from the import file are imported.
- If the name of a specific group is selected in the list, only the text items from that group are imported. Some import files don't contain any named groups of text items, in which case the only selections possible will be *All text groups* or *No text*.

Areas to Curves Check Box

Check the *Import areas as curves* option to convert each area object into one or more curve (line) objects.

Primary ID List

Select the attribute field that is to be assigned to the primary identifier of each imported object.

Secondary ID List

Select the attribute field that is to be assigned to the secondary identifier of each imported object.

Preview Objects To Import Display

This area of the dialog displays a rough preview of the items that are selected for import from the import file. Any lines selected for import are displayed in black. Any areas selected for import are displayed in light gray with a black border. Any nodes selected for import are displayed as black crosses. Any text items selected for import are displayed as dark gray rectangles. Any changes to the dialog controls that effect which objects are selected for import will be reflected in the preview display.

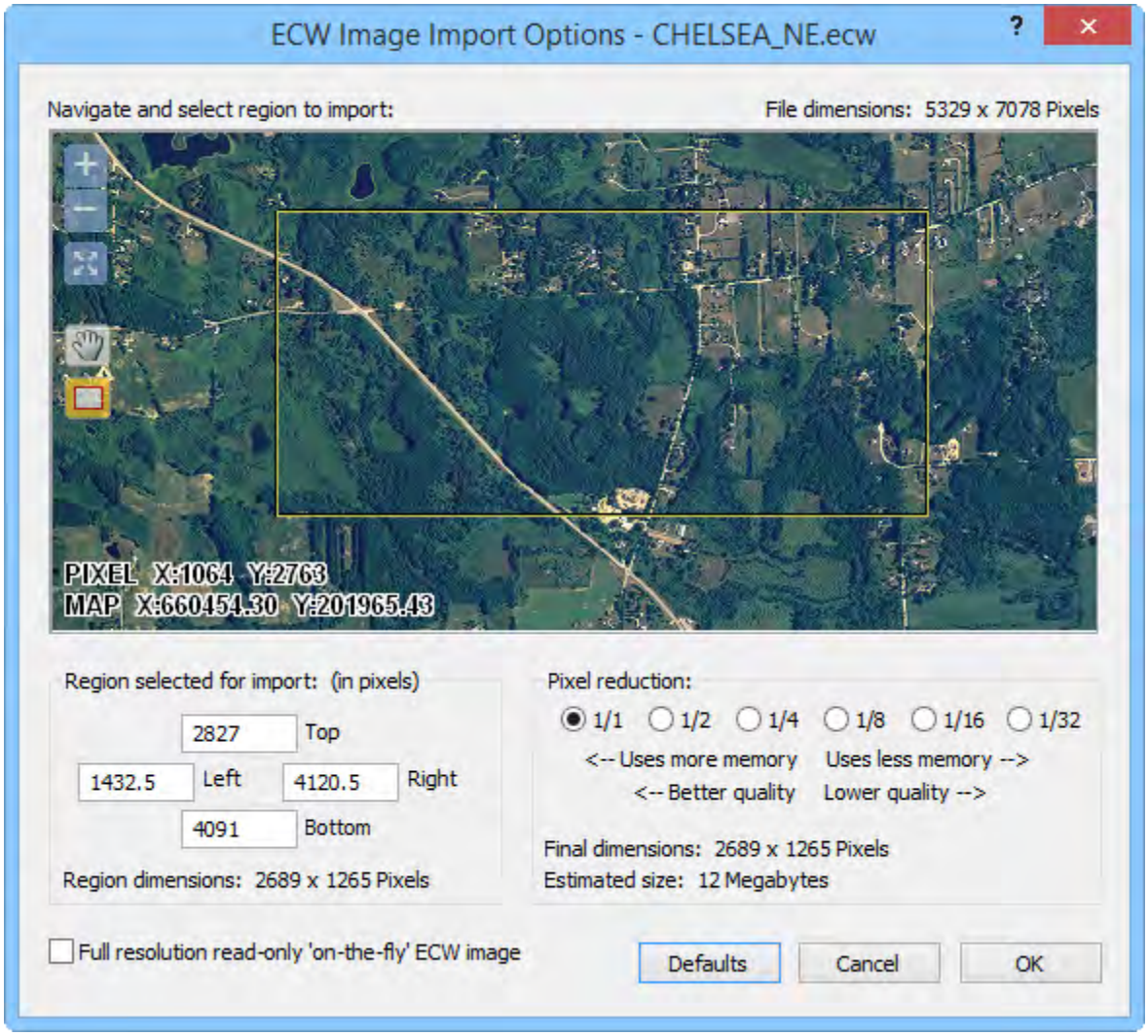
OK

Click the *OK* button to proceed with the import process.

Cancel

Click the *Cancel* button to close the dialog without importing.

ER Mapper .ECW Import Options Dialog



Customize the ECW pixel reduction, import region, or read-only import in the ECW Image Import Options dialog.

Image Preview

The preview section displays a picture of the area to be imported. The preview section contains a low resolution preview of the area.

- Click the  buttons on the left side of the image preview to zoom in or out on the area. Changing the zoom level does not affect the area to import. The area to import is indicated in the preview section by a yellow outline.
- Click the  button to fit the entire image in the preview window.
- Click the  button on the left side of the image preview to quickly move the image in the preview section. Click and hold the left mouse button down and drag the image to change the view. Changing the view does not affect the area to import. The area to import is indicated in the preview section by a yellow outline.
- Click the  button on the left side of the image preview to change the extents of the image that is imported. The initial region includes the entire image extents. Click and hold the left mouse button down and drag the mouse over the area to import. The zoom extents update

and the yellow box coincides with the area drawn. The *Region selected for import* section also updates. Only the portion of the image highlighted by the yellow box will be imported.

Import Region

The region for import can be selected in the image preview or in the *Region selected for import* section. The initial region includes the entire image extents. Type a value in pixels into the *Top*, *Left*, *Right*, and *Bottom* fields to specify the imported region. The yellow outline in the image preview is updated as the values are changed.

Pixel Reduction

Since some ECW images can be extremely large once expanded into memory, the import filter allows the image to be reduced in dimensions by 1/2 to 1/32 of the original size via the options in the **ECW Image Import Options** dialog. Choose from *1/1* (Uses the most memory, best quality), *1/2*, *1/4*, *1/8*, *1/16*, to *1/32* (Uses the least memory, lowest quality). The *Final dimensions* field displays the final dimensions of the imported region. The *Estimated size* field displays the estimated image size in Megabytes (MB).

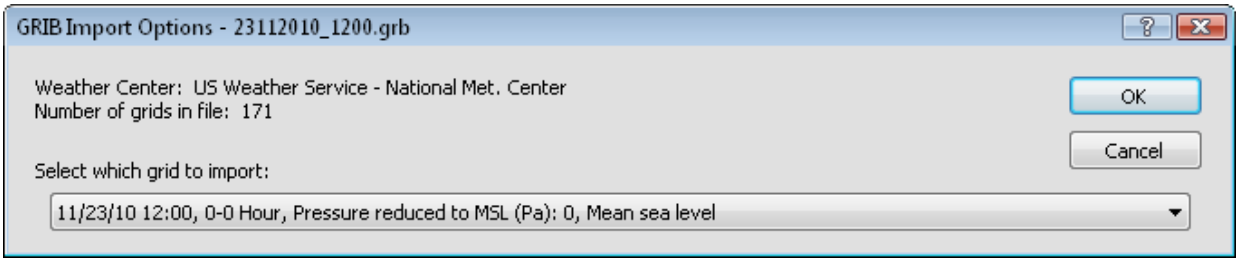
Read-only Import

The *Full resolution read-only 'on-the-fly' ECW image* option imports the full ECW file in its native form as a highly-compressed, read-only image. If a portion of the image is needed for display or output, the necessary portion is extracted and decompressed "on the fly." This option provides quick import and interactive panning and zooming without using large amounts of RAM or disk space. However once an image is imported with *Full resolution read-only 'on-the-fly' ECW image* checked, the image cannot be modified.

GRIB Import Options Dialog

The GRIB Import Options Dialog

Import a World Meteorological Organization .GRB file into **Surfer** and the **GRIB Import Options** dialog opens.



*Select which grid to import in the **GRIB Import Options** dialog.*

Select Which Grid to Import

Click on the existing grid name in the *Select which grid to import*. In the list, click on the desired grid and click *OK*. The grid is imported.

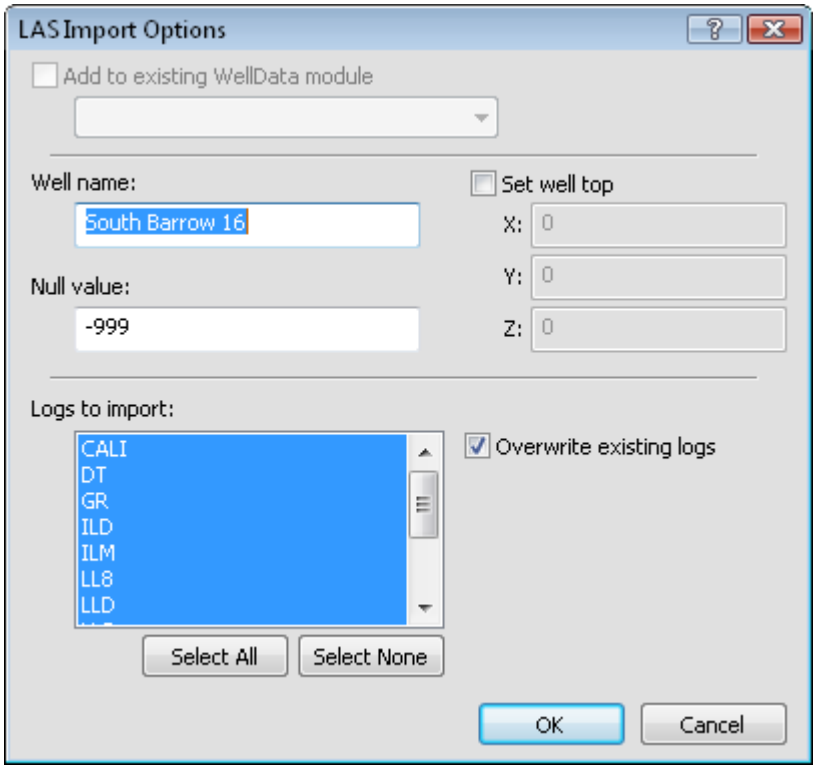
LAS Import Options Dialog

LAS files are files in the Log ASCII Standard format that contain log data for wells. In addition to the log data, .LAS files contain information about the header and parameter information about each

of the logs contained in the .LAS. The .LAS import filter reads the log data and automatically loads it into a *WellData* module.

LAS Import Options Dialog

Choose the **File | Import** command to load a .LAS file. The **LAS Import Options** dialog allows you to specify options which determine how information in the file is imported.



*Customize import options in the **LAS Import Options** dialog.*

Add to Existing WellData Module

Check the box next to *Add to existing WellData module* if the well information in the data file should be imported into an existing *WellData* module in the **Network Manager**. Importing into an existing *WellData* module keeps all of the well data information in a single module in the **Network Manager**. This makes it easier to extract data points from the wells and grid the data. It also makes it easier to apply the same well properties in a *WellRender* module. After selecting *Add to existing WellData module*, click on the existing *WellData* module name and click on the desired well module to change it. Multiple well nodes can be added to the same network to display different properties for different wells.

If the box is unchecked, a new *WellData* module will be created.

Well Name

The *Well name* option contains the name of the well being imported. To change the well name, highlight the existing text and type the name of the well that is being imported. By default, the file name is used for the well name. The well name is displayed in the **Property Manager** when a well node is selected.

Null Value

The *Null value* is the value that is used when data does not exist in a log column. Null values are not plotted in the *WellRender* graphic module. To change the value, highlight the existing value and type the desired value.

Well Top X, Y, and Z Values

Check the box next to *Set well top* to type in collar values for the well being imported. The *X*, *Y*, and *Z* options contain the coordinates of the top of the well in X, Y, and Z coordinates. To change the coordinate, highlight the existing value and type a new number. If no X, Y, or Z values are set for the well columns and the *Set well top* option is unchecked, the top is set to *Invalid*.

Logs to Import

The *Logs to import* section contains a list of all logs in the .LAS file. Click on a log name to select or deselect it. Only selected logs are imported. Selected logs appear highlighted in blue in the list.

To select multiple logs, click on the first log, hold down the SHIFT key on the keyboard, and click on the last log. All logs between the first and last log are selected. To select non-continuous multiple logs, click on one log, hold down the CTRL key on the keyboard, and click on each additional log. Only the selected logs are imported.

Select All

Click the *Select All* button to select all of the logs in the *Logs to import* box.

Select None

Click the *Select None* button to deselect all of the logs in the *Logs to import* box.

Overwrite Existing Logs

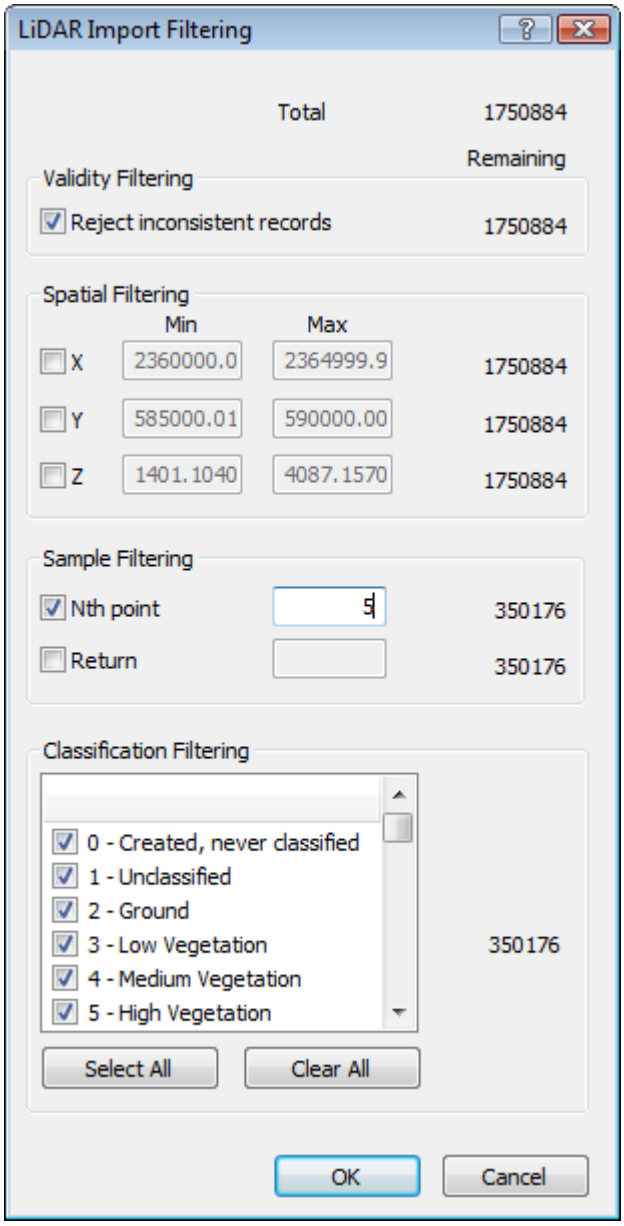
Check the box next to *Overwrite existing logs* to replace the data in the existing *WellData* module for the *Well name*. All data in the selected logs overwrites information that was previously imported with the same log name.

OK or Cancel

Click the *OK* button to import the file and close the dialog. Click the *Cancel* button to close the dialog without importing the file.

LiDAR Import Filtering Dialog

Click the **File | Import** command or the **File | Open** command to load a .LAS file. The **LiDAR Import Filtering** dialog allows you to specify options which determine how information in the file is imported.



*Customize import options in the **LiDAR Import Filtering** dialog.*

Total Points

The total number of points in the data file are listed at the top right of the dialog. As LiDAR data is typically extremely large, it is recommended that either *Spatial Filtering* or *Sample Filtering* be applied to the data.

Validity Filtering

Check the box next to the *Reject inconsistent records* option to remove all invalid or inconsistent data from import. When this box is unchecked, all data are passed to the import filter when importing a .LAS file.

Spatial Filtering

Check the boxes next to *X*, *Y*, or *Z* to filter the data based on the X, Y, or Z data limits. This is useful for selecting only a portion of the data based on geographic extents. After checking the box, highlight the value next to *X*, *Y*, or *Z* in either the *Min* or *Max* column. Type a new value. The number of points in the *Remaining* column will update to show how many points will exist in the data table after the filtering has been done on the data.

Sample Filtering

Check the box next to *Nth point* or *Return* to filter the data further. The *Nth point* removes every nth point, as set by the value *N*. If you want every other point, type 2 in the box next to *Nth point*. If you want every third point, type 3 in the box.

The *Return* value filters the data based on the laser return. Type the value of the laser return in the box next to *Return*. The return is the filter applied directly to the LAS data detecting waveform peaks. The timing of the peaks is given in the LAS file as a distinct *Return*. Most LiDAR systems designed for topographic mapping are optimized to record 3 return pulses. First returns can be used to create digital surface models that include features above the ground surface, such as buildings, bridges, and tree canopy. Intermediate returns are helpful in separating vegetation from solid objects among the above ground features. Final returns are normally the first approximation of the bare ground surface. If you only want the data from the ground surface, you might type in 3 for the *Return* value.

Classification Filtering

To import only data with a certain classification, check the box next to the classification type. For instance, if you only want to import data that is classified as LiDAR class *2-Ground*, uncheck all other boxes in the *Classification Filtering* section and check the box next to *Ground*.

Select All or Clear All

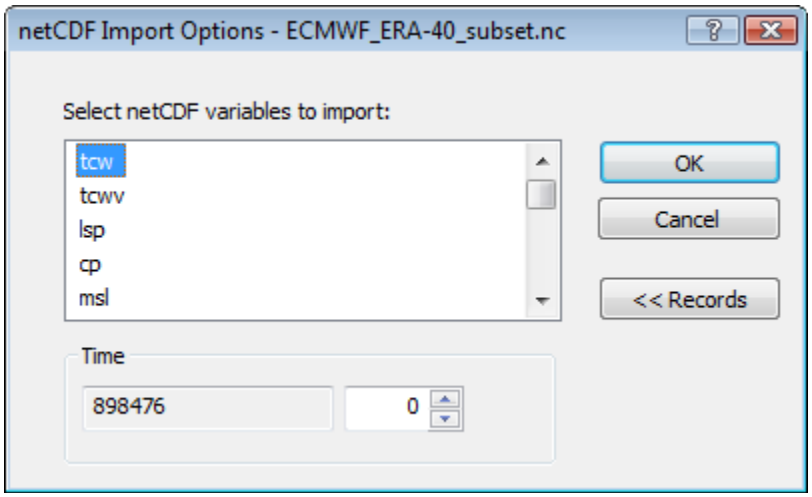
Click the *Select All* button to check all of the classes in the *Classification Filtering* section. Click the *Clear All* button to uncheck all of the classes in the *Classification Filtering* section. After clicking the *Clear All* button, one or more individual classes should be checked. Otherwise, no information will be imported.

OK or Cancel

Click *OK* to close the **LiDAR Import Filtering** dialog and import the .LAS file. Click *Cancel* to close the **LiDAR Import Filtering** dialog and not import the .LAS file.

NC NetCDF Import Options Dialog

Voxler imports and exports NetCDF .NC lattice files. These files are uniform grids.



Select the netCDF variable to import and set the Time record.

Variable

A NetCDF file may contain multiple variables. In the *Select netCDF variables to import* section, click on the desired variable to import. Only one variable may be imported at a time, and only the selected variable will be imported.

Records

The *Records* button is available if the selected variable is a record variable. If the variable is not a record variable, the *Records* button is unavailable.

Time

The *Time* option lets the user select which record to load. The coordinate value for the unlimited dimension at that record index is shown in the text control on the left. Note that if a variable has an unlimited dimension other than the first dimension, it will not be considered importable by the filter.

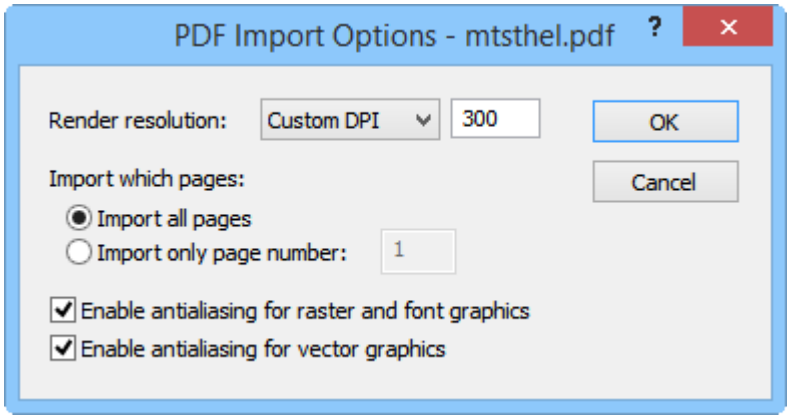
The filter treats record variables specially. Record variables have one or more unlimited dimensions. The n-th record contains all the variable data for the n-th dimension index.

PDF Import Options Dialog

The **PDF Import Options** dialog is displayed when importing PDF files. All objects from the PDF file are imported as images.

GeoPDF

When the PDF file is a GeoPDF file and contains coordinates, the PDF is imported with the coordinates.



*The **Import Options** dialog controls what is imported from the PDF file.*

Render Resolution

The *Render resolution* option controls the resolution of the imported images. Available options are 50, 75, 100, 125, 150, 200, 250, 300, 600, 1200 or *Custom* DPI. The higher the DPI, the clearer the image is when imported, but the larger the file size. The default is 150 DPI. When *Custom DPI* is selected, type a value into the *Render resolution* box to specify the resolution of the imported image. The *Render resolution Custom DPI* value must be between 50 and 1200.

Import Which Pages

When a PDF contains more than one page, the *Import which pages* allows control over which page is imported into the program. *Import all pages* imports each page in the PDF as a separate image. *Import only page number* imports only the specified page number into the program. To change the page number imported, select the *Import only page number* option and highlight the current value and type the page number to import.

Antialiasing

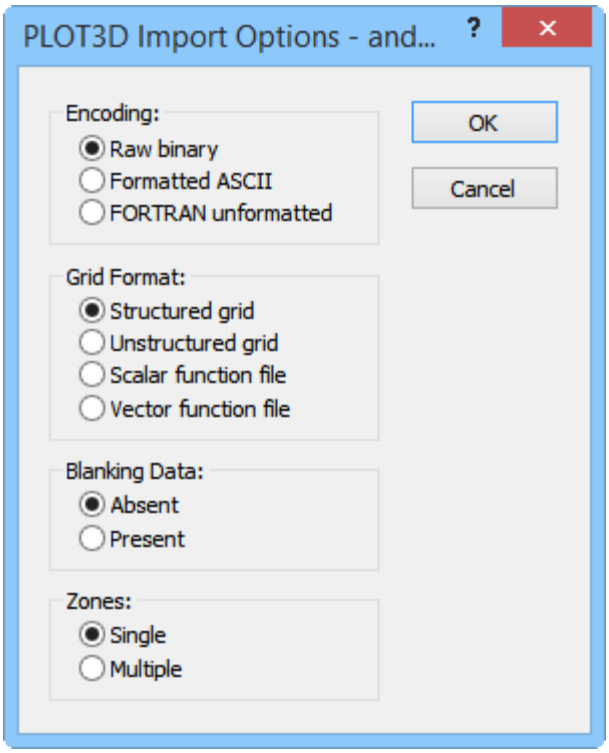
Check the box next to *Enable antialiasing for raster and font graphics* to smooth text and images in the PDF file. Check the *Enable antialiasing for vector graphics* to smooth lines, polygons, and other vector elements in the PDF. Checking these boxes results in a smoother appearance to the imported PDF. Unchecking these boxes can result in objects in the images appearing pixelly.

PLOT3D Import Options Dialog

The **PLOT3D Import Options** dialog allows you to specify the encoding, grid format, presence or absence of blanking data, and zones of an imported .P3D or .XYZ file.

PLOT3D Import Options Dialog

This dialog appears when the **File | Import** command is chosen and *PLOT-3D (*.p3d, *.xyz)* is chosen as the import file type.



*Use the **PLOT3D Import Options** dialog to specify options for the imported .P3D or .XYZ file.*

Encoding

The *Encoding* option specifies how the data values are encoded in the Plot-3d files. Select *Raw binary*, *Formatted ASCII*, or *FORTRAN unformatted.*

Grid Format

The *Grid Format* option specifies which of the four possible structures of the Plot-3d data set is contained in the import file. Choose *Structured grid*, *Unstructured grid*, *Scalar function file*, or *Vector function file.*

Blanking Data

The *Blanking Data* option specifies whether blanking data is present in or absent from the imported file. Select *Absent* or *Present*.

Zones

The *Zones* option specifies whether the data in the imported file is stored in *Single* or *Multiple* zone format.

OK

Click the *OK* button to proceed with the import process.

Cancel

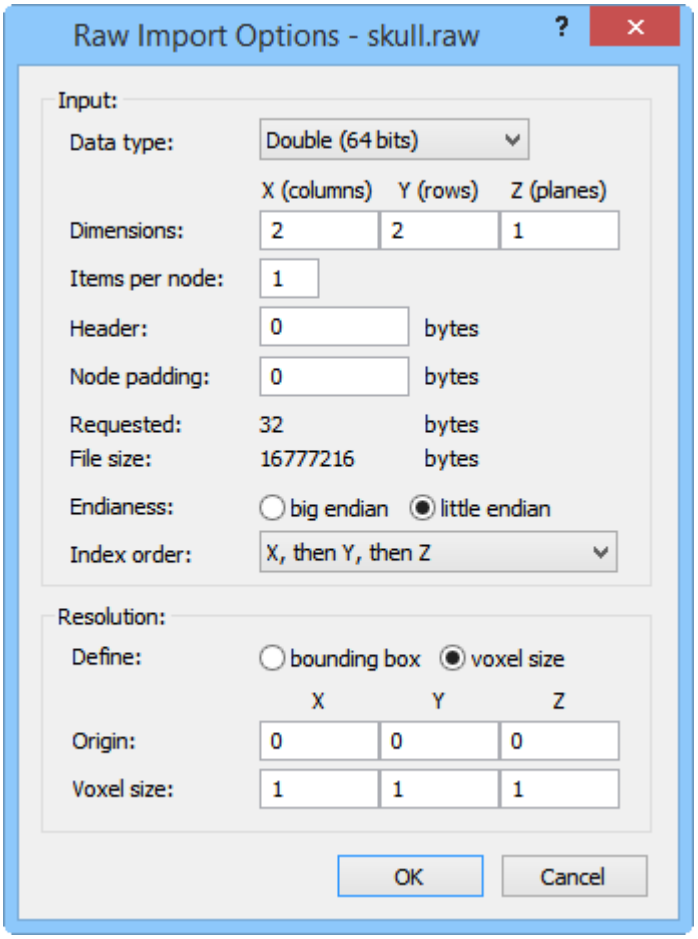
Click the *Cancel* button to close the dialog without importing.

RAW Import Options Dialog

The **RAW Binary Lattice Import Options** dialog allows you to specify the data type, dimensions, header bytes, node spacing, endianess, index order, and resolution of an imported .RAW data file.

RAW Binary Lattice Import Options Dialog

This dialog appears when the **File | Import** command is chosen and *Raw Binary Lattice (*.raw, *.bin)* is selected as the file type to import.



*Use the **Raw Import Options** dialog to specify options for the imported .RAW or .BIN file.*

Data Type

The *Data type* option allows you to select the type of data to be read from the import file and how many data components of this type should be read for each lattice node.

Dimensions

The *Dimensions* option allows you to specify the three dimensions of the lattice in the *X (columns)*, *Y (rows)*, and *Z (planes)* boxes. The values must be positive.

Items per Node

The *Items per node* value displays the number of components that are associated with each point in the .RAW file.

Header Bytes

The *Header* option allows you to specify how many bytes of non-data should be skipped at the beginning of the file before reading the first item of data. Enter a value in the *Header* box.

Node Padding

The *Node padding* option allows you to specify how many bytes of non-data should be skipped between each data item. This value is zero for most files. Enter a value in the *Node padding* box.

Requested Bytes (Static control)

The *Requested Bytes* option displays how many bytes would be read from the file based on your selection in the above controls.

File Size (Static control)

The *File size* option displays the actual size of the file in bytes.

Endianess

The *Endianess* option allows you to specify whether the import data is in *little-endian* or *big-endian* byte order. Click the button to left of *little-endian* or *big-endian* to select a byte order. Integers are usually stored as a sequence of bytes, so that the encoded value can be obtained by simple concentration. The two most common are litte-endian and big-endian. Little-endian refers to increasing numeric significance with increasing memory addresses. Big-endian is the opposite and refers to "big end first".

Index Order

The *Index order* option allows the user to specify whether the precedence of the lattice data in the file is in x,y,z (x fastest) or z,y,x (z fastest) order.

Resolution

The *Resolution* controls allows you to specify the size (in application display units) of the imported lattice. Two methods are available:

- *bounding box*, which allows you to specify the extents of the lattice's bounding box (from which the voxel size is calculated); and
- *voxel size*, which allows you to specify the extents of each voxel (from which the bounding box is calculated).

OK

Click the *OK* button to proceed with the import process.

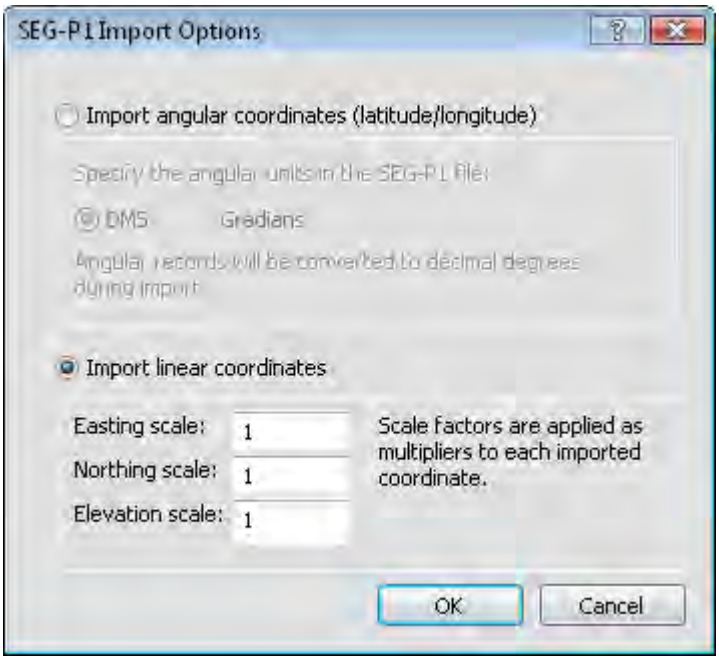
Cancel

Click the *Cancel* button to close the dialog without importing.

SEG P1 Import Options Dialog

The **SEG-P1 Import Options** dialog allows you to specify options which determine how information in the file is imported.

A SEG P1 file is a collection of point locations, usually used in geophysical shotpoints.



Specify the SP1 import options in the ***SEG-P1 Import Options*** *dialog.*

Coordinate Format

Select *Import angular coordinates (latitude/longitude)* if the data in the SP1 or SEG file is in latitude and longitude. Select *Import linear coordinates* if the data in the SP1 or SEG file is in a Cartesian coordinate system or an unknown linear coordinate system. Units can be in degrees, gradians, meters, feet, or any unknown units.

DMS or Gradians

When the *Import angular coordinates (latitude/longitude)* is selected, the *Specify the angular units in the SEG-P1 file* option becomes available. If the file contains degrees, minutes, and seconds, select *DMS*. If the file contains gradians, select *Gradians*.

Scale Factors

When the *Import linear coordinates* is selected, the *Easting scale*, *Northing scale*, and *Elevation scale* options are available. Highlight the existing value and type the new scale factor value in the box. Scale factors are multiplied by the values in the import file to get appropriate coordinates. This might be necessary if the *Easting* and *Northing* are in one set of units and the *Elevation* is in another set of units.

OK or Cancel

Click *OK* to continue importing the tabular SP1 data into a new table or the vector format SP1 file into a new base map. Click *Cancel* to quit importing the file.

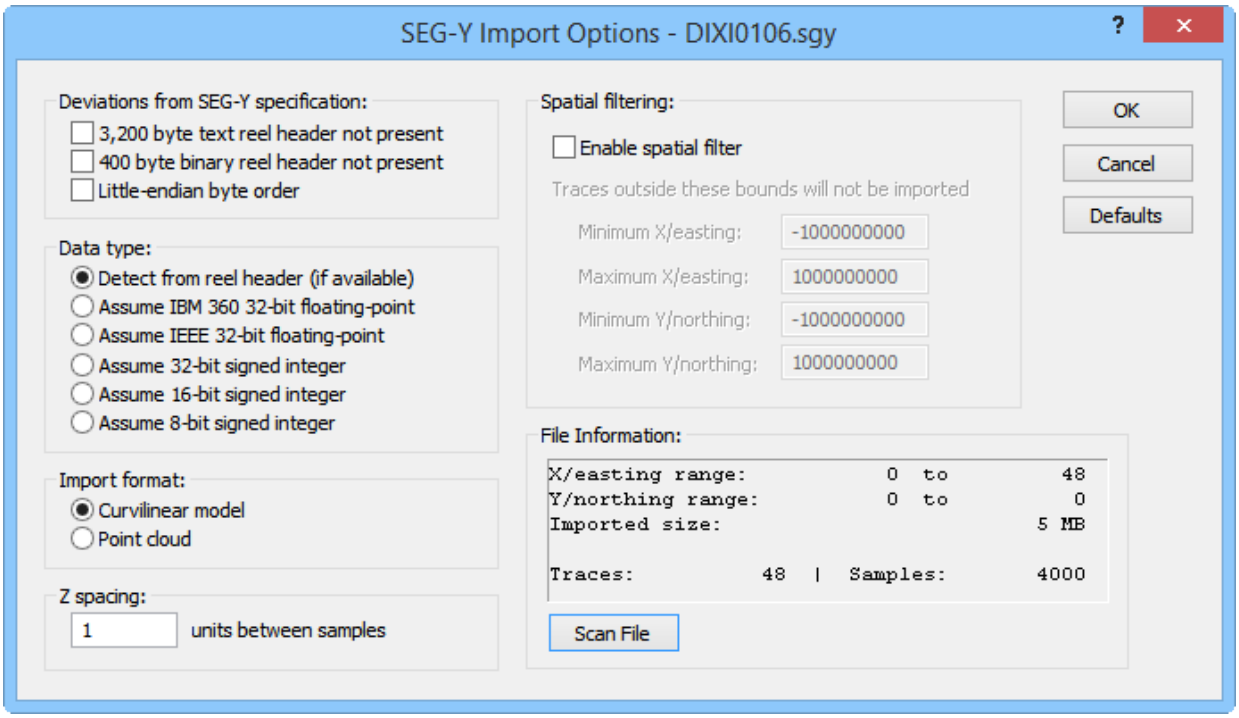
SEGY Import Options Dialog

The **SEGY Import Options** dialog allows you to specify the deviations from specification, data type for seismic sample values, point coordinate axes, and coordinate values for an imported .SGY or .SEGY file.

It is possible for SEG-Y files to have zeros or nonsensical values in the trace coordinates. When this is the case, coordinates from 0 to N will be automatically generated. The coordinate scaling information from the trace headers will be applied to the generated coordinates.

SEGY Import Options Dialog

This dialog appears when the **File | Import** command is chosen and *SEG-Y Seismic Data Log (*.sgy, *.segy)* is selected as the file type to import.



*Use the **SEGY Import Options** dialog to specify options for the imported .SGY or .SEGY file.*

Deviations From SEG-Y Specification

A SEG-Y file that conforms to the SEG-Y specification contains a 3,200-byte "reel header" data structure before the rest of the information in the file. However, many SEG-Y files do not conform to the specification in this regard. Check the box next to *3,200 byte text reel header not present* if the import file does not contain the 3,200 byte reel header.

A SEG-Y file that conforms to the SEG-Y specification contains a 400 byte binary "reel header" data structure before the rest of the information in the file. However, many SEG-Y files do not conform to the specification in this regard. Check the box next to *400 byte binary reel header not present* if the import file does not contain the 400 byte reel header.

A SEG-Y file that conforms to the SEG-Y specification contains data in big-endian byte order. However, many SEG-Y files do not conform to the specification in this regard. Check the box next to *Little-endian byte order* if the import file contains data in little-endian byte order.

Data Type

The *Data type* option allows the user to specify the data type for the seismic sample values in the import file.

- Select *Detect From header or use default* for files containing valid headers. Without a valid header, the default data type is assumed.
- Select *Assume IBM 360 32-bit floating point* and specify that the import file has IBM 360 floats unless the file contains a valid header for another type.
- Select *Assume IEEE 32-bit floating point* to specify that the import file contains IEEE floats unless the file contains a valid header specifying another type.
- Select *Assume 32-bit signed integer* to specify that the import file contains 32-bit integer data.
- Select *Assume 16-bit signed integer* to specify that the import file contains 16-bit integer data.
- Select *Assume 8-bit signed integer* to specify that the import file contains 8-bit integer data.

Import Format

The *Import format* controls specify how the data samples from the import file are structured during the import process. The possible selections are:

- *Import data as unstructured marker set(s)* - Each sample is loaded as one point in a **Voxler** point set. Use the *Define point coordinate axes* control to determine how the samples are organized when the point set format is used. Each of the three axes may be assigned one of the following: *Sample value (amplitude)*, which means that the amplitude of the sample determines the position along the axis; *Relative sample number*, which means that the relative sample number (within each trace) determines the position along the axis; *Absolute sample number*, meaning that the absolute sample number (within all traces) determines the position along the axis; and *Zero*, in which all nodes are zero for this axis.
- *Import data as 2D lattice* - Each trace of seismic samples from the import file is imported as one column of a two-dimensional lattice. This allows the traces to be easily viewed or manipulated in side-by-side format in the software. This format may be desirable if the traces are from multiple seismic recorders, or multiple events at the same recorder (rather than sequential parts of the same event recording at one recorder).
- *Import Data as 1D Lattice* - The traces of seismic samples from the import file are concatenated into one long 1D lattice. This format may be desirable if the traces are sequential parts of one continuous piece of paper tape (rather than multiple tapes from multiple seismic recorders, or multiple events at the same recorder).

Z Spacing

The *Z spacing* value specifies the number of units between samples. The *Z spacing* value can be positive or negative. Positive *Z spacing* values correspond with depth in the SEGY file, i.e. the z value increases as the sample moves "down" geographically. Negative *Z spacing* values correspond with elevation values in the SEGY file, i.e. the z values increases as the sample move "up" geographically.

If the imported data is oriented incorrectly, try changing the sign of the *Z spacing* value.

Spatial Filtering

The *Spatial Filtering* controls the region to be imported. Click the *Enable spatial filter* check box to specify the minimum and maximum coordinate extents for the imported region.

- Type the minimum X (westernmost) value to be imported in the *Minimum X/easting* field.
- Type the maximum X (easternmost) value to be imported in the *Maximum X/easting* field.
- Type the minimum Y (southernmost) value to be imported in the *Minimum Y/northing* field.
- Type the maximum Y (northernmost) value to be imported in the *Maximum Y/northing* field.

File Information

The *File Information* section displays the *X/easting range*, *Y/northing range*, *Imported size*, *Size after spatial filter* (if applicable), and the number of *Traces* and *Samples*. To view the file information, first correctly specify the *Deviations from SEG-Y specification*, *Data type*, and *Import format* options. Next click the *Scan file* button. An error is displayed if the import options are not specified correctly.

Defaults

Click the *Defaults* button to revert the import options to the default values.

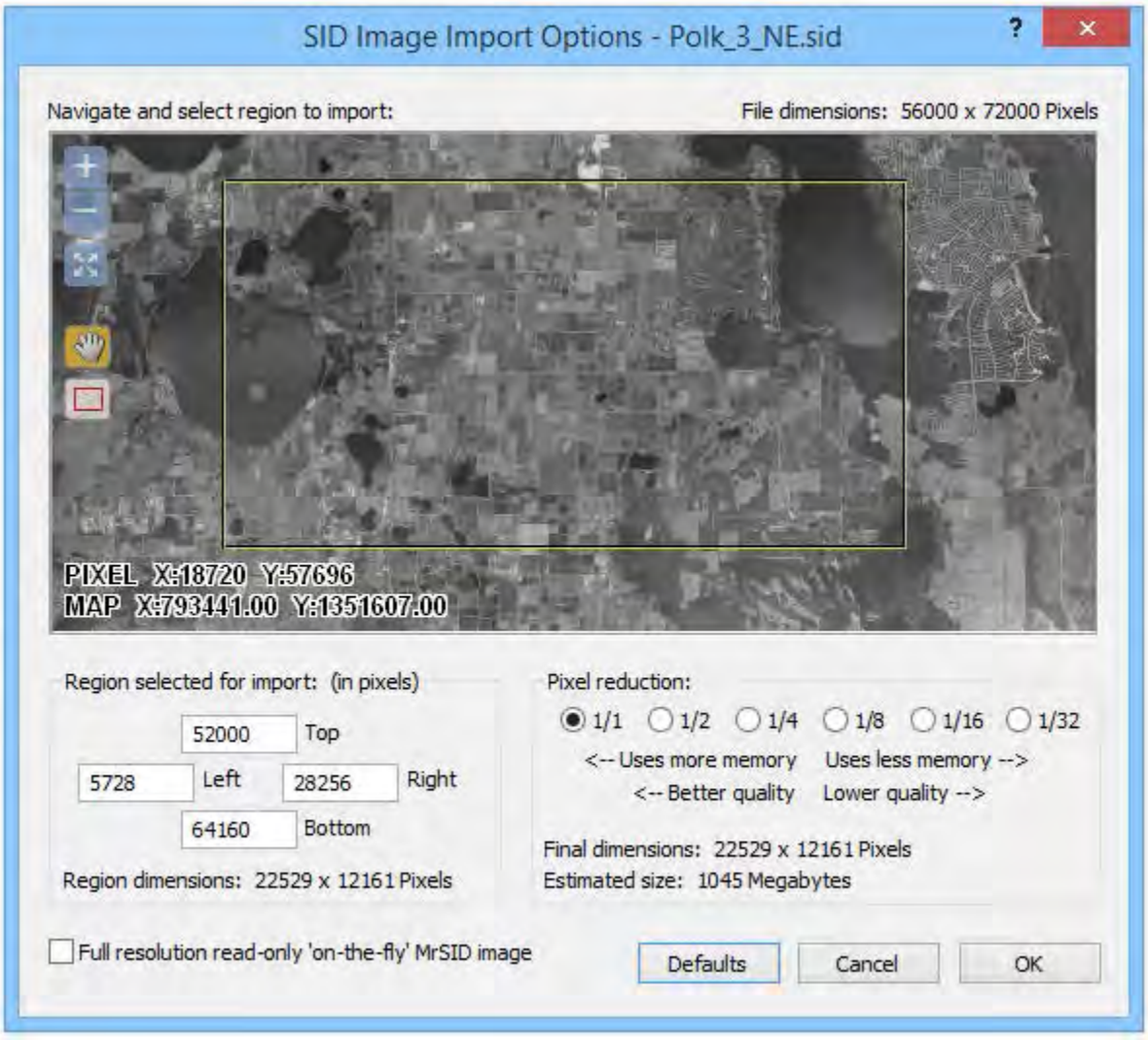
OK

Click the *OK* button to proceed with the import process.

Cancel

Click the *Cancel* button to close the dialog without importing.

LizardTech MrSID .SID Import Options Dialog



*Customize the SID pixel reduction, import region, or read-only import in the **SID Image Import Options** dialog.*

Image Preview

The preview section displays a picture of the area to be imported. The preview section contains a low resolution preview of the area.

- Click the  buttons on the left side of the image preview to zoom in or out on the area. Changing the zoom level does not affect the area to import. The area to import is indicated in the preview section by a yellow outline.
- Click the  button to fit the entire image in the preview window.
- Click the  button on the left side of the image preview to quickly move the image in the preview section. Click and hold the left mouse button down and drag the image to change the view. Changing the view does not affect the area to import. The area to import is indicated in the preview section by a yellow outline.

- Click the  button on the left side of the image preview to change the extents of the image that is imported. The initial region includes the entire image extents. Click and hold the left mouse button down and drag the mouse over the area to import. The zoom extents update and the yellow box coincides with the area drawn. The *Region selected for import* section also updates. Only the portion of the image highlighted by the yellow box will be imported.

Import Region

The region for import can be selected in the image preview or in the *Region selected for import* section. The initial region includes the entire image extents. Type a value in pixels into the *Top*, *Left*, *Right*, and *Bottom* fields to specify the imported region. The yellow outline in the image preview is updated as the values are changed.

Pixel Reduction

Since some SID images can be extremely large once expanded into memory, the import filter allows the image to be reduced in dimensions by 1/2 to 1/32 of the original size via the options in the **SID Image Import Options** dialog. Choose from *1/1* (Uses the most memory, best quality), *1/2*, *1/4*, *1/8*, *1/16*, to *1/32* (Uses the least memory, lowest quality). The *Final dimensions* field displays the final dimensions of the imported region. The *Estimated size* field displays the estimated image size in Megabytes (MB).

Read-only Import

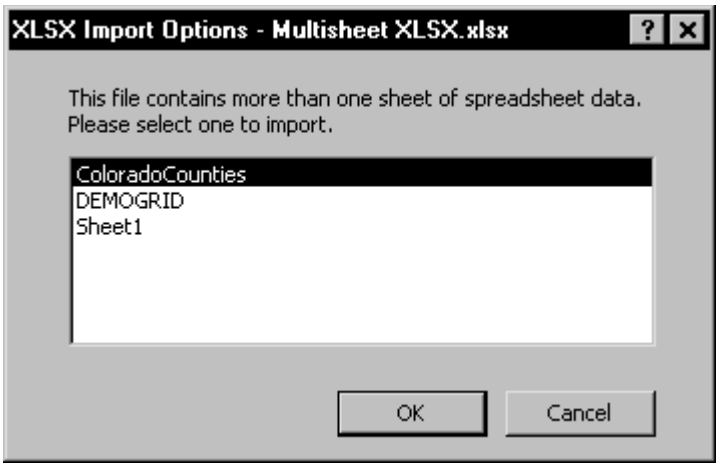
The *Full resolution read-only 'on-the-fly' MrSID image* option imports the full MrSID file in its native form as a highly-compressed, read-only image. If a portion of the image is needed for display or output, the necessary portion is extracted and decompressed "on the fly." This option provides quick import and interactive panning and zooming without using large amounts of RAM or disk space. However once an image is imported with *Full resolution read-only 'on-the-fly' MrSID image* checked, the image cannot be modified.

Excel .XLSX Import Options Dialog

The Excel filter imports .XLS and .XLSX file types.

Importing an Excel File with Multiple Sheets

One sheet of Excel spreadsheet data can be imported at a time. If you are importing an Excel file with multiple sheets, the **XLSX Import Options** dialog will appear.



*Select the sheet to import in the **XLSX Import Options** dialog.*

Multiple Sheets

If the Excel spreadsheet has multiple sheets, they are displayed. Left-click to select one sheet to import and click the *OK* button.

OK

Click the *OK* button to proceed with the import process.

Cancel

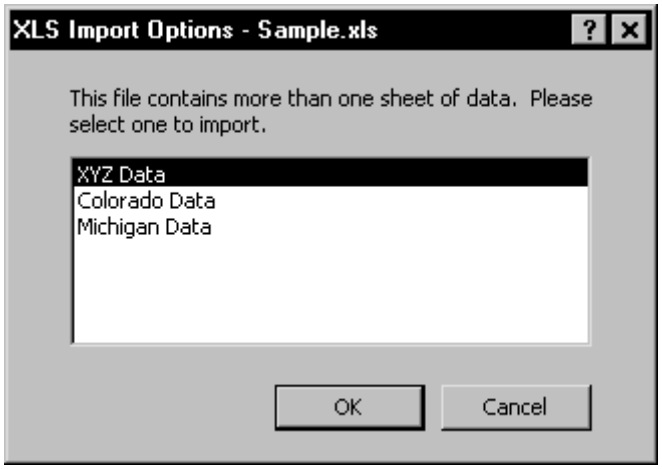
Click the *Cancel* button to close the dialog without importing.

Excel .XLS Import Options Dialog

The Excel filter imports .XLS and .XLSX file types.

Importing an Excel File with Multiple Sheets

One sheet of Excel spreadsheet data can be imported at a time. If you are importing an Excel file with multiple sheets, the **XLS Import Options** dialog will appear.



*Select the sheet to import in the **XLS Import Options** dialog.*

Multiple Sheets

If the Excel spreadsheet has multiple sheets, they are displayed. Left-click to select one sheet to import and click the *OK* button.

OK

Select one sheet to import and click the *OK* button to proceed with the import process.

Cancel

Click the *Cancel* button to close the dialog without importing.

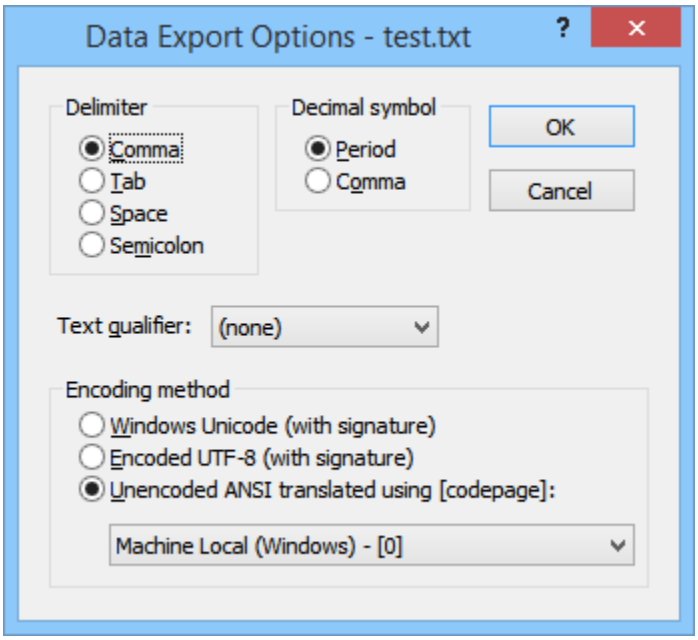
Export Options Dialogs

Data Export Options Dialog

The **Data Export Options** dialog is used to determine the type of delimiter to separate data in an output data file.

Data Export Options Dialog

This dialog appears when the **File | Save Data** command is used to export a tabular data from delimited text files (i.e. .CSV, .DAT, .TXT). These file formats are assumed to have one record per line in which each record contains a fixed number of numeric data fields.



*Specify the delimiter and text qualifier in the **Data Export Options** dialog.*

Delimiter

Choose *Comma*, *Tab*, *Space*, or *Semicolon* as the character to use to delimit fields in the saved .DAT, or .TXT file.

Decimal Symbol

Select the option in the *Decimal symbol* group that should be used to as the decimal symbol. When *Comma* is selected, the number 123 and 45 hundredths would appear in the file as 123,45. When *Period* is selected, the number would appear in the file as 123.45. This option is only available with .TXT files.

Text Qualifier

Select *(none)* for no qualifiers in the export file. Select *Double quotes* to place double quote characters "" around fields in the export file. Select *Single quotes* to place single quote characters '' around fields in the export file.

Encoding Method

The *Encoding method* section allows the choice of *Windows Unicode* data, *Encoded UTF-8* data, or *Unencoded ANSI translated using* data when exporting or saving an ASCII data file. *Windows Unicode* and *Encoded UTF-8 data* are often referred to as international data. It would include character sets from Russia, Israel, China, Greece, Hungary, among others. If the data does not appear correctly in the exported file, the *Encoding method* may be specified incorrectly.

ANSI encoding contains characters within the first 256 characters of a font. These are normally in English. After selecting *Unencoded ANSI translated using [codepage]*, select the *codepage* from the list that will read the data correctly.

OK and Cancel Buttons

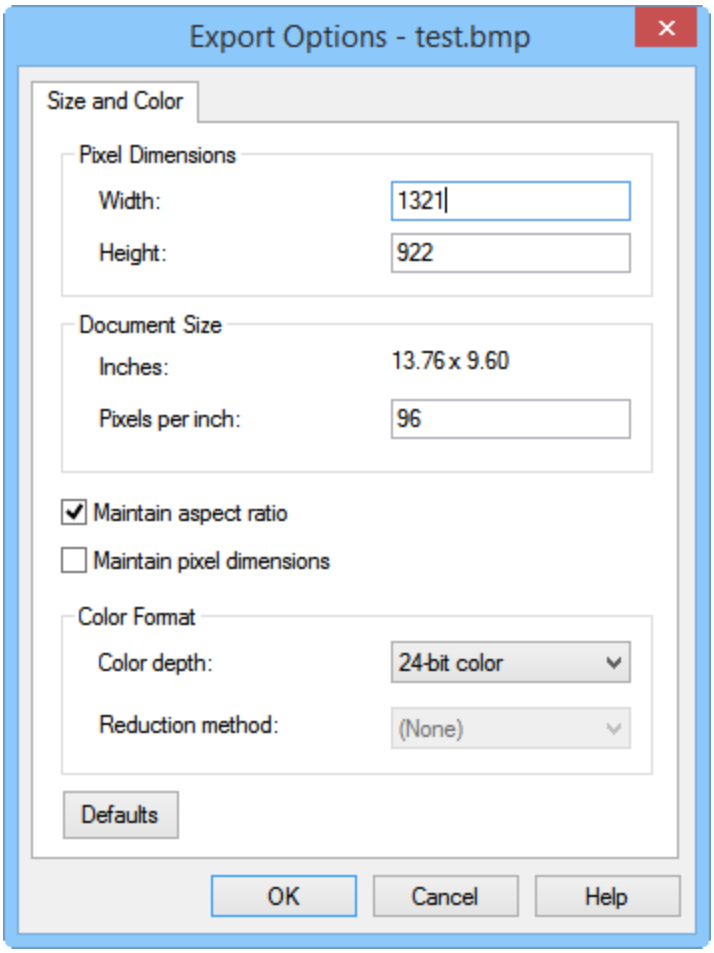
Click the *OK* button to proceed with the export process, or click the *Cancel* button to close the dialog without exporting the data set.

Export Options Dialog

The **Export Options** dialog allows you to specify options to determine how the image is exported.

Size and Color Page

This dialog appears when the **File | Export** command is chosen and an image format is selected as the output file type.



*The **Size and Color** page of the **Export Options** dialog controls options for image export.*

You may specify the output bitmap size in pixels (by modifying the *Width* and *Height* controls). You may also specify a resolution in dots per inch (by modifying the *Pixels per inch* control) and let the application calculate the output bitmap size for you. Higher resolution will yield a better-looking image, but at the expense of requiring more memory and disk space to hold the bitmap.

Image Size In Pixels

Choose the default *Width* and *Height* pixel settings for the output bitmap. Values are in pixels. The larger the values, the larger the output image.

Pixels Per Inch

Choose the *Pixels per inch* to change the number of pixels in the output image. The *Width* and *Height* of the *Pixel Dimensions* changes accordingly.

The .GIF file format is set to 72 *Pixels per inch* for all .GIF images and cannot be changed. GIF images are always 72 DPI by definition. For higher quality images, it is suggested that PNG, TIFF, or BMP is used instead of GIF.

Image Size

The *Document Size* section contains a *Inches* option that displays the current selected image size in inches. The image size is updated when the *Pixel Dimensions* or the *Pixles per inch* values are adjusted.

Maintain Aspect Ratio

Check the *Maintain aspect ratio* box if you want the image to maintain an equal horizontal and vertical resolution. When this option is checked, the output image maintains an aspect ratio of 1:1 in the output image. Higher resolution yields a better-looking image, but keep in mind that more memory and disk space are required to hold a high-resolution image.

Maintain Pixel Dimenions

Check the *Maintain pixel dimensions* box if you want the image to export at the selected *Width* and *Height*, but with a different number of *Pixels per inch*. This results in the same number of pixels, but a different *Document Size*.

Color Format

The *Color Format* gives you the option to output your image with *Color depth*. The greater the color depth, the more faithfully the image will represent the colors assigned to objects in your document. Different output formats support different color depths. Some output formats support 256 colors only, while others also support True Color (16 million colors).

Select one of the options from the Color Depth drop-down list. The options are: 8-bit grayscale, 16-bit grayscale, 32-bit grayscale, .1-bit color indexed, 4-bit color indexed, 8-bit color indexed, or 24-bit true color.

For example, Windows .BMP format supports *monochrome*, *16 colors*, *256 colors* and *True Colo*r. Greater color depth will yield a better-looking image, but at the expense of requiring more memory and disk space to hold the image.

Reduction Method

If you select a color indexed *Color depth*, you can choose a *Reduction Method*. Select one of the options from the *Reduction Method* drop-down list. The options are: *Ordered Dither*, *Diffuse Dither*, *Popularity*, *MedianCut555*, or *MedianCut888*.

If you select a color indexed *Color depth*, you can choose a *Reduction method*. Select one of the options from the *Reduction method* drop-down list. The options are: *Ordered Dither*, *Diffuse Dither*, *Popularity*, *MedianCut555*, or *MedianCut888*.

Dithering determines how similar colors are distributed among clusters of pixels in the reduced image. Ordered dithering uses a repeating pattern. Diffuse dithering uses a pseudo-random pattern.

Quantization determines how the colors for the exported image are selected from the palette of 16 million possible colors. *Popularity* uses the most frequently occurring colors in the image, and *MedianCut* selects colors based on the 'median cut' method that tries to select the most even distribution of colors over the range of colors that appear in the image. The Median method can use either 5 or 8 bits of sample data for each of the three color planes in the image. Larger sample sizes require more memory to perform the conversion for export, so the smallest sample size that produces an acceptable image is recommended.

Defaults

Click the *Defaults* button to return the export options to the default selections.

OK

Click the *OK* button to save your changes and export the file.

Cancel

Click the *Cancel* button to exit the dialog without saving your changes or exporting the file.

Amira Mesh .AM, .COL Export Options

The **AMIRA Export Options** dialog determines how blank values are treated in the export file.

The AMIRA Export Options Dialog

This dialog appears when a module is selected, the **File | Save Data** command is chosen, and a lattice format that contains one or more blanked nodes is selected as the output file type.



*Control how the blank nodes are exported in the **AMIRA Export Options** dialog.*

Data Min

Choose *Data min* to export blank nodes with the indicated minimum value (varies depending on contents of lattice).

Data Max

Choose *Data max* to export blank nodes with the indicated maximum value (varies depending on contents of lattice).

Type Min

Choose *Type min* to export blank nodes with the indicated minimum value for the lattice's data type (usually zero).

Type Max

Choose *Type max* to export blank nodes with the indicated maximum value of the lattice's data type.

Fixed

Choose *Fixed* to export blank nodes with some other value, and enter the desired value in the adjacent edit control.

OK

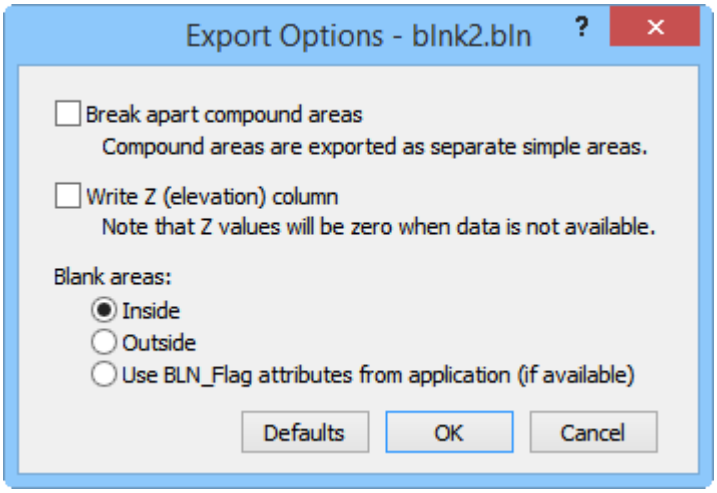
Click the *OK* button to save your changes and export the file.

Cancel

Click the *Cancel* button to exit the dialog without saving your changes or exporting the file.

Golden Software Blanking .BLN Export Options Dialog

The **Export Options** dialog allows you to specify options which determine how information in the file is exported.



*Specify the BLN export options in the **Export Options** dialog.*

Break Apart Compound Areas

Choose *Break apart compound areas* to have compound areas (those containing islands or lakes) output as separate area entities. This option should be chosen if the .BLN file is to be used as a boundary file. Do not choose this option if the .BLN file is to be used as a blanking file.

Write Z (elevation) column

Choose *Write Z (elevation) column* to export the .BLN file with the Z values.

Blank Areas

Select whether to blank areas *Inside* or *Outside*. Selecting *Inside* writes the blanking flag as 1 in the .BLN file. Selecting *Outside* writes the blanking flag as 0 in the .BLN file.

Voxler does not associate attributes with objects within *Geometry Source* modules. When *Use BLN_Flag attributes from application (if available)* is selected, the blanking flag is set to 1 for blank inside.

Defaults

The *Defaults* button sets all options to default conditions.

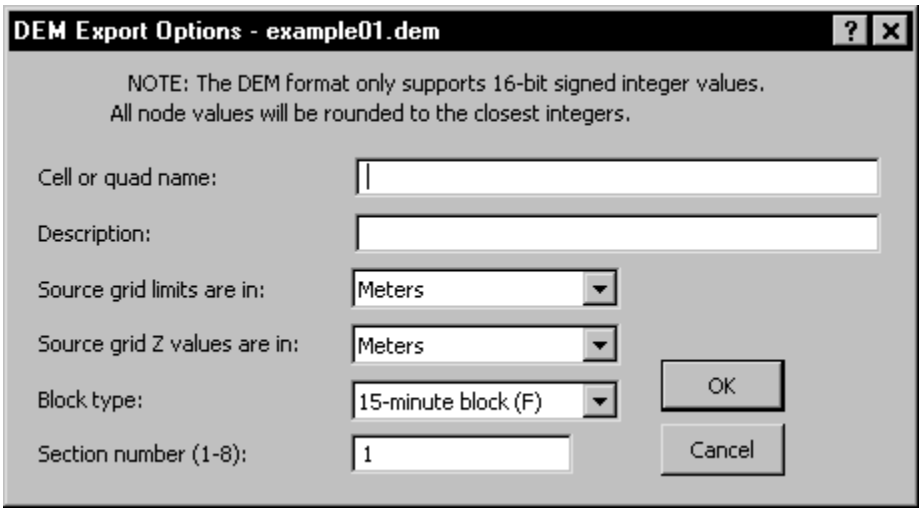
USGS Digital Elevation Model .DEM Export Options Dialog

The Select Slices Dialog

If the file format does not support multiple-slice lattices, the **Select Slices** dialog appears.

The DEM Export Options Dialog

This dialog appears when a lattice or image module is selected and the **File | Save Data** command is chosen.



*Customize the DEM options in the **DEM Export Options** dialog.*

Cell or quad name

Type a name for the DEM image in the *Cell or quad name* box.

Description

Type a description for the DEM image in the *Description* box.

Source Grid Limits

Specify the type of units for the XY coordinates in the source grid from the *Source grid limits are in* drop-down list. Click the arrow to display the list. Possible selections are *Radians*, *Feet*, *Meters*, *Arcseconds*, and *Degrees*. The export units need to be the same as the map units. For example, if the map units are lat/long, and you choose Meters as the X unit code, then there will be no conversion. If the map is in lat/long, then you have to choose to export as lat/long. You put in what the units of the map are, not what you want the export file to be in.

Source Grid Z Values

Specify the type of units for the Z (vertical) coordinates in the source grid from the *Source grid Z values are in* drop-down list. Click the arrow to display the list. Possible selections are *Feet* or *Meters*. The export units need to be the same as the map units. For example, if the map units are lat/long, and you choose Meters as the X unit code, then there will be no conversion. If the map is in lat/long, then you have to choose to export as lat/long. You put in what the units of the map are, not what you want the export file to be in.

Block type

Specify the block type for the DEM image from the *Block type* drop-down list. Click the arrow to display the list.. Possible selections are *15-minute*, *7-minute*, and *Other*.

Standards for Digital Elevation Models

Part 2: Specifications

National Mapping Program

Technical Instructions

U.S. Department of the Interior

U.S. Geological Survey

National Mapping Division

Appendix 2-I

Sectional indicator

The 30-minute DEM's are distributed in groups of files that make up a 30- by 30-minute area of coverage representing the DEM for the east or west half of a 1:100,000-scale source map. The normal distribution group is four 15-minute files per 30-minute area. The quadrangle name field in the header record contains the name of the 1:100,000-scale source map. However, the pieces or sections into which each is divided are identified within the header type A record to the size and placement of each. In byte 138-140 each section is identified by a 3 character code

XNN where:

X is a single letter indicating size

F = 15-minute block

S = 7.5-minute block

NN is a two-digit number indicating the specific quad. Figure 2-I-1 and 2-I-2 illustrate this division with the sections labeled with the code that appears in bytes 138-140 of the header record.

F01 F02 F03 F04

F05 F06 F07 F08

Figure 2-I-1

A 1:100,000-scale quad divided into eight 15-minute quads, 4 per 30-minute area.

S01 S02 S03 S04 S05 S06 S07 S08

S09 S10 S11 S12 S13 S14 S15 S16

S17 S18 S19 S20 S21 S22 S23 S24

S25 S26 S27 S28 S29 S30 S31 S32

Figure 2-I-2

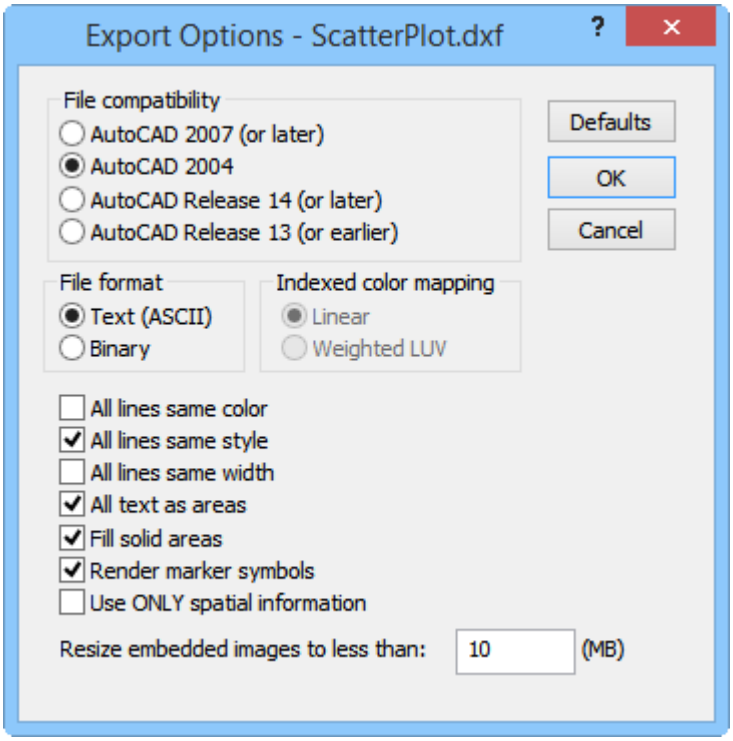
A 1:100,000-scale quad divided into 32 7.5-minute quads, 16 per 30-minute area.'

Section Number

Specify the section number by entering a number from one to eight in the *Section number (1-8)* box. Specify the section number by entering a number from one to eight for a 15-minute block. Specify the section number by entering a number from one to 32 for a 7.5-minute block. Ignore the section number for "Other" block types.

AutoCAD DXF Export Options Dialog

The **Export Options** dialog allows you to specify options which determine how information in the file is exported.



*Customize DXF export options in the **Export Options** dialog, **DXF Options** page.*

File Compatibility

Select the appropriate file compatibility. Available options are AutoCAD 2007 (or later), AutoCAD 2004, AutoCAD Release 14 (or later), and AutoCAD Release 13 (or earlier).

AutoCAD 2007 (or later)

Choose *File compatibility* of *AutoCAD 2007 (or later)* if the DXF file needs to be imported into AutoCAD 2007 or later. The 2007 version supports Unicode character encoding in the UTF-8 format.

AutoCAD 2004

Choose *File compatibility* of *AutoCAD 2004 (or later)* if the DXF file needs to be imported into AutoCAD 2004 or later. The 2004 version supports Unicode character encoding in the \U+XX format. The 2004 version (and all newer versions) also support RGB color support.

AutoCAD Release 14 (or later)

Choose *File compatibility* of *AutoCAD Release 14 (or later)* if the DXF file needs to be imported into AutoCAD Release 14 or later. The version 14 and previous versions support indexed color mapping.

AutoCAD Release 13 (or earlier)

Choose *File compatibility* of *AutoCAD Release 13 (or earlier)* if the DXF file needs to be imported into an earlier release of AutoCAD.

File Format

Choose *Text (ASCII)* or *Binary* to specify the format of the exported DXF file. See the AutoCAD DXF *File Description* for information on the organization of DXF files.

Indexed Color Mapping

Select the *Linear* or *Weighted LUV* option when either *AutoCAD Release 14 (or later)* or *AutoCAD Release 13 (or earlier)* is selected as the *File compatibility.* Note: Even though the older file formats can be imported into AutoCAD Release 14 and AutoCAD 2004 or later, the AutoCAD Release 14 or later file format uses several features that result in smaller, faster loading DXF files, and AutoCAD 2004 or later supports true color in addition to the indexed color.

All Lines Same Color

Choose *All lines same color* if you don't want an AutoCAD color number (1-255) assigned to each of your lines. The default color for the layer will be used instead. All exported graphical entities are assigned to a layer named GSLAYER.

All Lines Same Style

Choose *All lines same style* if you want exported lines to be assigned the default style (for the GSLAYER) when imported into **AutoCAD**. Otherwise, exported lines will retain their style (solid, dashed, etc.).

All Lines Same Width

Choose *All lines same width* if you want exported lines to be assigned the default width (for the GSLAYER) when imported into **AutoCAD**. Otherwise, exported lines will be the width assigned in the application document.

All Text As Areas

Voxler does not export text to DXF files.

Text can be exported as DXF solid polygons (*All text as areas* checked). These polygons will always be oriented properly. Whether or not these solid polygons (like all solid polygons) will be filled or not is controlled by the *Fill solid areas* option (see below).

Text can also be exported as **AutoCAD** text entities (*All text as areas* unchecked). No matter what typeface is specified in the application document, all text entities are assigned **AutoCAD's** STANDARD font. Once inside **AutoCAD**, the text entities can be edited in the normal **AutoCAD** fashion. As long as there is no shear, perspective, or clipping, DXF text entities will be exported as text. This means that the DXF text entities will be sized and oriented similar to the text objects in the application document. When shear, perspective, or clipping occur, the text is exported as solid polygons. Shear occurs when the character glyphs are not perpendicular to the text baseline. Perspective occurs when the height of glyphs in the text string are not all the same, as in a 3-D view where the glyphs are smaller the farther they are from the observer. Clipping occurs when part of the text object is partially inside and partially outside the map limits.

Fill Solid Areas

Choose *Fill solid area*s if you want the interior of solid areas (polygons) to be filled. Otherwise, the areas will be exported as AutoCAD CLOSED POLYLINE entities.

Render Marker Symbols

Check the *Render marker symbols* check box if the marker symbols should be exported to the DXF file. The marker symbols are exported as lines and polygons. When the *Render marker symbols* option is not checked, marker symbols are exported to the DXF file as points.

Use ONLY Spatial Information

Choose *Use ONLY spatial information* if you want to export only spatial information and not object attributes or text labels. Spatial information is only concerned with the location of objects in space (i.e., their coordinates) and not with their attributes (such as line or fill style, marker symbol used, text labels, etc.) For example, if this option is chosen, all text will be ignored, markers will be exported as point entities instead of polygonal glyphs and coordinates output to the DXF file will be stored in map units instead of inches. This is useful when exporting base maps when only the spatial information is desired.

The AutoCAD program's behavior when importing DXF files (via the DXFIN command) is different depending on whether the **AutoCAD** drawing file [.DRW] is brand new or already contains drawing entities. If the file is brand new, attributes (such as line style) are loaded from the Tables section, so lines encountered in the Entities section will have the proper line style (solid, dash, dash-dot, etc.). However, if an old drawing file is already open, **AutoCAD** will ignore the Tables section and only read the Entities section. If the DXF file contains lines with styles not already defined, **AutoCAD** will issue an error message and abort the DXF import. It is recommended you choose the All lines same style option when exporting DXF files that will be imported into existing **AutoCAD** drawings. **AutoCAD** will then assign the default style to all lines in the imported layer (named GSLAYER).

Resize Embedded Images to Less Than

The *Resize embedded images to less than* option specifies the maximum size (in megabytes) an embedded image is allowed to be. If an exported image exceeds this size, its resolution will be reduced so it doesn’t exceed the designated maximum size. Increase this value to get better looking images at the expense of larger export files.

Defaults

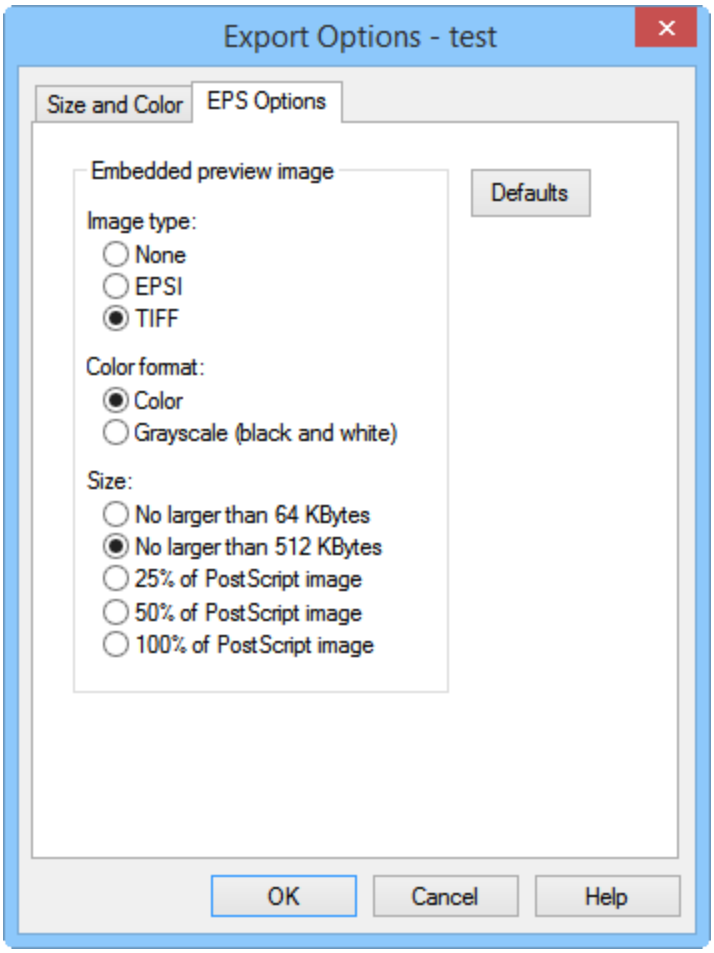
The *Defaults* button sets all buttons and check boxes to default conditions.

Encapsulated PostScript .EPS Export Options Dialog

The **Export Options** dialog allows you to specify options which determine how information in the .EPS file is exported.

Export Options

This dialog appears when the **File | Export** command is chosen or when an image module is selected and the **File | Save Data** command is chosen.



*Specify the EPS options in the **EPS Options** page of the **Export Options** dialog.*

Image Type

The *Image type* controls determine how the EPS preview image is encoded in the export file. Possible selections are *EPSI* or *TIFF* formats, or *None* for no preview image at all.

Color Format

The *Color format* controls determines whether the preview image is exported in *Color* or *Grayscale (black and white)*.

Size

The *Size* controls determine what size of EPS preview image is encoded in the export file. The selections are:

No larger than 64KBytes

No larger than 512Kbytes

25% of PostScript Image

50% of PostScript Image

100% of PostScript Image

Default

Click the *Default* button to return the EPS options to the default settings.

OK

Click the *OK* button to save your changes and export the file.

Cancel

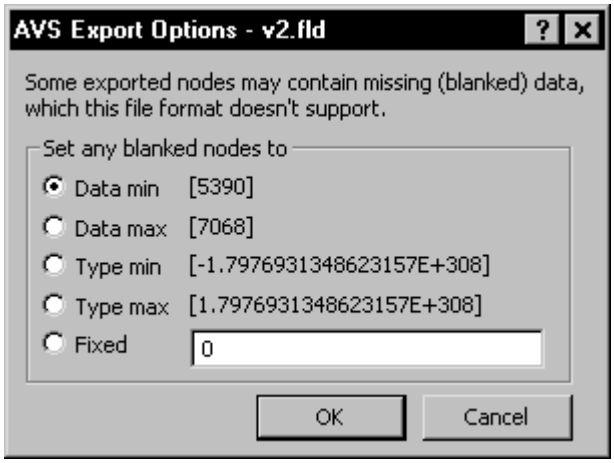
Click the *Cancel* button to exit the dialog without saving your changes or exporting the file.

AVS Field .FLD Export Options Dialog

The **AVS Export Options** dialog determines how blank values are treated in the exported .FLD file.

The AVS Export Options Dialog

The **AVS Export Options** dialog is only displayed if the lattice contains one or more blanked nodes. Select a data, lattice, or image module and choose the **File | Save Data** command to save an .FLD file and open the dialog.



*Control how the blank nodes are exported in the **AVS Export Options** dialog.*

Data Min

Choose *Data min* to export blank nodes with the indicated minimum value (varies depending on contents of lattice).

Data Max

Choose *Data max* to export blank nodes with the indicated maximum value (varies depending on contents of lattice).

Type Min

Choose *Type min* to export blank nodes with the indicated minimum value for the lattice's data type (usually zero).

Type Max

Choose *Type max* to export blank nodes with the indicated maximum value of the lattice's data type.

Fixed

Choose *Fixed* to export blank nodes with some other value, and enter the desired value in the adjacent edit control.

OK

Click the *OK* button to save your changes and export the file.

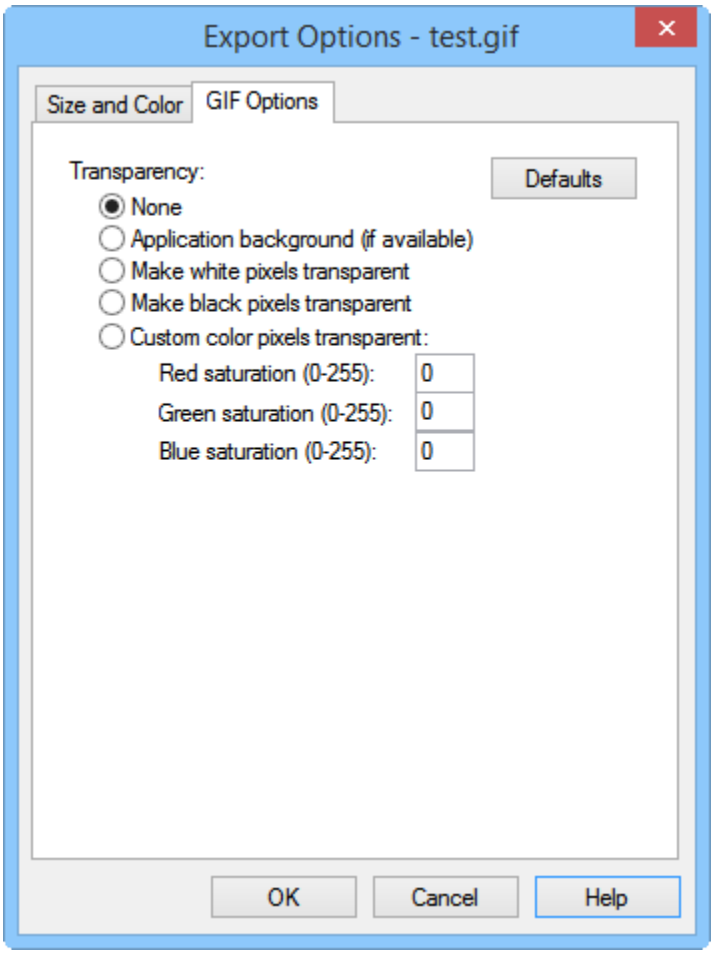
Cancel

Click the *Cancel* button to exit the dialog without saving your changes or exporting the file.

Graphics Interchange Format .GIF Export Options Dialog

The **GIF Options** page in the **Export Options** dialog controls how transparency is handled for the GIF export.

GIF Options



*Specify the GIF transparency in the **GIF Options** export dialog.*

Transparency

The *Transparency* options specify which, if any, color will be made transparent in the exported GIF image file.

- Select *None* to export the view in the Viewer Window as is.
- Select *Application background (if available)* to export the GIF with a transparent background. The *Reduction method* on the Size and Color page must be set to *Popularity*, *MedianCut555*, or *MedianCut888* to export the GIF with a transparent background.
- Select *Make white pixels transparent* to export all white pixels (color value R255 G255 B255) as transparent.
- Select *Make black pixels transparent* to export all black pixels (color value R0 G0 B0) as transparent.
- Select *Custom color pixels transparent* to specify which color will be exported as transparent. Type the red, green, and blue color saturation values for the desired color into the *Red saturation (0-255)*, *Green saturation (0-255)*, and *Blue saturation (0-255)* input boxes.

Defaults

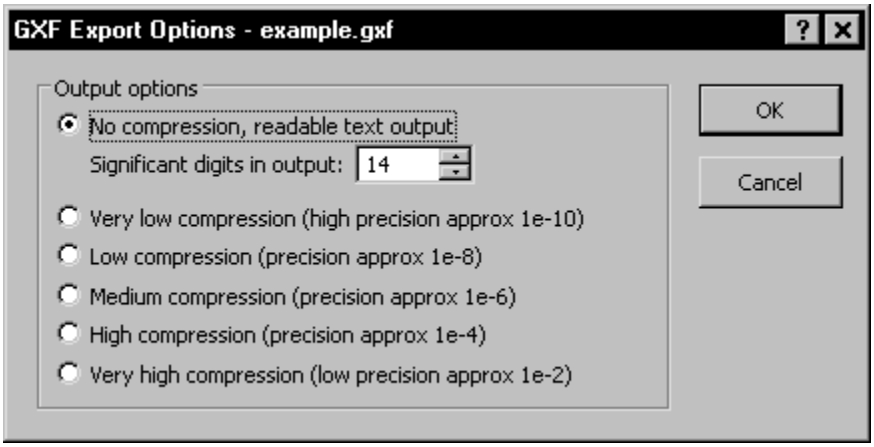
Click *Defaults* to return the options to their default values, i.e. *Transparency* is *None* and the *Custom color pixels transparent* color saturation values to 0.

Grid Exchange .GXF Export Options Dialog

The **GXF Export Options** dialog allows you to specify options which determine how information in the .GXF file is exported.

GXF Export Options Dialog

Select a lattice or image module and choose the **File | Save Data** command to save a .GXF file and open the **GXF Export Options** dialog.



*Specify the GXF output options in the **GXF Export Options** dialog.*

Output Options

The controls in the output options dialog determine which compression format is used when exporting the data. Compression methods that yield higher compression yield lower data precision, and vice versa.

No Compression, Readable Text Output

The *No Compression* selection exports the data in a human-readable ASCII text format with the specified number of significant digits in each number.

Very Low, Low, Medium, High, Very High Compression

The *Very Low*, *Low*, *Medium*, *High*, and *Very High* selections export the data in a binary floating-point format with the data precision indicated in parenthesis ( ) next to the corresponding selection.

OK

Click the *OK* button to save your changes and export the file.

Cancel

Click the *Cancel* button to exit the dialog without saving your changes or exporting the file.

Hierarchical Data Format .HDF Export Options Dialog

The **HDF Export Options** dialog determines how blank values are treated in the export file.

The HDF Export Options Dialog

The export options dialog is only displayed if the lattice contains one or more blanked nodes. Choose the **File | Save Data** command to save a .HDF file and open the **HDF Export Options** dialog.



*Control how the blank nodes are exported in the **HDF Export Options** dialog.*

Object name

Specify a name for the exported HDF object next to *Object name*.

Export Lattice as Raster Image(s)

If the object being exported is a lattice, it can be exported in raster image format instead of lattice format by checking the *Export lattice as raster image(*s) control.

Set Any Blanked Nodes To

The controls in the *Set any blanked notes to* group box determine how blank values are treated in the export file.

Data Min

Choose *Data min* to export blank nodes with the indicated minimum value (varies depending on contents of lattice).

Data Max

Choose *Data max* to export blank nodes with the indicated maximum value (varies depending on contents of lattice).

Type Min

Choose *Type min* to export blank nodes with the indicated minimum value for the lattice's data type (usually zero).

Type Max

Choose *Type max* to export blank nodes with the indicated maximum value of the lattice's data type.

Fixed

Choose *Fixed* to export blank nodes with some other value, and enter the desired value in the adjacent edit control.

OK

Click the *OK* button to save your changes and export the file.

Cancel

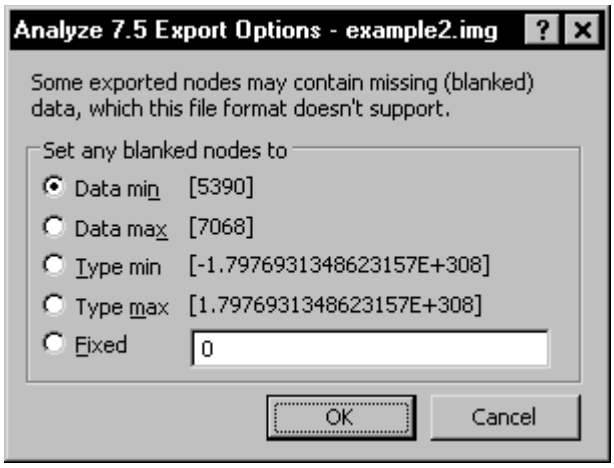
Click the *Cancel* button to exit the dialog without saving your changes or exporting the file.

Analyze 7.5 Medical Image .HDR, .IMG Export Options Dialog

The **Analyze 7.5 Export Options** dialog determines how blank values are treated in the export file.

The Analyze 7.5 Export Options Dialog

The export options dialog is only displayed if the lattice contains one or more blanked nodes. Select a lattice or image module and choose the **File | Save Data** command to save a .HDR, .IMG file and open the **Analyze 7.5 Export Options** dialog.



*Control how the blank nodes are exported in the **Analyze 7.5 Export Options** dialog.*

Data Min

Choose *Data min* to export blank nodes with the indicated minimum value (varies depending on contents of lattice).

Data Max

Choose *Data max* to export blank nodes with the indicated maximum value (varies depending on contents of lattice).

Type Min

Choose *Type min* to export blank nodes with the indicated minimum value for the lattice's data type (usually zero).

Type Max

Choose *Type max* to export blank nodes with the indicated maximum value of the lattice's data type.

Fixed

Choose *Fixed* to export blank nodes with some other value, and enter the desired value in the adjacent edit control.

OK

Click the *OK* button to save your changes and export the file.

Cancel

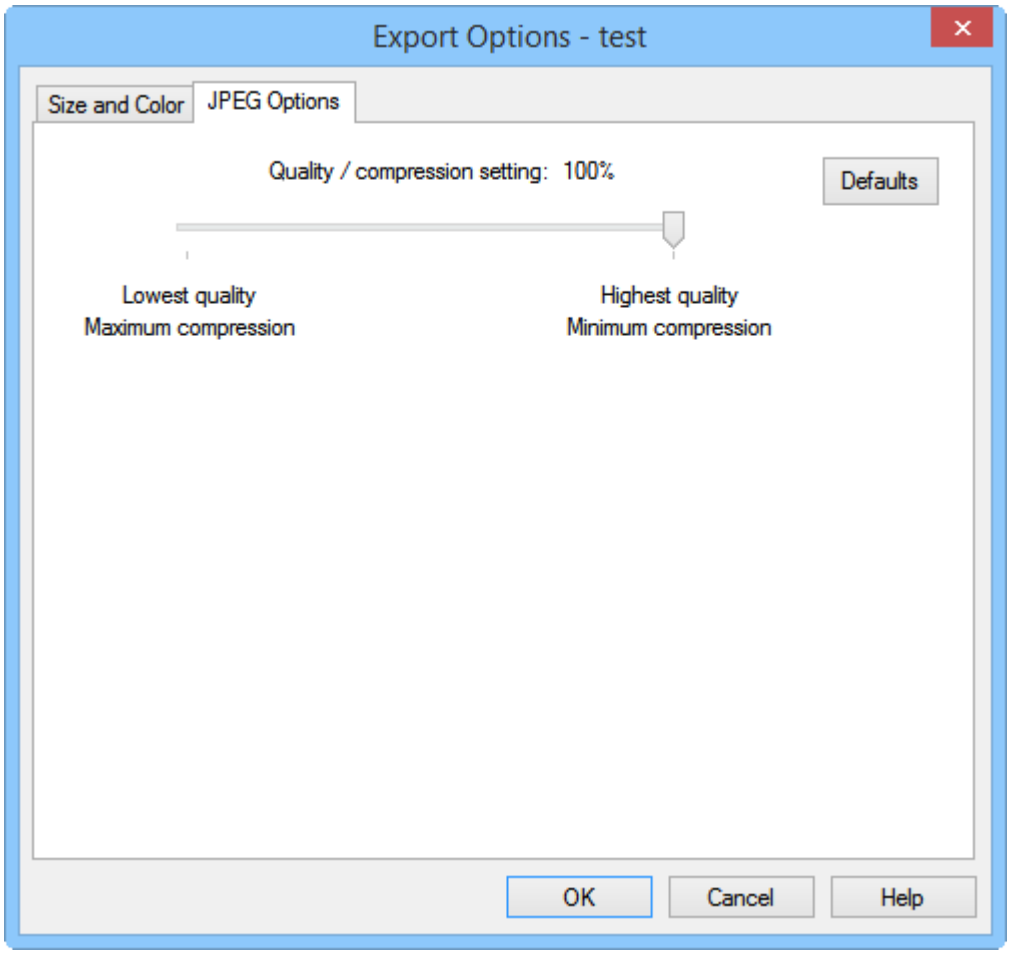
Click the *Cancel* button to exit the dialog without saving your changes or exporting the file.

JPEG Image Export Options Dialog

The **Export Options** dialog allows you to choose the quality of an exported .JPG image.

Export Options

This dialog appears when the **File | Export** command is chosen and *JPEG Compressed Bitmap (*.jpg, *. jpeg)* is selected as the output file type.



*Specify the quality of the exported JPEG Compressed Bitmap image in the **Export Options** dialog.*

Quality / Compression Setting

Adjust the slider for the desired compromise between image compression and image quality. Move the slider to the left for *Lowest quality/Maximum compression*. Move the slider to the right for *Highest quality/Minimum compression*. The greater the compression, the smaller the file size.

Defaults

Click the *Defaults* button to set all controls to their default values.

OK

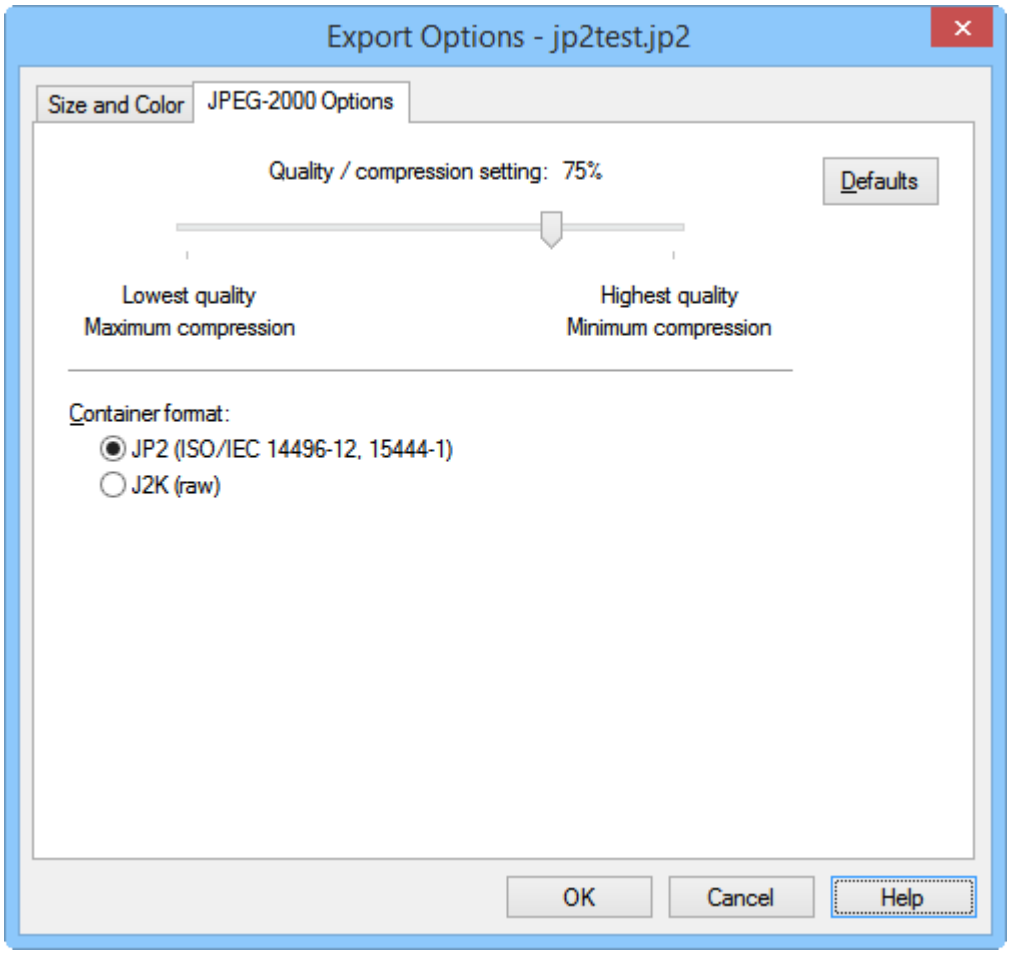
Click the *OK* button to save your changes, export the file, and close the dialog.

Cancel

Click the *Cancel* button to close the dialog without saving your changes or exporting the file.

Export Options Dialog - JPEG-2000 Options Page

The JPEG-2000 Options page is located in the Export Options dialog.



*Specify the Quality/Compression Setting on the **JPEG Options** page of the **Export Options** dialog.*

Quality / Compression Settings

Adjust the slider for the desired compromise between image compression and image quality. Move the slider to the left for *Lowest quality/Maximum compression.* Move the slider to the right for *Highest quality/Minimum compression.*

Container Format

The *Container format* controls the export format. Select *JP2 (ISO/IEC 14496-12, 15444-1)* to export the image in the ISO JP2 compliant format. This format exports the image data, and organization and contents of the file. Select *J2K (raw)* to export only the image data.

Defaults

The *Defaults* button sets all controls to their default values.

Iris Explorer .LAT Export Options Dialog

The **LAT Export Options** dialog controls determine how blank values are treated in the export file.

The LAT Export Options Dialog

The **LAT Export Options** dialog is only displayed if the lattice contains one or more blanked nodes. Select a data, lattice, or image module and choose the **File | Save Data** command to open the **LAT Export Options** dialog.



*Control how the blank nodes are exported in the **LAT Export Options** dialog.*

Set Any Blanked Nodes To

The controls in the *Set any blanked notes to* groupbox determine how blank values are treated in the export file.

Data Min

Choose *Data min* to export blank nodes with the indicated minimum value (varies depending on contents of lattice).

Data Max

Choose *Data max* to export blank nodes with the indicated maximum value (varies depending on contents of lattice).

Type Min

Choose *Type min* to export blank nodes with the indicated minimum value for the lattice's data type (usually zero).

Type Max

Choose *Type max* to export blank nodes with the indicated maximum value of the lattice's data type.

Fixed

Choose *Fixed* to export blank nodes with some other value, and enter the desired value in the adjacent edit control.

OK

Click the *OK* button to save your changes and export the file.

Cancel

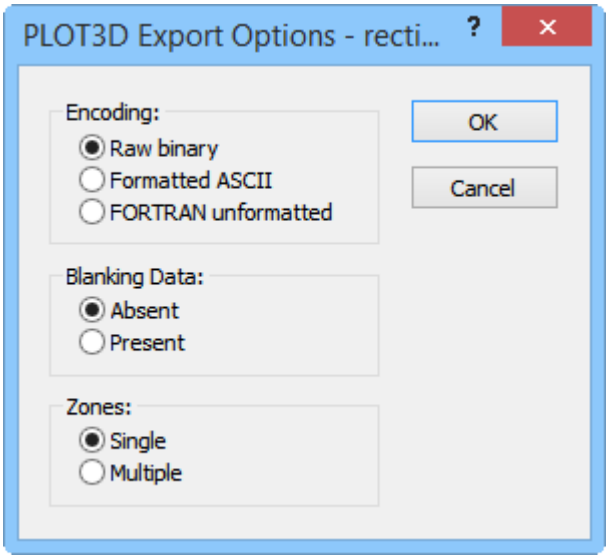
Click the *Cancel* button to exit the dialog without saving your changes or exporting the file.

PLOT3D Export Options Dialog

The **PLOT3D Export Options** dialog allows you to specify the encoding, presence or absence of blanking data, and zones of an exported .P3D or .XYZ file.

The PLOT3D Export Options Dialog

This dialog appears when the **File | Save Data** command is chosen and *PLOT-3D (*.p3d, *.xyz)* is selected as the output file type.



*Use the **PLOT3D Export Options** dialog to specify options for the exported .P3D or .XYZ file.*

Encoding

The *Encoding* option specifies how the data values are encoded in the Plot-3d files. Select *Raw binary*, *Formatted ASCII*, or *FORTRAN unformatted.*

Blanking Data

The *Blanking Data* option specifies whether blanking data is present in or absent from the exported file.

Zones

The *Zones* option specifies whether the data in the exported file is stored in single- or multiple-zone format.

OK

Click the *OK* button to save your changes and close the dialog.

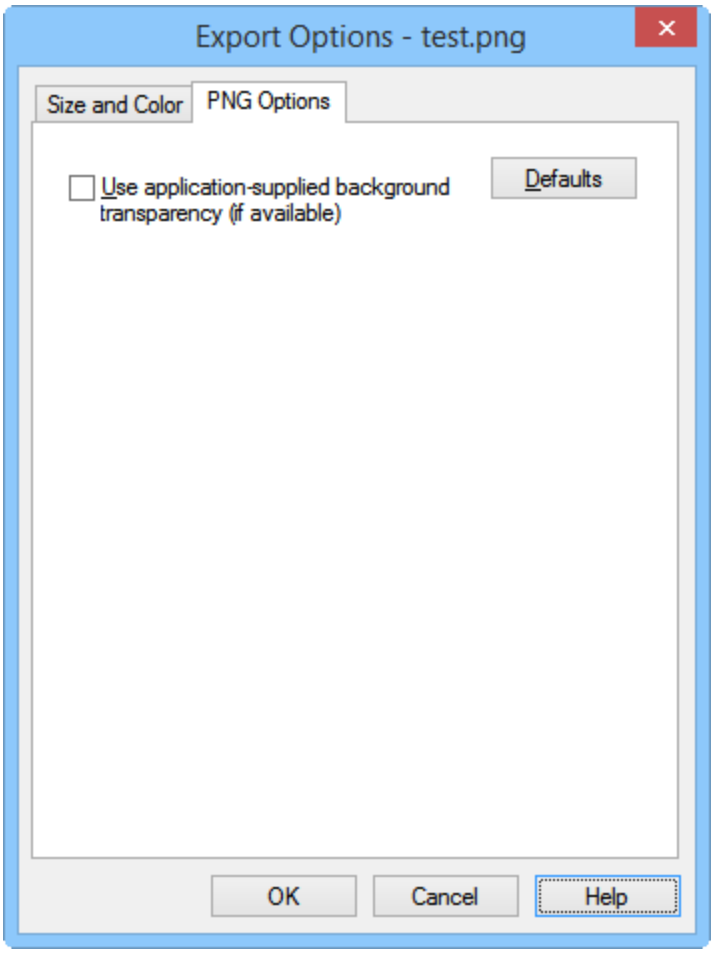
Cancel

Click the *Cancel* button to close the dialog without saving your changes.

Portable Network Graphic .PNG Export Options Dialog

The **PNG Options** page in the **Export Options** dialog controls how transparency is handled for the GIF export.

PNG Options



*Select whether to export the PNG image with application-supplied background transparency in the **PNG Options** page of the Export Options dialog*

Transparency

Check the *Use application-supplied background transparency (if applicable)* check box to export the image with a transparent background. To export the image with the transparent background, the *Color depth* on the **Size and Color** page of the **Export Options** dialog must be set to *4-bit color indexed*, *8-bit color indexed*, or *32-bit color w/ alpha*.

Defaults

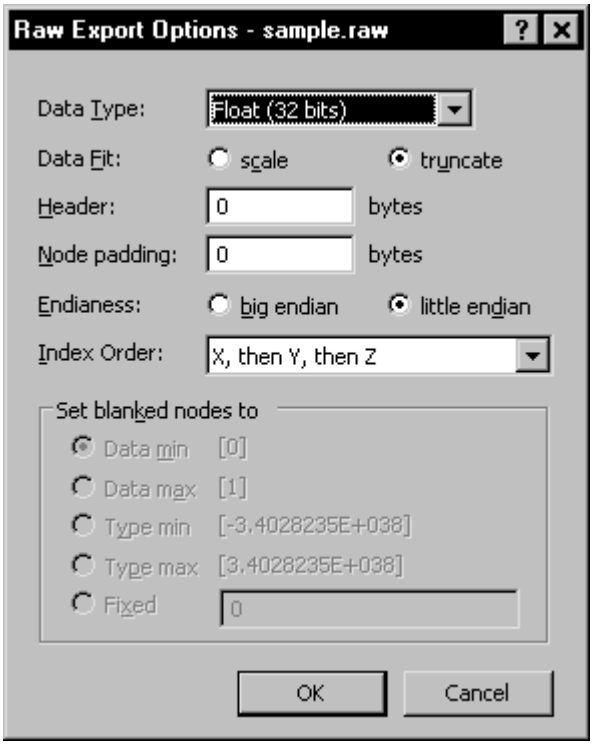
Click *Defaults* to return the options to their default values, i.e. *Transparency* is *None* and the *Custom color pixels transparent* color saturation values to 0.

Raw Export Options Dialog

The **Raw Export Options** dialog allows you to specify data type, data fit, header bytes, node padding, and endianess of the exported .RAW or .BIN data file.

The Raw Export Options Dialog

Select a lattice or image module and choose the **File | Save Data** command to open the **Raw Export Options** dialog.



*Use the **Raw Export Options** dialog to specify options for the exported .RAW or .BIN file.*

Data Type

The Data Type option allows the user to select the type of data that is written to the export file. The data type options include Single 8 bits, Unsigned 8 bits, Signed 6 bits, Unsigned 16 bits, Signed 32 bits, Unsigned 32 bits, Signed 64 bits, Float (32 bits), and Double (64 bits).

Data Fit

The *Data Fit* option allows the user to specify whether the export source data is scaled to truncated to fit the export data type chosen above. This is particularly important when the source data is larger, e.g., doubles, long integers, than the export data type, e.g., floats, short integers.

Header

The *Header* option allows you to specify how many bytes of non-data should be skipped at the beginning of the file before writing the first item of data. Enter a value in the *Header* box.

Node Padding

The *Node padding* option allows you to specify how many bytes of zeros should be written between each data item. This value is zero for most files. Enter a value in the *Node padding* box.

Endianess

The *Endianess* option allows you to specify whether the export data is in *little-endian* or *big-endian* byte order. Click the button to left of *little-endian* or *big-endian* to select a byte order. Integers are usually stored as a sequence of bytes, so that the encoded value can be obtained by simple concentration. The two most common are litte-endian and big-endian. Little-endian refers to increasing numeric significance with increasing memory addresses. Big-endian is the opposite and refers to "big end first".

Index Order

The *Index Order* option allows you to specify whether the precedence of the lattice data in the file is in X, Y, Z (X fastest) or Z, Y, X (Z fastest) order. The options are

X, then Y, then Z

Y, then X, then Z

Z, then X, then Y

X, then Z, then Y

Y, then Z, then X

Z, then Y, then X

Set Any Blanked Nodes To

The controls in the *Set any blanked notes to* groupbox determine how blank values are treated in the export file.

Data Min

Choose *Data min* to export blank nodes with the indicated minimum value (varies depending on contents of lattice).

Data Max

Choose *Data max* to export blank nodes with the indicated maximum value (varies depending on contents of lattice).

Type Min

Choose *Type min* to export blank nodes with the indicated minimum value for the lattice’s data type (usually zero).

Type Max

Choose *Type max* to export blank nodes with the indicated maximum value of the lattice’s data type.

Fixed

Choose *Fixed* to export blank nodes with some other value, and enter the desired value in the adjacent edit control.

OK

Click the *OK* button to save your changes and close the dialog.

Cancel

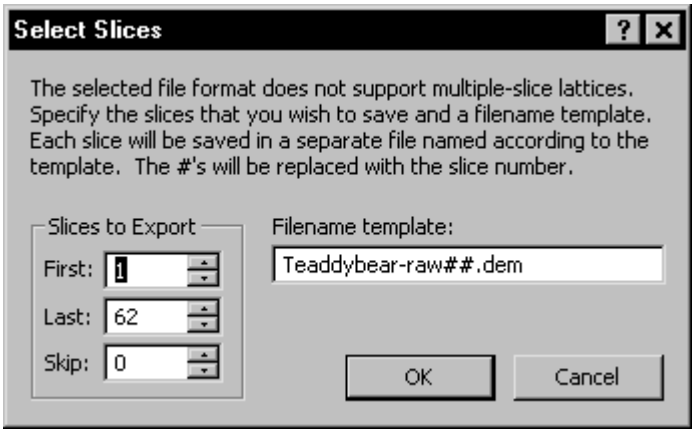
Click the *Cancel* button to exit the dialog without saving your changes or exporting the file.

Select Slices Dialog

The **Select Slices** dialog appears when a multi- slice uniform lattice is output to an image or DEM format. The output occurs as a series of slices in the Z direction.

Select Slices Dialogs

Select a multi- slice uniform lattice, choose the **File | Save Data** command, and select an image or DEM file format to open the **Select Slices** dialog.



Specify the range of the slices to export as individual files, making note of a slice number to skip (if any).

You are prompted for the range of slices to output and a file name template for use in adding the slice number to each generated file name.

OK

Click the *OK* button to save your changes and close the dialog.

Cancel

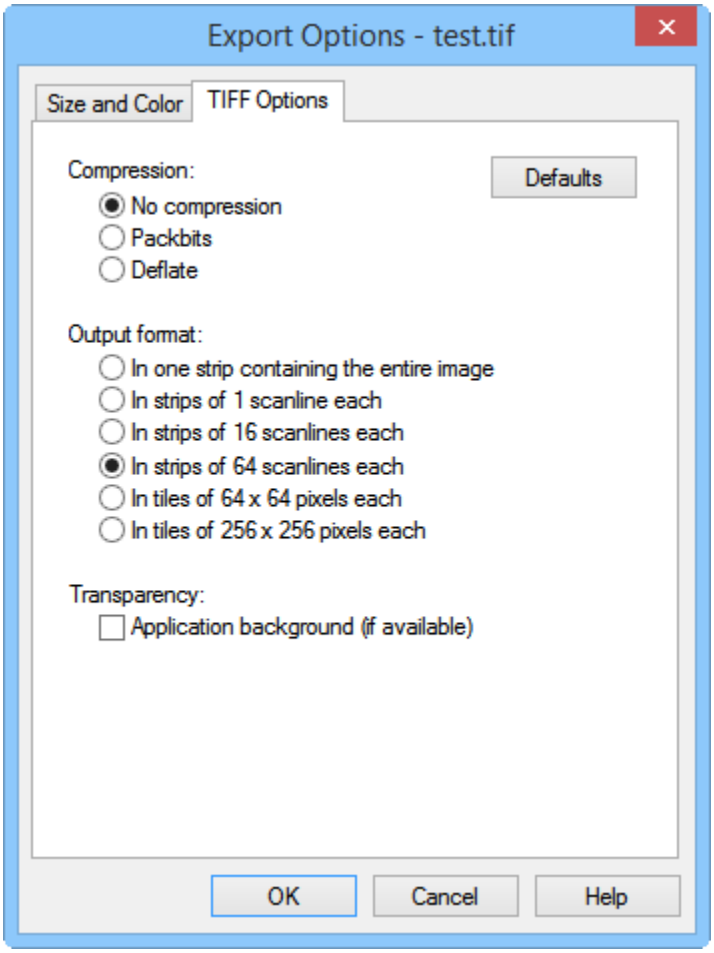
Click the *Cancel* button to close the dialog without saving your changes.

TIFF Image Export Options Dialog

The **Export Options** dialog allows you to choose the compression, output format, and background transparency of an exported .TIFF file.

Export Options Dialog

Choose the **File | Export** command or choose the **File | Save Data** command and select *TIF/TIFF Tagged Image (TIFF) (*. tif, *.tiff)* as the output file type to open the **TIFF Options** page of the **Export Options** dialog.



*Specify details of an exported .TIF file in the **Export Options** dialog.*

Compression

No compression results in a very large output file size. *Packbits* compression involves finding repeated data values; as a result, it is a good choice for images without large color ranges. *Deflate* is a dictionary encoding method that produces significant reduction in file size without losing any image information.

Output Format

Choose from a variety of *strip* and *tile* output formats. The option *In one strip containing the entire image* is recommended for most uses and is compatible with most software.

The *Output Format* options are:

In one strip containing the entire image

In strips of 1 scanline each

In strips of 16 scanlines each

In strips of 64 scanlines each

In tiles of 64 x 64 pixels each

In tiles of 256 x 256 pixels each

Multi-Plane Lattice Splitting

Check the *Split Multi-Plane Lattices* box to write output lattices as a series of single plane images (compatible with most software). Uncheck this box to write output lattices as single images with a TIFF IMAGEDEPTH tag greater than one (compatible with SGI image library software).

Transparency

Check the *Application background (if applicable)* check box to export the image with a transparent background. To export the image with the transparent background, the *Color depth* on the **Size and Color** page of the **Export Options** dialog must be set to *4-bit color indexed*, *8-bit color indexed*, or *32-bit color w/ alpha*.

OK

Click the *OK* button to save your changes, export the file, and close the dialog.

Cancel

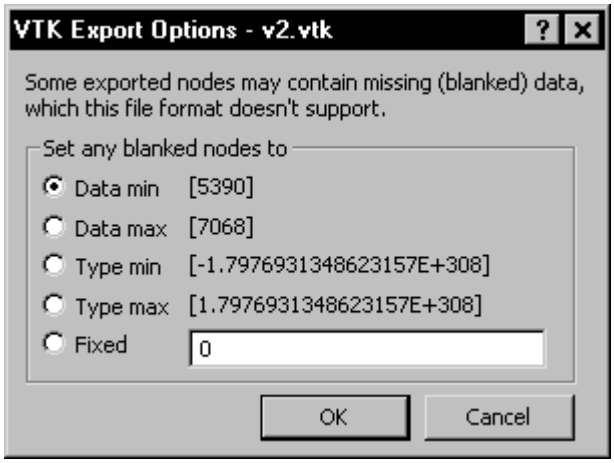
Click the *Cancel* button to close the dialog without saving your changes or exporting the file.

Visual Tool Kit .VTK Export Options Dialog

The **VTK Export Options** dialog determines how blank values are treated in the export file.

The VTK Export Options Dialog

The **VTK Export Options** dialog is only displayed if the lattice contains one or more blanked nodes. Select a module and choose the **File | Save Data** command to open the **VTK Export Options** dialog.



*Set blank node values for .VTK export in the **VTK Export Options** dialog.*

Set Any Blanked Nodes To

The controls in the *Set any blanked notes to* group determine how blank values are treated in the export file.

Data Min

Choose *Data min* to export blank nodes with the indicated minimum value (varies depending on contents of lattice).

Data Max

Choose *Data max* to export blank nodes with the indicated maximum value (varies depending on contents of lattice).

Type Min

Choose *Type min* to export blank nodes with the indicated minimum value for the lattice's data type (usually zero).

Type Max

Choose *Type max* to export blank nodes with the indicated maximum value of the lattice's data type.

Fixed

Choose *Fixed* to export blank nodes with some other value, and enter the desired value in the adjacent edit control.

OK

Click the *OK* button to save your changes and export the file.

Cancel

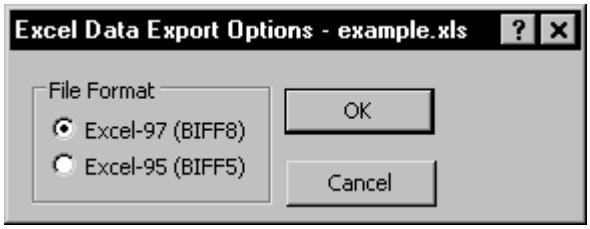
Click the *Cancel* button to exit the dialog without saving your changes or exporting the file.

Excel .XLS Export Options Dialog

The **Excel Data Export Options** dialog allows you to specify if the Excel file will be written in *Excel 97* or *Excel 95* format.

Excel Data Export Options Dialog

Select a data module and choose the **File | Save Data** command to open the **Excel Data Export Options** dialog.



*Select the File Format to use when exporting in the **XLS Import Options** dialog.*

File Format

Files can be saved in either *Excel-97 (BIFF8)* or *Excel-95 (BIFF5)* formats.

OK

Click the *OK* button to save your changes and export the file.

Cancel

Click the *Cancel* button to exit the dialog without saving your changes or exporting the file.

Use Caution when Saving Excel Files!

A file can be saved in an Excel format, but only one worksheet can be saved. The Excel export filter does not allow for saving multiple worksheets in a single Excel document. If a multi-worksheet Excel file is opened and saved as an .XLS file, be aware that only the single worksheet is saved in the document. If the existing file is overwritten, all the unused worksheets are destroyed. In this case, a warning message is issued. The message reads: *Saving this worksheet will destroy all but one of the sheets in the existing *.xls file.* To overwrite the file, click the *OK* button. To choose a different file name, click the *Cancel* button.

Automation Import Options

Automation Import File Types

For the automation *Import* command, the options parameter "Type" is defined by the strings on the left side of this table for the file types listed on the right side.

Type	File Formats
accdb	*.accdb
adf	*.adf
am	*.am, *.col
analyze	*.img
asc	*.asc, *.aig, *.agr, *.grd
avs	*.fld
avsx	*.x, *.ximg
bln	*.bln
bmp	*.bmp
bnd	*.bil, *.bip, *.bsq
cps	*.asc, *.dat, *.grd, *.cps, *.cps3
csv	*.csv
dat	*.dat
dbf	*.dbf
dem	*.dem
dicom	*.dic, *.dcm
dlg	*.dlg, *.lgo, *.lgs

dted	*.dt?
dxf	*.dxf
ers	*.ers
etopo5	*.dos, *.dat
e00	*.e00
flt	*.flt
geo	*.grd, *.ggf
gif	*.gif
globe	?10g
gsb	*.gsb
gsi	*.gsi
gtopo30	*.hdr
gxf	*.gxf
hdf	*.hdf
iv	*.iv
jpg	*.jpg, *.jpeg
lat	*.lat
leica	*.info
mdb	*.mdb
mif	*.mif
nema	*.an1, *.an2
pcx	*.pcx
plt	*.plt
ply	*.ply
png	*.png
pnm	*.pnm, *.ppm, *.pgm, *.pbm
p3d	*.p3d, *.xyz
raw	*.raw, *.bin
rst	*.rst, *.img
sdtstvp	*.ddf
sdtsdem	*.ddf
segy	*.sgy, *.segy
sgi	*.rgb, *.rgba, *.bw
sgrd	*.grd
shp	*.shp
sid	*.sid
sun	*.ras, *.sun
slk	*.slk

stacked	*.asi
stk	*.stk
tga	*.tga
tif	*.tif, *.tiff
txt	*.txt
vdat	*.vdat
vtk	*.vtk
xls	*.xls
xlsx	*.xlsx
zmap	*.asc, *.dat, *.grd, *.xyz, *.zmap, *.zyc, *.zycor

Used by: Import

Esri ArcInfo Binary Grid .ADF Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	ADF

Example

This example shows how to import an ADF file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "adf")
  CommandApi.Option("Path", "C:/Path/To/Import/file.adf")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Microsoft Access .ACCDB Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	ACCDB
SQLSTR	string	SQL string indicating selection
Table	integer	Numeric value of the table to import

Example

This example shows how to import an Access ACCDB file with the various options.

```
CommandApi.Construct("Import")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("AutoConnect", "True")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "accdb")
CommandApi.Option("Options", "SQLSTR=; Table=2")
CommandApi.Option("Path", "C:/Path/To/Import/file.accdb")
CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Amira Mesh .AM, .COL Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	AM

Example 1

This example shows how to import an AM file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "am")
```

```
  CommandApi.Option("Path", "C:/Path/To/Import/file.am")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Example 2

This example shows how to import an COL file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "am")
  CommandApi.Option("Path", "C:/Path/To/Import/file.col")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Analyze 7.5 Medical Image .IMG Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	analyze

Example

This example shows how to import an IMG file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "analyze")
  CommandApi.Option("Path", "C:/Path/To/Import/file.img")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Arc/Info ASCII Grid File .ASC, .AIG, .AGR, .GRD Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	ASC

Example 1

This example shows how to import an Arc/Info ASC grid file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "asc")
  CommandApi.Option("Path", "C:/Path/To/Import/file.asc")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Example 2

This example shows how to import an Arc/Info AIG grid file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "asc")
  CommandApi.Option("Path", "C:/Path/To/Import/file.aig")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Example 3

This example shows how to import an Arc/Info AGR grid file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "asc")
  CommandApi.Option("Path", "C:/Path/To/Import/file.agr")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Example 4

This example shows how to import an Arc/Info GRD grid file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "asc")
  CommandApi.Option("Path", "C:/Path/To/Import/file.grd")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

AVS Field .FLD Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	AVS

Example

This example shows how to import an Arc ASC grid file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
```

```
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "avs")
  CommandApi.Option("Path", "C:/Path/To/Import/file.fld")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

AVSX .X, .XIMG Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	AVSX

Example 1

This example shows how to import an AVS XIMG file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "avsx")
  CommandApi.Option("Path", "C:/Path/To/Import/file.ximg")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Example 2

This example shows how to import an AVS X file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "avsx")
  CommandApi.Option("Path", "C:/Path/To/Import/file.x")
```

```
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Golden Software Blanking .BLN Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	BLN

Example

This example shows how to import a Golden Software BLN file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "bln")
  CommandApi.Option("Path", "C:/Path/To/Import/file.bln")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Bitmap .BMP Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	BMP

Example

This example shows how to import a BMP file with the various options.

```
CommandApi.Construct("Import")
```

```
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "bmp")
  CommandApi.Option("Path", "C:/Path/To/Import/file.bmp")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Atlas Boundary [.BNA] Import Automation Options

Since the **Import Options** dialog is not displayed when the program is driven from an automation script, an options string can be specified in the script. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	BNA
AreasToCurves	0 or 1	If set to 1, area objects are converted to closed curves.

Example 1

This example shows how to import a BNA file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "bna")
  CommandApi.Option("Options", "AreasToCurves=0")
  CommandApi.Option("Path", "C:/Path/To/Import/file.bna")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Band Interleaved .BIL, .BIP, .BSQ Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	BND

Example 1

This example shows how to import a BIL file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "bnd")
  CommandApi.Option("Path", "C:/Path/To/Import/file.bil")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Example 2

This example shows how to import a BIP file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "bnd")
  CommandApi.Option("Path", "C:/Path/To/Import/file.bip")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Example 3

This example shows how to import a BSQ file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
```

```
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "bnd")
  CommandApi.Option("Path", "C:/Path/To/Import/file.bsq")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

CPS-3 Grid .ASC, .ADX, .DAT, .GRD, .CPS, .CPS3 Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	CPS

Example 1

This example shows how to import a CPS3 ASC file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "cps")
  CommandApi.Option("Path", "C:/Path/To/Import/file.asc")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Example 2

This example shows how to import a CPS3 file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "cps")
  CommandApi.Option("Path", "C:/Path/To/Import/file.cps3")
```

```
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Example 3
This example shows how to import a CPS3 DAT file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "cps")
  CommandApi.Option("Path", "C:/Path/To/Import/file.dat")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

.CSV Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	CSV

Example
This example shows how to import a CSV file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "csv")
  CommandApi.Option("Path", "C:/Path/To/Import/file.csv")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

.DAT Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	DAT
ColumnBreaks	Comma separated Integers	Set of integers telling Voxler where the column breaks are. These integers will be separated by commas. Only use if UseFixedWidth = 1. Example: ColumnBreaks=9,18,27
DateOrder	0 = Auto 1 = MDY 2 = DMY 3 = YMD 4 = MYD 5 = DYM 6 = YDM	Integer that specifies the order in which dates are written in the data file. When set to 0, the standard date order from the *Locale* is used.
DecimalSymbol	period = period comma = comma	String that specifies the character that separates the integer and fractional components of the data values.
Delimiter	string	String which tells Voxler which delimiter to use to separate each column. Voxler recognises *space*, *tab*, *comma*, *semicolon* as strings, otherwise use the literal string. Only use if UseFixedWidth = 0. Example: Delimiter=(Delimiter=comma
EatWhiteSpace	1 or 0	0 does not skip extra spaces between values. 1 skips extra spaces and tabs between values.
ImportCodePage	Integer	The number of the ANSI code page to use when importing Unicode data. Valid code page number in the range 0 through 65535.
Locale	String	Locale ID in decimal values. The default locale is determined by the locale setting in the Windows Control Panel.
SkipExtraDelimiters	1 or 0	0 skips extra delimiters. Multiple delimiters are treated as a single

		delimiter. 1 makes each delimiter a separate delimiter.
StartRow	Integer	The row to start importing from. Everything before this row will be discarded.
TextQualifier	string	Tells Voxler what quote characters to use. Options are *singlequote*, *doublequote*, or *none*.
UseFixedWidth	1 or 0	1 uses a fixed width column. 0 uses a delimited column.

Example 1

This example shows how to import a DAT file with the various options in a single option string.

```
CommandApi.Construct("Import")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ProgressEnabled", "False")
CommandApi.Option("UndoRedoEnabled", "True")
CommandApi.Option("AutoConnect", "True")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "dat")
CommandApi.Option("Options", "AggressiveConversion=0; Delimiter=space,tab,comma,semicolon;
EatWhitespace=1; SkipExtraDelimiters=0; StartRow=0; TextQualifier=doublequote,quote;
UseFixedWidth=0")
CommandApi.Option("Path", "C:/Path/To/Import/file.dat")
CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Example 2

This example shows how to import a DAT file with the various options on multiple option strings.

```
CommandApi.Construct("Import")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ProgressEnabled", "False")
CommandApi.Option("UndoRedoEnabled", "True")
CommandApi.Option("AutoConnect", "True")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "dat")
CommandApi.Option("Options", "AggressiveConversion=0")
CommandApi.Option("Options", "Delimiter=space,tab,comma,semicolon")
CommandApi.Option("Options", "EatWhitespace=1")
CommandApi.Option("Options", "SkipExtraDelimiters=0")
CommandApi.Option("Options", "StartRow=0")
CommandApi.Option("Options", "TextQualifier=doublequote,quote")
CommandApi.Option("Options", "UseFixedWidth=0")
```

```
CommandApi.Option("Path", "C:/Path/To/Import/file.dat")
CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

ASCII Database .DBF Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	DBF

Example

This example shows how to import a DBF file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "dbf")
  CommandApi.Option("Path", "C:/Path/To/Import/file.dbf")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

USGS Digital Elevation Model .DEM Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	DEM

Example

This example shows how to import a DBF file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "dem")
  CommandApi.Option("Path", "C:/Path/To/Import/file.dem")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

DICOM Medical Image .DIC, .DCM, .AN1, .AN2 Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	DICOM
AutoContrast	1 or 0	Detect and set appropriate contrasts to try and improve visibility of a dic image. 0 does not use Voxler's autocontrast feature. 1 does.

Example

This example shows how to import a DIC file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "True")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "dicom")
  CommandApi.Option("Options", "AutoContrast=1")
  CommandApi.Option("Path", "C:/Path/To/Import/file.dic")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Example
This example shows how to import a DCM file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "True")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "dicom")
  CommandApi.Option("Options", "AutoContrast=1")
  CommandApi.Option("Path", "C:/Path/To/Import/file.dcm")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

USGS Digital Line Graph .DLG, .LGO, .LGS Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	DLG
AllAreas	1 or 0	Whether to consider all images when processing the dlg file. 1 does not use all images. 0 does.
AllLines	1 or 0	Whether to consider all lines when processing the dlg file. 1 does not use all lines. 0 does.
AllNodes	1 or 0	Whether to consider all nodes when processing the dlg file. 1 does not use all nodes. 0 does.
AreasToCurves	1 or 0	Whether to convert areas to closed curves. 1 does not convert areas to curves. 0 does.
AttribSelListBox	1 or 0	Whether to use an attribute list or a range selection. 1 uses a range selection. 0 uses an attribute list.

Deviation	Number	Set the deviation angle to any number. Default to 0.0.
FreeLines	1 or 0	Whether to consider only free-standing lines when processing the dlg file. This is related to the *AllLines* option. 1 does not use free standing lines. 0 does.
FreeNodes	1 or 0	whether to consider only free-standing nodes when processing the dlg file. This is related to the *AllNodes* option. 1 does not use free standing nodes. 0 does.
IDSel	Blank or range of integers	Only consider entities with IDs in the specified range. Default is to leave it blank.
MajorSel	Blank or range of integers	Only consider entities with Major Attributes numbered within the range. Default is to leave it blank.
MinorSel	Blank or range of integers	Only consider entities with Minor Attributes numbered within the range. Default is to leave it blank.
NoMapFrame	1 or 0	Whether to consider the area that makes up the map frame when processing the dlg file. 1 does use the map frame. 0 does not.
Projection	0, 1, 2	0 = coordinates are UTM or Albers Equal Area projected. This is the default. 1 = coordinates are converted to Lat/Long. 2 = coordinates are not projected.
Synthesize	1 or 0	Whether to synthesize the IDs for nodes, areas and lines. 0 does not synthesize the IDs. 1 does.
ThinAuto	1 or 0	Whether to automatically thin the vertices. 0 does not

		automatically thin. 1 does.

Example 1
This example shows how to import a USGS LGS file with the various options on single line.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "True")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "dlg")
  CommandApi.Option("Options", "AllAreas=4;AllLines=0;AllNodes=0;AreasToCurves=8;_
  AttribSelListBox=0;Deviation=0.0;FreeLines=32;FreeNodes=0;IDSel=;MajorSel=;MinorSel=;_
  NoMapFrame=2048;Projection=0;Synthesize=0;ThinAuto=1024")
  CommandApi.Option("Path", "C:/Path/To/Import/file.lgs")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Example 1
This example shows how to import a USGS DLG file with the various options on multiple lines.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "True")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "dlg")
  CommandApi.Option("Options", "AllAreas=4")
  CommandApi.Option("Options","AllLines=0")
  CommandApi.Option("Options", "AllNodes=0")
  CommandApi.Option("Options", "AreasToCurves=8")
  CommandApi.Option("Options", "AttribSelListBox=0")
  CommandApi.Option("Options", "Deviation=0.0")
  CommandApi.Option("Options", "FreeLines=32")
  CommandApi.Option("Options", "FreeNodes=0")
  CommandApi.Option("Options", "IDSel=")
  CommandApi.Option("Options", "MajorSel=")
  CommandApi.Option("Options", "MinorSel=")
  CommandApi.Option("Options", "NoMapFrame=2048")
  CommandApi.Option("Options", "Projection=0")
```

```
  CommandApi.Option("Options", "Synthesize=0")
  CommandApi.Option("Options", "ThinAuto=1024")
CommandApi.Option("Path", "C:/Path/To/Import/file.dlg")
CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

DTED .DT0, .DT1, .DT2 Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	DTED

Example

This example shows how to import a DTED file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "dted")
  CommandApi.Option("Path", "C:/Path/To/Import/file.dt")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

AutoCAD .DXF Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	DXF
ApplyViewAngle	1 or 0	Apply the viewing angle, if there is one, to the image.

		1 applies the view angle. 0 does not.
SkipPaperspace	1 or 0	Import any paperspace entities in the dxf drawing. 1 imports the paper space objects. 0 does not.

Example

This example shows how to import a DXF file with the various options.

```
CommandApi.Construct("Import")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("Filter", "dxf")
CommandApi.Option("Options", "ApplyViewAngle=1;SkipPaperspace=0")
CommandApi.Option("Path", "C:/Path/To/Import/file.dxf")
CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Esri ArcInfo Export Format .E00 Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	E00
AreasToCurves	1 or 0	Whether to set all polygons to polylines. 0 does not set all polygons to polylines. 1 does.
WhichAreas	1 or 0	Whether to import all areas. 0 imports all areas. 1 imports none.
WhichLines	0, 1, 2	Whether to import all lines. 0 imports all lines. 1 imports freestanding lines. 2 imports no lines.
WhichText	-2, -1, or positive integer	-2 imports all text. -1 imports no text. Positive integers tells Voxler which text group to import.

Example

This example shows how to import an E00 file with the various options.

```
CommandApi.Construct("Import")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("Filter", "e00")
CommandApi.Option("Options",
"ApplyViewAngle=1;AreasToCurves=0;ProjectionLatLong=1;SkipPaperspace=0;WhichAreas=0;Whi
chLines=0;WhichNodes=0;WhichText=1")
CommandApi.Option("Path", "C:/Path/To/Import/file.e00")
CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

ER Mapper [.ECW] Import Automation Options

Since the **Import Options** dialog is not displayed when the program is driven from an automation script, an options string can be specified in the script. The string consists of comma-separated parameters, which specify the behavior of the various import options

Option	Action	Default	Description
Scale	1 = 1/1 (uses most memory, best quality) 2 = 1/2 4 = 1/4 8 = 1/8 16 = 1/16 32 = 1/32 (uses least memory, lowest quality)	1	Since some images can be extremely large once expanded into memory, the import filter allows the image to be reduced in dimensions by 1/2 to 1/32 of the original size.
LRX	integer	maximum X	Use LRX to limit the import to a subregion of the file. LRX is the X pixel coordinate for the lower right corner of the desired subregion.
LRY	integer	maximum Y	Use LRY to limit the import to a subregion of the file. LRY is the Y pixel coordinate for the lower right corner of the desired subregion.
ULX	integer	0	Use ULX to limit the import to a subregion of the file. ULX is the X pixel coordinate for the upper left corner of the desired subregion.
ULY	integer	0	Use ULY to limit the import to a subregion of the file. ULY is the Y pixel coordinate for the upper left corner of the desired subregion.

Remarks

The pixel coordinates increase in the X direction from left to right and increase in the Y direction from top to bottom. The pixel coordinate origin is the upper left corner. For example an image with a 2000 pixel width and 3000 pixel height has the following coordinates: the upper left corner is coordinate (0,0) and the lower right corner is the x and y maximums (2000,3000).

However, the image still will import correctly even if the ULY is used for the maximum Y and LRY is used for the minimum Y.

ER Mapper Grid .ERS Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	ERS

Example

This example shows how to import an ERS file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "ers")
  CommandApi.Option("Path", "C:/Path/To/Import/file.ers")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

USGS ETOPO5 .DOS, .DAT Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	ETOPO5

Example

This example shows how to import an ETOPO5 DOS file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "etopo5")
  CommandApi.Option("Path", "C:/Path/To/Import/file.dos")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Esri Binary Float Grid .FLT, .HDR Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	FLT

Example

This example shows how to import a FLT file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "flt")
  CommandApi.Option("Path", "C:/Path/To/Import/file.flt")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Geosoft Binary Grid .GRD, .GGF Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	GEO

Example 1

This example shows how to import a Geosoft GRD file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "geo")
  CommandApi.Option("Path", "C:/Path/To/Import/file.grd")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Example 2

This example shows how to import a Geosoft GGF file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "geo")
  CommandApi.Option("Path", "C:/Path/To/Import/file.ggf")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Graphics Interchange Format .GIF Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	GIF

Example

This example shows how to import a GIF file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "gif")
  CommandApi.Option("Path", "C:/Path/To/Import/file.gif")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

GLOBE DEM Data .?10g Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	GLOBE

Example

This example shows how to import a Globe DEM 10G file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "globe")
  CommandApi.Option("Path", "C:/Path/To/Import/file.10g")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

GRIB Global Weather Data Grid Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Option	Action	Default	Description
Filter	string		GRIB
WhichGrid	-1 = Import all grids Integer = grid number	""	This is the desired grid's sequence number within the file. 0 = first grid, 1 = second grid, etc.

Golden Software Boundary .GSB Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	GSB

Example

This example shows how to import a Golden Software GSB file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "gsb")
  CommandApi.Option("Path", "C:/Path/To/Import/file.gsb")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Golden Software Interchange .GSI Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	GSI

Example

This example shows how to import a Golden Software GSI file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "gsi")
  CommandApi.Option("Path", "C:/Path/To/Import/file.gsi")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

GTopo30 .HDR, .STX, .DEM Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	GTOPO30

Example

This example shows how to import a GTOPO30 HDR file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "gtopo30")
  CommandApi.Option("Path", "C:/Path/To/Import/file.hdr")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Grid Exchange .GXF Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	GXF

Example

This example shows how to import a GXF file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "gxf")
  CommandApi.Option("Path", "C:/Path/To/Import/file.gxf")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Hierarchical Data Format .HDF Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	HDF

Example

This example shows how to import a HDF file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
```

```
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "hdf")
  CommandApi.Option("Path", "C:/Path/To/Import/file.hdf")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Inventor .IV Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	IV

Example

This example shows how to import a IV file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "iv")
  CommandApi.Option("Path", "C:/Path/To/Import/file.iv")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

JPEG Compressed Bitmap .JPG, .JPEG Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	JPG

Example

This example shows how to import a JPG file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "jpg")
  CommandApi.Option("Path", "C:/Path/To/Import/file.jpeg")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

JPEG-2000 File Interchange Format .JP2 Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	JPG2000

Example

This example shows how to import a JP2 file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "jpg2000")
  CommandApi.Option("Path", "C:/Path/To/Import/file.jp2")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

.LAS Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	LAS
CreateModule	1 or 0	If 1, create a new module. If 0, add to an existing module (use *TargetName)*.
NullValue	Number	The null value to use in the logs.
OverwriteLogs	1 or 0	If 1, overwrite any logs with duplicate names. If 0, use unique naming to append logs.
SetWellTop	1 or 0	If 1, use the well top passed in the *WellTop* field. If 0, set the well top automatically if possible or leave undefined.
TargetName	string	The name of the existing module to add data to, used when *CreateModule* is 0.
WellName	string	The name of the well to create or the name of the existing well to add new data to.
WellTop	XYZ point	The optional top value to assign to a newly created well or to an existing well if the current top information is invalid.
Logs	string list	A comma delimited collection of all logs.
SelectedLogs	string list	A comma delimited collection of all logs to import.

Example

This example shows how to import a LAS file with the various options.

```
CommandApi.Construct("Import")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("AutoConnect", "True")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Path", "C:/Path/To/Import/file.las")
CommandApi.Option("Filter", "las")
CommandApi.Option ("CreateModule", "1")
```

```
CommandApi.Option ("NullValue", "15")
CommandApi.Option ("OverwriteLogs", "1")
CommandApi.Option ("SetWellTop", "1")
CommandApi.Option ("WellName", "New_Well")
CommandApi.Option ("WellTop","15,20,25")
CommandApi.Option ("SelectedLogs", "ILM, NPHI,RHOB,SP")
CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

LAS LiDAR Binary File Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Option	Action	Default	Description
Filter	string		LIDARP; LIDARW
RejectInconsistent	0 = No 1 = Yes	0	1 to reject inconsistent records, 0 to keep inconsistent records.
AcceptXMin	Value		Minimum X value, greater than or equal to
AcceptXMax	Value		Maximum X value, less than or equal to
AcceptYMin	Value		Minimum Y value, greater than or equal to
AcceptYMax	Value		Maximum Y value, less than or equal to
AcceptZMin	Value		Minimum Z value, greater than or equal to
AcceptZMax	Value		Maximum Z value, less than or equal to
AcceptNthPoint	Value		Filters out every Nth point.
AcceptReturn	Value		Filters by laser return value
AcceptClass	0 = No 1 = Yes		0 rejects the specified class, and 1 accepts the specified class. Takes the form "AcceptClass[x]=" where x is class number (0-31)
AcceptAllClasses	0 = No 1 = Yes		1 to accept ALL classes. Overrides AcceptClass specification.
AcceptNoClasses	0 = No 1 = Yes		1 to accept NO classes. Overrides AcceptClass specification.

Example

This example shows how to import a LiDAR LAS file with various options.

```
CommandApi.Construct("Import")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("AutoConnect", "True")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Path", "C:/Path/To/Import/file.las")
CommandApi.Option("Filter", "lidarp")
CommandApi.Option ("Options", "RejectInconsistent=1; AcceptNthPoint=2; AcceptClass[3]=0")
CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Iris Explorer .LAT Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	LAT

Example

This example shows how to import a LAT file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "lat")
  CommandApi.Option("Path", "C:/Path/To/Import/file.lat")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Leica Confocal Scanning .INFO Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description

Filter	string	LEICA

Example

This example shows how to import a Leica INFO file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "leica")
  CommandApi.Option("Path", "C:/Path/To/Import/file.info")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Microsoft Access .MDB Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	MDB
SQLSTR	string	SQL string indicating selection
Table	integer	Numeric value of the table to import

Example

This example shows how to import an Access MDB file with the various options.

```
CommandApi.Construct("Import")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("AutoConnect", "True")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "mdb")
CommandApi.Option("Options", "SQLSTR=; Table=2")
CommandApi.Option("Path", "C:/Path/To/Import/file.mdb")
CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

MapInfo Interchange Format .MIF Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	MIF
AreasToCurves	1 or 0	Whether to set all polygons to polylines. 0 does not set all polygons to polylines. 1 does.
IgnoreStyles	1 or 0	Whether to import without colors/fills/patterns etc. 0 does not import without the colors, fills, patterns, etc. 1 does import without these.

Example

This example shows how to import a MapInfo MIF file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("AutoConnect", "True")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "mif")
  CommandApi.Option("Options", "AreasToCurves=1;IgnoreStyles=1")
  CommandApi.Option("Path", "C:/Path/To/Import/file.mif")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

DICOM Medical Image .DIC, .DCM, .AN1, .AN2 Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	NEMA

Example

This example shows how to import a NEMA AN1 file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "nema")
  CommandApi.Option("Path", "C:/Path/To/Import/file.an1")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

NetCDF .NC File Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Option	Default	Description
VariableName	""	Set to the variable name to import. The variable name must match exactly with the variable name listed in the file.
RecordIndex	""	Set to the record index to import for variable names with records.

PLOT-3D .P3D, .XYZ Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	P3D
DataFormat	0, 1, 2	0 = Data is in binary format 1 = Data is formatted. 2 = Data is unformatted.
GridFormat	0, 1, 2, 3	0 = The data is structured 1 = The data is unstructured 2 = The data is in scalar format.

		3 = The data is in vector format.
Multizone	0, 1	0= Is not multizone. 1 = Is multizone.
Blanked	0, 1	0 = Do not use blanking. 1 = Do use blanking.

Example

This example shows how to import a P3D file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "p3d")
  CommandApi.Option("Options", "DataFormat=0;GridFormat=1;Multizone=0;Blanked=0")
  CommandApi.Option("Path", "C:/Path/To/Import/file.p3d")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

ZSoft Paintbrush .PCX Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	PCX

Example

This example shows how to import a PCX file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "pcx")
```

```
  CommandApi.Option("Path", "C:/Path/To/Import/file.pcx")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

PDF Adobe Import Automation Options

Since the **Import Options** dialog is not displayed when the program is driven from an automation script, an options string can be specified in the script. The string consists of comma-separated parameters, which specify the behavior of the various import options. A typical example would be:

```
 "Resolution=300, WhichPage=7"
```

This would set the resolution of the images in the PDF file to 300 DPI and imports only page 7.

Option	Action	Default	Description
Filter	string		PDF
WhichPages	0 = all pages > 1 = specific page	0	The page number of the page to import. 0 imports all pages. Type the specific page number to import only that page.
Resolution	Value from 50 to 1200	150	The resolution of the imported PDF in DPI.
AntialiasVector	0 = false 1 = true	1	Turns on/off the smoothing of vector objects in the imported PDF file, such as lines and polygons.
AntialiasRaster	0 = false 1 = true	1	Turns on/off the smoothing of text and images in the imported PDF file.

Example

This example shows how to import a PDF file with the various options.

```
CommandApi.Construct("Import")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "pdf")
CommandApi.Option("Options", "WhichPages=2; Resolution=300; AntialiasVector=0;_
        AntialiasRaster=1")
CommandApi.Option("Path", "C:/Path/To/Export/file.pdf")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "0")
CommandApi.Do()
```

Golden Software PlotCall .PLT Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	PLT
AreasToCurves	1 or 0	Whether to set all polygons to polylines. 0 does not set all polygons to polylines. 1 does.

Example

This example shows how to import a Golden Software PLT file with the various options.

```
CommandApi.Construct("Import")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("AutoConnect", "True")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "plt")
CommandApi.Option("Options", "AreasToCurves=1")
CommandApi.Option("Path", "C:/Path/To/Import/file.plt")
CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Stanford Polygon .PLY Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	PLY

Example

This example shows how to import a PLY file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
```

```
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "ply")
  CommandApi.Option("Path", "C:/Path/To/Import/file.ply")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Portable Network Graphic .PNG Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	PNG

Example

This example shows how to import a PNG file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "png")
  CommandApi.Option("Path", "C:/Path/To/Import/file.png")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Portable Bitmap .PBM, .PGM, .PPM, .PNM Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	PNM

Example

This example shows how to import a PNM file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "pnm")
  CommandApi.Option("Path", "C:/Path/To/Import/file.pnm")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

RAW Binary Lattice .RAW, .BIN Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	RAW
BigEndian	0,1	Whether to assume the data is in Motorola MSB format. 0 is in Intel LSB format. 1 is in Motorola MSB format.
DataType	0, 1, 2, 3, 4, 5, 6, 7, 8	0 -- char 1 -- BYTE 2 -- short 3 -- unsigned short 4 -- int 5 -- unsigned int 6 -- __int64 7 -- float 8 -- double
DimX	integer	Number of X columns in lattice.
DimY	integer	Number of Y columns in lattice.
DimZ	integer	Number of Z columns in lattice.

FitByScaling	0, 1	0 = Do not scale data values of different types to fit the requested data types; also set if data values are truncated. 1 = scales data of different types to fit the requested data structure.
HasSpacing	0, 1	0 = Do not treat Raw_Min and Raw_Spacing as valid. 1 = Do treat Raw_Min and Raw_Spacing as valid.
IndexOrder	0,.1, 2, 4, 5, 6	Indexing order defined by: 0="X, then Y, then Z" 1="Y, then X, then Z" 2="Z, then X, then Y" 4="X, then Z, then Y" 5="Y, then Z, then X" 6="Z, then Y, then X"
ItemsPerNode	integer	Number of data elements for each node
SkipAfter	integer	Number of bytes to skip after each node.
StartOffset	integer	Offset of first data node in file.
XMax	integer	Largest X value in grid.
XMin	integer	Smallest X value in grid.
XSpacing	integer	Spacing between X nodes in grid.
YMax	integer	Largest Y value in grid.
YMin	integer	Smallest Y value in grid.
YSpacing	integer	Spacing between Y nodes in grid.
ZMax	integer	Largest Z value in grid.
ZMin	integer	Smallest Z value in grid.
ZSpacing	integer	Spacing between Z nodes in grid.

Example

This example shows how to import a RAW file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
```

```
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "raw")
  CommandApi.Option("Options", "BigEndian=0")
  CommandApi.Option("Options", "DataType=1")
  CommandApi.Option("Options", "DimX=2")
  CommandApi.Option("Options", "DimY=2")
  CommandApi.Option("Options", "DimZ=1")
  CommandApi.Option("Options", "FitByScaling=0")
  CommandApi.Option("Options", "HasSpacing=1")
  CommandApi.Option("Options", "IndexOrder=0")
  CommandApi.Option("Options", "ItemsPerNode=1")
  CommandApi.Option("Options", "SkipAfter=0")
  CommandApi.Option("Options", "StartOffset=0")
  CommandApi.Option("Options", "XMax=1")
  CommandApi.Option("Options", "XMin=0")
  CommandApi.Option("Options", "XSpacing=1")
  CommandApi.Option("Options", "YMax=1")
  CommandApi.Option("Options", "YMin=0")
  CommandApi.Option("Options", "YSpacing=1")
  CommandApi.Option("Options", "ZMax=0")
  CommandApi.Option("Options", "ZMin=0")
  CommandApi.Option("Options", "ZSpacing=1")
  CommandApi.Option("Path", "C:/Path/To/Import/file.raw")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Idrisi Raster Image .RST, .IMG Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	RST

Example

This example shows how to import a RST file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
```

```
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "rst")
  CommandApi.Option("Path", "C:/Path/To/Import/file.rst")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

USGS Digital Elevation Model .DEM Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	SDTSDEM

Example

This example shows how to import a SDTS DEM file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "sdtsdem")
  CommandApi.Option("Path", "C:/Path/To/Import/file.ddf")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

SDTS Topological Vector Profile and Raster Profile .TVP, .DDF Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description

Filter	string	SDTSTVP

Example

This example shows how to import a SDTS TVP DDF file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "sdtstvp")
  CommandApi.Option("Path", "C:/Path/To/Import/file.ddf")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

SEG-Y Seismic Data Log .SGY, .SEGY Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	SEGY
Segy_ForceIBM	0, 1	0 = Do not force an assumption that the data is in IBM 360 floating-point format. 1 = Do force IBM 360 floating point format.
Segy_ForceIEEE	0, 1	0 = Do not force an assumption that the data is in IEEE floating-point format. 1 = Do force IEEE floating-point format.
Segy_ForceINT16	0, 1	0 = Do not force an assumption that the data is in 16-bit integer format. 1 = Do force 16-bit integer format.
Segy_ForceINT32	0, 1	0 = Do not force an assumption that the data is in 32-bit integer format. 1 = Do force 32-bit integer format.
Segy_ForceINT64	0, 1	0 = Do not force an assumption that the data is in 64-bit integer format. 1 = Do force 64-bit integer format.
Segy_ForceINT8	0, 1	0 = Do not force an assumption that the data is in 8-bit integer format. 1 = Do force 8-bit integer format.

Segy_LittleEnding	0, 1	0 = Do not force an assumption that the data is in little endian format. 1 = Do force little endian format.
Segy_NoBinaryReelHeader	0, 1	0 = Data has a binary header. 1 = Data does not have a binary header.
Segy_NoDisplayReelHeader	0, 1	0 = File has a display reel header. 1 = File does not have a display reel header.
Segy_PassFormat	0, 1, 2	0 = A pointset. See "Segy_XCoordType", "Segy_YCoordType", "Segy_ZCoordType". 1 = 2D lattice. 2 = 1 D lattice.
Segy_XCoordType	0, 1, 2, 3, 4	If Segy_Passformat = 0, specifies which data to use for X coords of pointset. 0=Sample value 1=Trace number 2=Relative sample number 3=Absolute sample number 4=Zero
Segy_YCoordType	0, 1, 2, 3, 4	If Segy_Passformat = 0, specifies which data to use for y coords of pointset. 0=Sample value 1=Trace number 2=Relative sample number 3=Absolute sample number 4=Zero
Segy_ZCoordType	0, 1, 2, 3, 4	If Segy_Passformat = 0, specifies which data to use for z coords of pointset. 0=Sample value 1=Trace number 2=Relative sample number 3=Absolute sample number 4=Zero

Example

This example shows how to import a SEGY file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("AutoConnect", "True")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "segy")
  CommandApi.Option("Options",
"Segy_ForceIBM=0;Segy_ForceIEEE=0;Segy_ForceINT16=0;Segy_ForceINT32=0;Segy_ForceINT8
```

```
=0;Segy_LittleEndian=0;Segy_NoBinaryReelHeader=0;Segy_NoDisplayReelHeader=0;Segy_PassFo
rmat=1;Segy_XCoordType=0;Segy_YCoordType=0;Segy_ZCoordType=0")
  CommandApi.Option("Path", "C:/Path/To/Import/file.sgy")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Silicon Graphics (SGI) RGB Image .RGB, .RGBA, .BW Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	SGI

Example

This example shows how to import a SGI RGB file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "sgi")
  CommandApi.Option("Path", "C:/Path/To/Import/file.rgb")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Surfer Grid .GRD Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	SGRD

Example

This example shows how to import a **Surfer** GRD file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "sgrd")
  CommandApi.Option("Path", "C:/Path/To/Import/file.grd")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Esri Shapefile .SHP Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	SHP
AreasToCurves	0, 1	Whether to set all polygons to polylines. 0 does not set all polygons to polylines. 1 doesset all polygons to polylines.
ImportCodePage	integer	The number of the ANSI code page to use when importing Unicode data. Valid code page number in the range 0 through 65535. If a .CPG file exists in the directory with the same name as the .SHP file, the code page stored in the .CPG file will be used. If a .DBF file exists in the directory with the same name as the .SHP file, the .DBF file attribute data must be translated from ANSI text to

		Unicode representation. If no .CPG file exists, the Language ID value stored in the .DBF file's header determines the code page used. However, if the Language ID value is zero or invalid, the "Western European (Windows)" code page (1252) will be used.

Example

This example shows how to import a SHP file with the various options.

```
CommandApi.Construct("Import")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("AutoConnect", "True")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "shp")
CommandApi.Option("Options", "AreasToCurves=1")
CommandApi.Option("Path", "C:/Path/To/Import/file.shp")
CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

LizardTech MrSID .SID Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Default	Description
Filter	string		SID
Scale	1 = 1/1 (uses most memory, best quality) 2 = 1/2 4 = 1/4 8 = 1/8 16 = 1/16	1	Since some images can be extremely large once expanded into memory, the import filter allows the image to be reduced in dimensions by 1/2 to 1/32 of the original size.

	32 = 1/32 (uses least memory, lowest quality)		
LRX	integer	maximum X	Use LRX to limit the import to a subregion of the file. LRX is the X pixel coordinate for the lower right corner of the desired subregion.
LRY	integer	maximum Y	Use LRY to limit the import to a subregion of the file. LRY is the Y pixel coordinate for the lower right corner of the desired subregion.
ULX	integer	0	Use ULX to limit the import to a subregion of the file. ULX is the X pixel coordinate for the upper left corner of the desired subregion.
ULY	integer	0	Use ULY to limit the import to a subregion of the file. ULY is the Y pixel coordinate for the upper left corner of the desired subregion.

Remarks

The pixel coordinates increase in the X direction from left to right and increase in the Y direction from top to bottom. The pixel coordinate origin is the upper left corner. For example an image with a 2000 pixel width and 3000 pixel height has the following coordinates: the upper left corner is coordinate (0,0) and the lower right corner is the x and y maximums (2000,3000).

However, the image still will import correctly even if the ULY is used for the maximum Y and LRY is used for the minimum Y.

Example

This example shows how to import a SID file with the various options.

```
CommandApi.Construct("Import")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("AutoConnect", "True")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "shp")
CommandApi.Option("Options", "Scale=4")
```

```
CommandApi.Option("Path", "C:/Path/To/Import/file.sid")
CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Sylk Spreadsheet .SLK Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	SLK

Example

This example shows how to import a SLK file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "slk")
  CommandApi.Option("Path", "C:/Path/To/Import/file.slk")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Amira Stacked Images .ASI Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	STACKED

Example

This example shows how to import an ASI file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "stacked")
  CommandApi.Option("Path", "C:/Path/To/Import/file.asi")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Metamorph .STK Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	STK

Example

This example shows how to import a STK file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "stk")
  CommandApi.Option("Path", "C:/Path/To/Import/file.stk")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Sun Raster Image .RAS, .SUN Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	SUN

Example

This example shows how to import a SUN file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "sun")
  CommandApi.Option("Path", "C:/Path/To/Import/file.sun")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Truevision Targa .TGA Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	TGA

Example

This example shows how to import a TGA file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "tga")
  CommandApi.Option("Path", "C:/Path/To/Import/file.tga")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Tagged Image File Format .TIF, .TIFF Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	TIFF

Example

This example shows how to import a TIF file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "tiff")
  CommandApi.Option("Path", "C:/Path/To/Import/file.tif")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Text Data .TXT Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	TXT
ColumnBreaks	Comma separated Integers	Set of integers telling Voxler where the column breaks are. These integers will be separated by commas. Only use if UseFixedWidth = 1. Example: ColumnBreaks=9,18,27
DateOrder	0 = Auto 1 = MDY 2 = DMY	Integer that specifies the order in which dates are written in the data file. When set to 0, the standard date order from the *Locale* is used.

	3 = YMD 4 = MYD 5 = DYM 6 = YDM	
DecimalSymbol	period = period comma = comma	String that specifies the character that separates the integer and fractional components of the data values.
Delimiter	string	String which tells Voxler which delimiter to use to separate each column. Voxler recognises *space*, *tab, comma, semicolon* as strings, otherwise use the literal string. Only use if UseFixedWidth = 0. Example: Delimiter=(Delimiter=comma
EatWhiteSpace	1 or 0	0 does not skip extra spaces between values. 1 skips extra spaces and tabs between values.
ImportCodePage	Integer	The number of the ANSI code page to use when importing Unicode data. Valid code page number in the range 0 through 65535.
Locale	String	Locale ID in decimal values. The default locale is determined by the locale setting in the Windows Control Panel.
SkipExtraDelimiters	1 or 0	0 skips extra delimiters. Multiple delimiters are treated as a single delimiter. 1 makes each delimiter a separate delimiter.
StartRow	Integer	The row to start importing from. Everything before this row will be discarded.
TextQualifier	string	Tells Voxler what quote characters to use. Options are *singlequote*, *doublequote*, or *none.*
UseFixedWidth	1 or 0	1 uses a fixed width column. 0 uses a delimited column.

Example

This example shows how to import a TXT file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
```

```
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "txt")
  CommandApi.Option("Path", "C:/Path/To/Import/file.txt")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Voxler Data .VDAT Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	VDAT

Example

This example shows how to import a **Voxler** VDAT file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "vdat")
  CommandApi.Option("Path", "C:/Path/To/Import/file.vdat")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Visualization Toolkit .VTK Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	VTK

Example

This example shows how to import a VTK file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "vtk")
  CommandApi.Option("Path", "C:/Path/To/Import/file.vtk")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Excel Spreadsheet .XLS Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	XLS
Sheet	integer	Set to the integer representing a specific sheet you would like to import.

Example

This example shows how to import an Excel XLS file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "xls")
  CommandApi.Option("Options", "Sheet=4")
  CommandApi.Option("Path", "C:/Path/To/Import/file.xls")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Excel Spreadsheet .XLSX Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	XLSX
Sheet	integer	Set to the integer representing a specific sheet you would like to import.

Example

This example shows how to import an Excel XLSX file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "xlsx")
  CommandApi.Option("Options", "Sheet=4")
  CommandApi.Option("Path", "C:/Path/To/Import/file.xlsx")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Z-MAP Plus Grid .ASC, .DAT, .GRD, .XYZ, .ZMAP, .ZYC, .ZYCOR Import Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the import options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various import options.

Parameter	Type	Description
Filter	string	ZMAP

Example

This example shows how to import a ZMAP ZYCOR file with the various options.

```
CommandApi.Construct("Import")
  CommandApi.Option("GuiEnabled", "False")
  CommandApi.Option("ProgressEnabled", "False")
  CommandApi.Option("UndoRedoEnabled", "True")
  CommandApi.Option("AutoConnect", "False")
  CommandApi.Option("ClearOptions", "False")
  CommandApi.Option("Filter", "zmap")
  CommandApi.Option("Path", "C:/Path/To/Import/file.zycor")
  CommandApi.Option("PersistOptions", "True")
CommandApi.Do()
```

Used by: Import

Automation Export Options

Automation Export File Types

For the automation *Export* command, the options parameter "Type" is defined by the strings on the left side of this table for the file types listed on the right side.

Type	File Formats
adf	*.adf
am	*.am *.col
analyze	*.img
asc	*.asc *.aig *.agr *.grd
avs	*.fld
avsx	*.x *.ximg
blnW	*.bln
bmp	*.bmp
bnaW	*.bna
bnd	*.bil *.bip *.bsq
csv	*.csv
dat	*.dat
dem	*.dem
eps	*.eps
ers	*.ers
flt	*.flt
geo	*.grd *.ggf
gif	*.gif
gxf	*.gxf
hdf	*.hdf

iv	*.iv
jpg	*.jpg *.jpeg
lat	*.lat
pcx	*.pcx
pdfi	*.pdf
png	*.png
pnm	*.pnm *.ppm *.pgm *.pbm
p3d	*.p3d *.xyz
raw	*.raw *.bin
sgi	*.rgb *.rgba *.bw
sgrd6a	*.grd
sgrd6b	*.grd
sgrd7b	*.grd
slk	*.slk
sun	*.ras *.sun
tga	*.tga
tif	*.tif *.tiff
txt	*.txt
vdat	*.vdat
vtk	*.vtk
xls	*.xls
xyz	*.dat

Used by: Export

Esri ArcInfo Binary Grid .ADF Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	ADF
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default

Example

This example shows how to export an ADF file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "adf")
CommandApi.Option("Path", "C:/Path/To/Export/file.adf")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

Amira Mesh Format .AM, .COL Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	AM
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default

Example

This example shows how to export an AM file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "am")
CommandApi.Option("Path", "C:/Path/To/Export/file.am")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

This example shows how to export an COL file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
```

```
CommandApi.Option("Filter", "am")
CommandApi.Option("Path", "C:/Path/To/Export/file.col")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

Analyze 7.5 Medical Image Format .IMG Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	ANALYZE
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default

Example

This example shows how to export an IMG file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "analyze")
CommandApi.Option("Path", "C:/Path/To/Export/file.img")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

Esri ASCII Grid Format .ASC, .AIG, .AGR, .GRD Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	ASC
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default

Example
This example shows how to export an ASC file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "asc")
CommandApi.Option("Path", "C:/Path/To/Export/file.asc")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

This example shows how to export an AIG file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "asc")
CommandApi.Option("Path", "C:/Path/To/Export/file.aig")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

This example shows how to export an AGR file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "asc")
CommandApi.Option("Path", "C:/Path/To/Export/file.agr")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

This example shows how to export a GRD file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "asc")
CommandApi.Option("Path", "C:/Path/To/Export/file.grd")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

AVS Field Format .FLD Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	AVS
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default

Example

This example shows how to export an FLD file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "avs")
CommandApi.Option("Path", "C:/Path/To/Export/file.fld")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

AVS X-Image Format .X, .XIMG Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	AVSX
ColorDepth	-32, -16, -8, 1, 4, 8, 24, 32	Specifies the color depth to use. ColorDepth=-32 -- 32 bit greyscale output ColorDepth=-16 -- 16 bit greyscale output ColorDepth=-8 -- 8 bit greyscale output ColorDepth=1 -- 1 bit color indexed output ColorDepth=4 -- 4 bit color indexed output ColorDepth=8 -- 8 bit color indexed output ColorDepth=24 -- 24 bit true color output ColorDepth=32 -- 32 bit true color with alpha output
ColorReductionMethod	1, 2, 3, 4, 5	Specifies the color reduction method to use. ColorReductionMethod=1 -- Use the Ordered Dither color reduction method ColorReductionMethod=2 -- Use the Diffused Dither color reduction method ColorReductionMethod=3 -- Use the Popularity color reduction method ColorReductionMethod=4 -- Use the MedianCut555 color reduction method ColorReductionMethod=5 -- Use the MedianCut888 color reduction method
HDPI	integer	Sets the number of horizontal pixels in the exported image that corresponds to a distance one horizontal inch. Either a Width or an HDPI option may be specified, but not both. The default HDPI is the resolution of the display device driver on your computer (this is 96 dots per inch on many Windows display devices, but number can very from device to device).
Height	integer	Sets the height of the exported image in pixels. Either a Height or a VDPI option may be specified, but not both
KeepAspect	0 or 1	If set to 1, *KeepAspect* adjusts the aspect ratio (the width versus height) of the exported image to match the aspect ratio of the Viewer window that is being exported. If an option is given to set the *Width* or *HDPI* of an image, *KeepAspect* automatically assigns the appropriate *Height* or *VDPI* to maintain the same aspect ratio as the Viewer window. Likewise, if an option is given to set the *Height* or *VDPI* of an image, *KeepAspect* automatically assigns the appropriate *Width* or

		HDPI to maintain the same aspect ratio of the Viewer window. Note that the *KeepAspect* option is ignored if both the *Width* and *Height* options or the *HDPI* and *VDPI* options are specified. If set to 0, the aspect ratio of the exported image is determined solely by the *Width* and *Height* or *HDPI* and *VDPI* options.
KeepPixelSize	0 or 1	Locks the pixel dimension width and height. Changes to the *VDPI* and *HDPI* will affect the document size, but not the pixel size, when this is set to 1. Changes to the *VDPI* and *HDPI* will affect the pixel dimensions, but not the document size, when this is set to 0. Note that the *KeepPixelSize* option is ignored if the either the *Width* or *Height* option is specified.
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default
VDPI	integer	Sets the number of vertical pixels in the exported image that corresponds to a distance of one vertical inch. Either a Height or a VDPI option may be specified, but not both. The default VDPI is the resolution of the display device driver on your computer (this is 96 dots per inch on many Windows display devices, but number can very from device to device)
Width	integer	Sets the width of the exported image in pixels. Either a Width or an HDPI option may be specified, but not both.

Example

This example shows how to export an X file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "avsx")
CommandApi.Option("Options", "HDPI=96; Height=418; VDPI=96; Width=576; ")
CommandApi.Option("Path", "C:/Path/To/Export/file.x")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "0")
CommandApi.Do()
```

Example 2

This example shows how to export an XIMG file with the various options.

```
CommandApi.Construct("Export")
```

```
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "avsx")
CommandApi.Option("Options", "HDPI=96")
CommandApi.Option("Options", "Height=418")
CommandApi.Option("Options", "VDPI=96")
CommandApi.Option("Options", "Width=576")
CommandApi.Option("Path", "C:/Path/To/Export/file.ximg")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

Golden Software BLN File Format .BLN Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	BLN
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default
BreakApartCompoundAreas	0 = No 1 = Yes	Compound areas (multi-ring polygons) will be split apart into multiple non-compound areas (simple polygons) during export.
WriteZ	0 = No 1 = Yes	Z values (elevation) will be included in the .BLN file. When Z values are not available for an object, 0 will be written in the Z column.
BlankMode	0 = Blank outside 1 = Blank inside	A value of 0 sets the blanking flag to 0, and regions outside the areas are blanked. A value of 1 sets the blanking flag to 1, and regions inside the areas are blanked.

Example

This example shows how to export an BLN file with the various options.

```
CommandApi.Construct("Export")
```

```
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "bln")
CommandApi.Option("Options", "BreakApartCompoundAreas=0;WriteZ=1;BlankMode=0")
CommandApi.Option("Path", "C:/Path/To/Export/file.bln")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

Windows Bitmap File Format .BMP Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	BMP
ColorDepth	-32, -16, -8, 1, 4, 8, 24, 32	Specifies the color depth to use. ColorDepth=-32 -- 32 bit greyscale output ColorDepth=-16 -- 16 bit greyscale output ColorDepth=-8 -- 8 bit greyscale output ColorDepth=1 -- 1 bit color indexed output ColorDepth=4 -- 4 bit color indexed output ColorDepth=8 -- 8 bit color indexed output ColorDepth=24 -- 24 bit true color output ColorDepth=32 -- 32 bit true color with alpha output
ColorReductionMethod	1, 2, 3, 4, 5	Specifies the color reduction method to use. ColorReductionMethod=1 -- Use the Ordered Dither color reduction method ColorReductionMethod=2 -- Use the Diffused Dither color reduction method ColorReductionMethod=3 -- Use the Popularity color reduction method ColorReductionMethod=4 -- Use the MedianCut555 color reduction method ColorReductionMethod=5 -- Use the MedianCut888 color reduction method
HDPI	integer	Sets the number of horizontal pixels in the exported image that corresponds to a distance one horizontal inch. Either a Width or an HDPI option may be specified, but not both. The default HDPI is the resolution of the display device driver on your computer (this is 96 dots

		per inch on many Windows display devices, but number can very from device to device).
Height	integer	Sets the height of the exported image in pixels. Either a Height or a VDPI option may be specified, but not both
KeepAspect	0 or 1	If this is set to 1, for any of the settings Height, Width, HDPI or VDPI you only need to set one option and Voxler will automatically calculate the other parameters in line with the aspect ratio of the input image. KeepAspect=0 -- Do not retain the aspect ratio KeepAspect=1 -- Do retain the aspect ratio
KeepPixelSize	0 or 1	Locks the pixel dimension width and height. Changes to the *VDPI* and *HDPI* will affect the document size, but not the pixel size, when this is set to 1. Changes to the *VDPI* and *HDPI* will affect the pixel dimensions, but not the document size, when this is set to 0. Note that the *KeepPixelSize* option is ignored if the either the *Width* or *Height* option is specified.
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default
VDPI	integer	Sets the number of vertical pixels in the exported image that corresponds to a distance of one vertical inch. Either a Height or a VDPI option may be specified, but not both. The default VDPI is the resolution of the display device driver on your computer (this is 96 dots per inch on many Windows display devices, but number can very from device to device)
Width	integer	Sets the width of the exported image in pixels. Either a Width or an HDPI option may be specified, but not both.

Example

This example shows how to export an BMP file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "bmp")
CommandApi.Option("Options", "HDPI=96; Height=418; VDPI=96; Width=576")
CommandApi.Option("Path", "C:/Path/To/Export/file.bmp")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "0")
CommandApi.Do()
```

Used by: Export

Atlas Boundary File Format .BNA Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	BNAW
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default

Example

This example shows how to export an BNA file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "bnaW")
CommandApi.Option("Path", "C:/Path/To/Export/file.bna")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

Banded Lattice File Format .BIL, .BIP, .BSQ Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	BND
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default

Example
This example shows how to export an BIL file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "bnd")
CommandApi.Option("Path", "C:/Path/To/Export/file.bil")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Example 2
This example shows how to export an BIP file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "bnd")
CommandApi.Option("Path", "C:/Path/To/Export/file.bip")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Example 3
This example shows how to export an BSQ file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "bnd")
CommandApi.Option("Path", "C:/Path/To/Export/file.bsq")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

Comma Separated Variable File Format .CSV Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	CSV
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default

Example

This example shows how to export an CSV file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "csv")
CommandApi.Option("Path", "C:/Path/To/Export/file.csv")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

Golden Software Data File Format .DAT Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Delimiter	comma, tab, space, semicolon	The delimiter is the separator between columns.
Filter	string	DAT
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default

TextQualifier	doublequote, singlequote, none	The text qualifier is the mark around text, keeping text in one cell.

Example

This example shows how to export an DAT file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "dat")
CommandApi.Option("Options", "Delimiter=comma; TextQualifier=doublequote")
CommandApi.Option("Path", "C:/Path/To/Export/file.dat")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

USGS DEM File Format .DEM Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	DEM
BlockType	0, 1, 2	Defines the data block type. BlockType=0 -- 15-minute block BlockType=1 -- 7.5-minute block BlockType=2 -- Other block type.
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default
SectNum	integer	The section indicator. For 15-min block it must be >=1 and <=8. For 7.5-min block it must be >=1 and <=32.
XYUnitCode	0, 1, 2, 3, 4	The unit code of the X and Y coordinate units. 0=radians 1=feet 2=meters 3=arc-seconds

		4=degrees
ZUnitCode	1, 2	The unit code of the Z coordinate units. 1=feet 2=meters

Example
This example shows how to export an DEM file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "dem")
CommandApi.Option("Options", "BlockType=1; SectNum=1; XYUnitCode=1; ZUnitCode=1")
CommandApi.Option("Path", "C:/Path/To/Export/file.dem")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

AutoCAD DXF Export Automation Options

Since the **Export Options** dialog is not displayed when the program is driven from an automation script, an options string can be specified in the script. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Option	Action	Default	Description
FileCompatibility	13 = AutoCAD Release 13 (or earlier) 14 = AutoCAD Release 14 (or later) 18 = AutoCAD 2004 (or later)	14	Specifies which version of the DXF format is to be used for export.
FormatASCII	0 = Binary 1 = Text (ASCII)	1	File format of exported DXF file. The ASCII file format is larger than the DXF binary format, but is compatible with a wider variety of software programs.
MaxBitmapSizeInMB	N	10	For AutoCAD 2004 format, this option specifies the largest size allowed for an individual bitmap in the DXF file, in Megabytes. Any exported bitmap larger than this size, will have its resolution reduced so it does not exceed the maximum size.

AllColorsSame	0 = No 1 = Yes	0	Convert all colors to the default color.
AllStylesSame	0 = No 1 = Yes	0	Convert all line styles to the default line style.
AllWidthsSame	0 = No 1 = Yes	0	Convert all line widths to the default line width.
AllTextToAreas	0 = No 1 = Yes	0	Convert all text entities in the exported document to solid areas
FillSolidAreas	0 = No 1 = Yes	0	Fill solid area interiors.
UseSpatialInfo	0 = No 1 = Yes	0	Use ONLY spatial information. Only the basic geometry (lines, areas, text) will be written to the DXF file, and all line styles, colors, fills, and other attributes will be discarded.
ColorMapping	0 = Linear 1 = Weighted LUV	0	Choose the indexed color mapping as either weighted LUV export color mapping, or linear export color mapping.
RenderMarkers	0 = No 1 = Yes	1	Render marker symbols as lines and polygons. When set to 0, marker symbols are exported to the DXF file as points.

Remarks

Boolean values within options strings are not equivalent to Booleans in **Scripter** BASIC. Use "1" instead of "True" and "0" instead of "False".

Encapsulated PostScript File Format .EPS Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	BMP
ColorDepth	-32, -16, -8, 1, 4, 8, 24, 32	Specifies the color depth to use. ColorDepth=-32 -- 32 bit greyscale output ColorDepth=-16 -- 16 bit greyscale output ColorDepth=-8 -- 8 bit greyscale output ColorDepth=1 -- 1 bit color indexed output ColorDepth=4 -- 4 bit color indexed output ColorDepth=8 -- 8 bit color indexed output

		ColorDepth=24 -- 24 bit true color output ColorDepth=32 -- 32 bit true color with alpha output
ColorReductionMethod	1, 2, 3, 4, 5	Specifies the color reduction method to use. ColorReductionMethod=1 -- Use the Ordered Dither color reduction method ColorReductionMethod=2 -- Use the Diffused Dither color reduction method ColorReductionMethod=3 -- Use the Popularity color reduction method ColorReductionMethod=4 -- Use the MedianCut555 color reduction method ColorReductionMethod=5 -- Use the MedianCut888 color reduction method
HDPI	integer	Sets the number of horizontal pixels in the exported image that corresponds to a distance one horizontal inch. Either a Width or an HDPI option may be specified, but not both. The default HDPI is the resolution of the display device driver on your computer (this is 96 dots per inch on many Windows display devices, but number can very from device to device).
Height	integer	Sets the height of the exported image in pixels. Either a Height or a VDPI option may be specified, but not both
KeepAspect	0 or 1	If this is set to 1, for any of the settings Height, Width, HDPI or VDPI you only need to set one option and Voxler will automatically calculate the other parameters in line with the aspect ratio of the input image. KeepAspect=0 -- Do not retain the aspect ratio KeepAspect=1 -- Do retain the aspect ratio
KeepPixelSize	0 or 1	Locks the pixel dimension width and height. Changes to the *VDPI* and *HDPI* will affect the document size, but not the pixel size, when this is set to 1. Changes to the *VDPI* and *HDPI* will affect the pixel dimensions, but not the document size, when this is set to 0. Note that the *KeepPixelSize* option is ignored if the either the *Width* or *Height* option is specified.
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default
PreviewColor	0, 1	Tells Voxler whether to use a color or grey scale preview. PreviewColor=0 -- Use grey scale preview PreviewColor=1 -- Use color preview

PreviewSize	0, 1, 2, 3, 4	Tells Voxler whether to use 64l, 512k or a percentage of total size for preview PreviewSize=0 -- use 64k PreviewSize=1 -- use 512k PreviewSize=2 -- use 25% of total PreviewSize=3 -- use 50% of total PreviewSize=4 -- use 100% of total
PreviewType	0, 1, 2	Tells Voxler what kind of preview to embed. PreviewType=0 -- No preview PreviewType=1 -- EPSI preview PreviewType=2 -- TIFF preview
VDPI	integer	Sets the number of vertical pixels in the exported image that corresponds to a distance of one vertical inch. Either a Height or a VDPI option may be specified, but not both. The default VDPI is the resolution of the display device driver on your computer (this is 96 dots per inch on many Windows display devices, but number can very from device to device)
Width	integer	Sets the width of the exported image in pixels. Either a Width or an HDPI option may be specified, but not both.

Example

This example shows how to export an EPS file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "eps")
CommandApi.Option("Options", "ColorDepth=24; ColorReductionMethod=5; HDPI=96;
Height=903; KeepAspect=1; VDPI=96; Width=716")
CommandApi.Option("Options", "PreviewColor=1; PreviewSize=1; PreviewType=2")
CommandApi.Option("Path", "C:/Path/To/Export/file.eps")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "0")
CommandApi.Do()
```

Used by: Export

ER Mapper Grid File Format .ERS Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	ERS
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default

Example

This example shows how to export an ERS file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "ers")
CommandApi.Option("Path", "C:/Path/To/Export/file.ers")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

Esri Float Grid File Format .FLT Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	FLT
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default

Example

This example shows how to export an FLT file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
```

```
CommandApi.Option("Filter", "flt")
CommandApi.Option("Path", "C:/Path/To/Export/file.flt")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

Geosoft Binary Grid File Format .GGF, .GRD Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	GEO
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default

Example

This example shows how to export an GGF file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "geo")
CommandApi.Option("Path", "C:/Path/To/Export/file.ggf")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Example 2

This example shows how to export an GRD file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "geo")
CommandApi.Option("Path", "C:/Path/To/Export/file.grd")
CommandApi.Option("PersistOptions", "True")
```

CommandApi.Option("ModuleId", "2")
CommandApi.Do()

Used by: Export

GIF Image File Format .GIF Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	GIF
BackgroundColor	16777216 = no transparency 16777217 = background transparency 16777215 = white transparency <16777216 = custom color transparency	If the value is less than 16777216, the color is composed of red, green, and blue and is set to a custom color. The color value is calculated by: Red + Green*256 + Blue*65536 where Red, Green, and Blue are values between 0 and 255.
ColorDepth	-32, -16, -8, 1, 4, 8, 24, 32	Specifies the color depth to use. ColorDepth=-32 -- 32 bit greyscale output ColorDepth=-16 -- 16 bit greyscale output ColorDepth=-8 -- 8 bit greyscale output ColorDepth=1 -- 1 bit color indexed output ColorDepth=4 -- 4 bit color indexed output ColorDepth=8 -- 8 bit color indexed output ColorDepth=24 -- 24 bit true color output ColorDepth=32 -- 32 bit true color with alpha output
ColorReductionMethod	1, 2, 3, 4, 5	Specifies the color reduction method to use. ColorReductionMethod=1 -- Use the Ordered Dither color reduction method ColorReductionMethod=2 -- Use the Diffused Dither color reduction method ColorReductionMethod=3 -- Use the Popularity color reduction method

		ColorReductionMethod=4 -- Use the MedianCut555 color reduction method ColorReductionMethod=5 -- Use the MedianCut888 color reduction method
HDPI	integer	Sets the number of horizontal pixels in the exported image that corresponds to a distance one horizontal inch. Either a Width or an HDPI option may be specified, but not both. The default HDPI is the resolution of the display device driver on your computer (this is 96 dots per inch on many Windows display devices, but number can very from device to device).
Height	integer	Sets the height of the exported image in pixels. Either a Height or a VDPI option may be specified, but not both
KeepAspect	0 or 1	If this is set to 1, for any of the settings Height, Width, HDPI or VDPI you only need to set one option and Voxler will automatically calculate the other parameters in line with the aspect ratio of the input image. KeepAspect=0 -- Do not retain the aspect ratio KeepAspect=1 -- Do retain the aspect ratio
KeepPixelSize	0 or 1	Locks the pixel dimension width and height. Changes to the *VDPI* and *HDPI* will affect the document size, but not the pixel size, when this is set to 1. Changes to the *VDPI* and *HDPI* will affect the pixel dimensions, but not the document size, when this is set to 0. Note that the *KeepPixelSize* option is ignored if the either the *Width* or *Height* option is specified.
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default
VDPI	integer	Sets the number of vertical pixels in the exported image that corresponds to a distance of one vertical inch. Either a Height or a VDPI option may be specified, but not both. The default VDPI is the resolution of the display device driver on your computer (this is 96 dots per inch on many Windows display devices, but number can very from device to device)
Width	integer	Sets the width of the exported image in pixels. Either a Width or an HDPI option may be specified, but not both.

Example

This example shows how to export an GIF file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "gif")
CommandApi.Option("Options", "BackgroundColor=16777215; ColorDepth=32; HDPI=96;
Height=418; VDPI=96; Width=576; _
Format=3; Compress=2; UseTransparency=1")
CommandApi.Option("Path", "C:/Path/To/Export/file.gif")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

Grid eXchange File Format .GXF Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	GXF
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default
Gxf_Compression	0, 1, 2, 3, 4, 5	Specifies the number of base-90 digits to use for compression. 0 = no compression (default) which creates a readable text output whose significant figures are controlled by the *GXF_SigFigs* option 1 = very low compression precision approximately 1e-10 2 =- low compression precision approximately 1e-8 3 = medium compression precision approximately 1e-6 4 = high compression precision approximately 1e-4 5 = very high compression precision approximately 1e-2

Gxf_SigFigs	integer	The number of significant figures used when outputting ASCII floating point numbers. Default = 14.

Example

This example shows how to export an GXF file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "gxf")
CommandApi.Option("Options", "Gxf_Compression=0; Gxf_SigFigs=14")
CommandApi.Option("Path", "C:/Path/To/Export/file.gxf")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

Hierarchical Data File Format .HDF Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	HDF
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default

Example

This example shows how to export an HDF file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "hdf")
CommandApi.Option("Path", "C:/Path/To/Export/file.hdf")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

SGI OpenInventor File Format .IV Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	IV
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default

Example

This example shows how to export an IV file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "iv")
CommandApi.Option("Path", "C:/Path/To/Export/file.iv")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

JPEG Compressed Bitmap File Format .JPG Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	JPG
ColorDepth	-32, -16, -8, 1, 4, 8, 24, 32	Specifies the color depth to use. ColorDepth=-32 -- 32 bit greyscale output ColorDepth=-16 -- 16 bit greyscale output ColorDepth=-8 -- 8 bit greyscale output

		ColorDepth=1 -- 1 bit color indexed output ColorDepth=4 -- 4 bit color indexed output ColorDepth=8 -- 8 bit color indexed output ColorDepth=24 -- 24 bit true color output ColorDepth=32 -- 32 bit true color with alpha output
ColorReductionMethod	1, 2, 3, 4, 5	Specifies the color reduction method to use. ColorReductionMethod=1 -- Use the Ordered Dither color reduction method ColorReductionMethod=2 -- Use the Diffused Dither color reduction method ColorReductionMethod=3 -- Use the Popularity color reduction method ColorReductionMethod=4 -- Use the MedianCut555 color reduction method ColorReductionMethod=5 -- Use the MedianCut888 color reduction method
HDPI	integer	Sets the number of horizontal pixels in the exported image that corresponds to a distance one horizontal inch. Either a Width or an HDPI option may be specified, but not both. The default HDPI is the resolution of the display device driver on your computer (this is 96 dots per inch on many Windows display devices, but number can very from device to device).
Height	integer	Sets the height of the exported image in pixels. Either a Height or a VDPI option may be specified, but not both
KeepAspect	0 or 1	If this is set to 1, for any of the settings Height, Width, HDPI or VDPI you only need to set one option and Voxler will automatically calculate the other parameters in line with the aspect ratio of the input image. KeepAspect=0 -- Do not retain the aspect ratio KeepAspect=1 -- Do retain the aspect ratio
KeepPixelSize	0 or 1	Locks the pixel dimension width and height. Changes to the *VDPI* and *HDPI* will affect the document size, but not the pixel size, when this is set to 1. Changes to the *VDPI* and *HDPI* will affect the pixel dimensions, but not the document size, when this is set to 0. Note that the *KeepPixelSize* option is ignored if the either the *Width* or *Height* option is specified.
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default
Quality	integer from 0 to 100	Represents a percentage of original image quality which the output image has. Higher

		value = higher quality. 100 = highest quality. 1 = lowest quality.
VDPI	integer	Sets the number of vertical pixels in the exported image that corresponds to a distance of one vertical inch. Either a Height or a VDPI option may be specified, but not both. The default VDPI is the resolution of the display device driver on your computer (this is 96 dots per inch on many Windows display devices, but number can very from device to device)
Width	integer	Sets the width of the exported image in pixels. Either a Width or an HDPI option may be specified, but not both.

Example

This example shows how to export an JPG file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "jpg")
CommandApi.Option("Options", "HDPI=83; Height=361; PreviewColor=0; Quality=70; VDPI=83; Width=498")
CommandApi.Option("Path", "C:/Path/To/Export/file.jpeg")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

JPEG-2000 File Interchange Format .JP2 Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	JPG2000
ColorDepth	-32, -16, -8, 1, 4, 8, 24, 32	Specifies the color depth to use. ColorDepth=-32 -- 32 bit greyscale output ColorDepth=-16 -- 16 bit greyscale output ColorDepth=-8 -- 8 bit greyscale output ColorDepth=1 -- 1 bit color indexed output ColorDepth=4 -- 4 bit color indexed output ColorDepth=8 -- 8 bit color indexed output

		ColorDepth=24 -- 24 bit true color output ColorDepth=32 -- 32 bit true color with alpha output
ColorReductionMethod	1, 2, 3, 4, 5	Specifies the color reduction method to use. ColorReductionMethod=1 -- Use the Ordered Dither color reduction method ColorReductionMethod=2 -- Use the Diffused Dither color reduction method ColorReductionMethod=3 -- Use the Popularity color reduction method ColorReductionMethod=4 -- Use the MedianCut555 color reduction method ColorReductionMethod=5 -- Use the MedianCut888 color reduction method
FormatJ2K	0 or 1	Sets the Container Format 0 = JP2 container 1 = J2K container
HDPI	integer	Sets the number of horizontal pixels in the exported image that corresponds to a distance one horizontal inch. Either a Width or an HDPI option may be specified, but not both. The default HDPI is the resolution of the display device driver on your computer (this is 96 dots per inch on many Windows display devices, but number can very from device to device).
Height	integer	Sets the height of the exported image in pixels. Either a Height or a VDPI option may be specified, but not both
KeepAspect	0 or 1	If this is set to 1, for any of the settings Height, Width, HDPI or VDPI you only need to set one option and Voxler will automatically calculate the other parameters in line with the aspect ratio of the input image. KeepAspect=0 -- Do not retain the aspect ratio KeepAspect=1 -- Do retain the aspect ratio
KeepPixelSize	0 or 1	Locks the pixel dimension width and height. Changes to the *VDPI* and *HDPI* will affect the document size, but not the pixel size, when this is set to 1. Changes to the *VDPI* and *HDPI* will affect the pixel dimensions, but not the document size, when this is set to 0. Note that the *KeepPixelSize* option is ignored if the either the *Width* or *Height* option is specified.
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default

Quality	integer from 0 to 100	Represents a percentage of original image quality which the output image has. Higher value = higher quality. 100 = highest quality. 1 = lowest quality.
VDPI	integer	Sets the number of vertical pixels in the exported image that corresponds to a distance of one vertical inch. Either a Height or a VDPI option may be specified, but not both. The default VDPI is the resolution of the display device driver on your computer (this is 96 dots per inch on many Windows display devices, but number can very from device to device)
Width	integer	Sets the width of the exported image in pixels. Either a Width or an HDPI option may be specified, but not both.

Example

This example shows how to export an JP2 file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "jpg2000")
CommandApi.Option("Options", "HDPI=83; Height=361; PreviewColor=0; Quality=70; VDPI=83; Width=498")
CommandApi.Option("Path", "C:/Path/To/Export/file.jp2")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

Iris Explorer File Format .LAT Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	LAT
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default

Example

This example shows how to export an LAT file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "lat")
CommandApi.Option("Path", "C:/Path/To/Export/file.lat")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

Plot 3D File Format .P3D, .XYZ Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Blanked	0, 1	The Blanked option specifies whether blanking data is present in or absent from the exported file. 0 = Blanking Absent 1 = Blanking Present
DataFormat	0, 1, 2	The DataFormat option specifies how the data values are encoded in the Plot-3d files. 0 = Raw Binary 1 = Formatted ASCII 2 = FORTRAN Unformatted
Filter	string	P3D
GridFormat	0, 1	Specifies the type of grid.
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default
MultiZone	0, 1	Specifies whether the P3D is single or multiple zones. 0 = single zone 1 = multiple zones

Example

This example shows how to export an P3D file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "p3d")
CommandApi.Option("Options", "Blanked=1; DataFormat=1; GridFormat=0; MultiZone=1")
CommandApi.Option("Path", "C:/Path/To/Export/file.p3d")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

ZSoft Paintbrush File Format .PCX Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	PCX
ColorDepth	-32, -16, -8, 1, 4, 8, 24, 32	Specifies the color depth to use. ColorDepth=-32 -- 32 bit greyscale output ColorDepth=-16 -- 16 bit greyscale output ColorDepth=-8 -- 8 bit greyscale output ColorDepth=1 -- 1 bit color indexed output ColorDepth=4 -- 4 bit color indexed output ColorDepth=8 -- 8 bit color indexed output ColorDepth=24 -- 24 bit true color output ColorDepth=32 -- 32 bit true color with alpha output
ColorReductionMethod	1, 2, 3, 4, 5	Specifies the color reduction method to use. ColorReductionMethod=1 -- Use the Ordered Dither color reduction method ColorReductionMethod=2 -- Use the Diffused Dither color reduction method ColorReductionMethod=3 -- Use the Popularity color reduction method ColorReductionMethod=4 -- Use the MedianCut555 color reduction method ColorReductionMethod=5 -- Use the MedianCut888 color reduction method
HDPI	integer	Specifies the number of pixels per horizontal inch.
Height	integer	Specifies the number of pixels tall an output image will be.

KeepAspect	0 or 1	If this is set to 1, for any of the settings Height, Width, HDPI or VDPI you only need to set one option and Voxler will automatically calculate the other parameters in line with the aspect ratio of the input image. KeepAspect=0 -- Do not retain the aspect ratio KeepAspect=1 -- Do retain the aspect ratio
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default
VDPI	integer	Specifies the number of pixels per vertical inch.
Width	integer	Specifies the number of pixels wide an output image will be.

Example

This example shows how to export an PCX file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "pcx")
CommandApi.Option("Options", "HDPI=96; Height=418; VDPI=96; Width=576")
CommandApi.Option("Path", "C:/Path/To/Export/file.pcx")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "0")
CommandApi.Do()
```

Used by: Export

Raster PDF File Format .PDF Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	PDFI
ColorDepth	-32, -16, -8, 1, 4, 8, 24, 32	Specifies the color depth to use. ColorDepth=-32 -- 32 bit greyscale output ColorDepth=-16 -- 16 bit greyscale output ColorDepth=-8 -- 8 bit greyscale output ColorDepth=1 -- 1 bit color indexed output ColorDepth=4 -- 4 bit color indexed output

		ColorDepth=8 -- 8 bit color indexed output ColorDepth=24 -- 24 bit true color output ColorDepth=32 -- 32 bit true color with alpha output
ColorReductionMethod	1, 2, 3, 4, 5	Specifies the color reduction method to use. ColorReductionMethod=1 -- Use the Ordered Dither color reduction method ColorReductionMethod=2 -- Use the Diffused Dither color reduction method ColorReductionMethod=3 -- Use the Popularity color reduction method ColorReductionMethod=4 -- Use the MedianCut555 color reduction method ColorReductionMethod=5 -- Use the MedianCut888 color reduction method
CompressImages	0 or 1	Compresses images in the PDF to reduce file size. 0 = False 1 = True
FitPage	0 or 1	Sets the size of the PDF to the application page size if True (1). Sets the PDF page size to the size of the exported objects if False (0).
HDPI	integer	Sets the number of horizontal pixels in the exported image that corresponds to a distance one horizontal inch. Either a Width or an HDPI option may be specified, but not both. The default HDPI is the resolution of the display device driver on your computer (this is 96 dots per inch on many Windows display devices, but number can very from device to device).
Height	integer	Sets the height of the exported image in pixels. Either a Height or a VDPI option may be specified, but not both
KeepAspect	0 or 1	If this is set to 1, for any of the settings Height, Width, HDPI or VDPI you only need to set one option and Voxler will automatically calculate the other parameters in line with the aspect ratio of the input image. KeepAspect=0 -- Do not retain the aspect ratio KeepAspect=1 -- Do retain the aspect ratio
KeepPixelSize	0 or 1	Locks the pixel dimension width and height. Changes to the *VDPI* and *HDPI* will affect the document size, but not the pixel size, when this is set to 1. Changes to the *VDPI* and *HDPI* will affect the pixel dimensions, but not the document size, when this is set to 0. Note that the *KeepPixelSize* option is ignored if the either the *Width* or *Height* option is specified.

ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default
VDPI	integer	Sets the number of vertical pixels in the exported image that corresponds to a distance of one vertical inch. Either a Height or a VDPI option may be specified, but not both. The default VDPI is the resolution of the display device driver on your computer (this is 96 dots per inch on many Windows display devices, but number can very from device to device)
Width	integer	Sets the width of the exported image in pixels. Either a Width or an HDPI option may be specified, but not both.

Example

This example shows how to export an PDF file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "pdf")
CommandApi.Option("Options", "HDPI=96; Height=418; VDPI=96; Width=576")
CommandApi.Option("Path", "C:/Path/To/Export/file.pdf")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "0")
CommandApi.Do()
```

Used by: Export

Portable Network Graphic File Format .PNG Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	PNG
ColorDepth	-32, -16, -8, 1, 4, 8, 24, 32	Specifies the color depth to use. ColorDepth=-32 -- 32 bit greyscale output ColorDepth=-16 -- 16 bit greyscale output ColorDepth=-8 -- 8 bit greyscale output ColorDepth=1 -- 1 bit color indexed output

		ColorDepth=4 -- 4 bit color indexed output ColorDepth=8 -- 8 bit color indexed output ColorDepth=24 -- 24 bit true color output ColorDepth=32 -- 32 bit true color with alpha output
ColorReductionMethod	1, 2, 3, 4, 5	Specifies the color reduction method to use. ColorReductionMethod=1 -- Use the Ordered Dither color reduction method ColorReductionMethod=2 -- Use the Diffused Dither color reduction method ColorReductionMethod=3 -- Use the Popularity color reduction method ColorReductionMethod=4 -- Use the MedianCut555 color reduction method ColorReductionMethod=5 -- Use the MedianCut888 color reduction method
HDPI	integer	Sets the number of horizontal pixels in the exported image that corresponds to a distance one horizontal inch. Either a Width or an HDPI option may be specified, but not both. The default HDPI is the resolution of the display device driver on your computer (this is 96 dots per inch on many Windows display devices, but number can very from device to device).
Height	integer	Sets the height of the exported image in pixels. Either a Height or a VDPI option may be specified, but not both
KeepAspect	0 or 1	If this is set to 1, for any of the settings Height, Width, HDPI or VDPI you only need to set one option and Voxler will automatically calculate the other parameters in line with the aspect ratio of the input image. KeepAspect=0 -- Do not retain the aspect ratio KeepAspect=1 -- Do retain the aspect ratio
KeepPixelSize	0 or 1	Locks the pixel dimension width and height. Changes to the *VDPI* and *HDPI* will affect the document size, but not the pixel size, when this is set to 1. Changes to the *VDPI* and *HDPI* will affect the pixel dimensions, but not the document size, when this is set to 0. Note that the *KeepPixelSize* option is ignored if the either the *Width* or *Height* option is specified.
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default
UseTransparency	0 or 1	Use application background transparency for export. If set to 1, background is transparent.
VDPI	integer	Sets the number of vertical pixels in the exported image that corresponds to a distance

		of one vertical inch. Either a Height or a VDPI option may be specified, but not both. The default VDPI is the resolution of the display device driver on your computer (this is 96 dots per inch on many Windows display devices, but number can very from device to device)
Width	integer	Sets the width of the exported image in pixels. Either a Width or an HDPI option may be specified, but not both.

Example

This example shows how to export an PNG file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "png")
CommandApi.Option("Options", "HDPI=96; Height=418; VDPI=96; Width=576;
UseTransparency=1")
CommandApi.Option("Path", "C:/Path/To/Export/file.png")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "0")
CommandApi.Do()
```

Used by: Export

Image File Format .PNM, .PPM, .PGM, .PBM Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	PNM
ColorDepth	-32, -16, -8, 1, 4, 8, 24, 32	Specifies the color depth to use. ColorDepth=-32 -- 32 bit greyscale output ColorDepth=-16 -- 16 bit greyscale output ColorDepth=-8 -- 8 bit greyscale output ColorDepth=1 -- 1 bit color indexed output ColorDepth=4 -- 4 bit color indexed output ColorDepth=8 -- 8 bit color indexed output ColorDepth=24 -- 24 bit true color output ColorDepth=32 -- 32 bit true color with alpha output

ColorReductionMethod	1, 2, 3, 4, 5	Specifies the color reduction method to use. ColorReductionMethod=1 -- Use the Ordered Dither color reduction method ColorReductionMethod=2 -- Use the Diffused Dither color reduction method ColorReductionMethod=3 -- Use the Popularity color reduction method ColorReductionMethod=4 -- Use the MedianCut555 color reduction method ColorReductionMethod=5 -- Use the MedianCut888 color reduction method
HDPI	integer	Specifies the number of pixels per horizontal inch.
Height	integer	Specifies the number of pixels tall an output image will be.
KeepAspect	0 or 1	If this is set to 1, for any of the settings Height, Width, HDPI or VDPI you only need to set one option and Voxler will automatically calculate the other parameters in line with the aspect ratio of the input image. KeepAspect=0 -- Do not retain the aspect ratio KeepAspect=1 -- Do retain the aspect ratio
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default
VDPI	integer	Specifies the number of pixels per vertical inch.
Width	integer	Specifies the number of pixels wide an output image will be.

Example

This example shows how to export an PNM file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "pnm")
CommandApi.Option("Options", "HDPI=96; Height=418; VDPI=96; Width=576")
CommandApi.Option("Path", "C:/Path/To/Export/file.pnm")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "0")
CommandApi.Do()
```

Used by: Export

RAW Binary Lattice File Format .RAW, .BIN Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	RAW
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default

Example

This example shows how to export an RAW file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "raw")
CommandApi.Option("Path", "C:/Path/To/Export/file.raw")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

SGI-RGB Image File Format .RGB, .RGBA, .BW Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	SGI
ColorDepth	-32, -16, -8, 1, 4, 8, 24, 32	Specifies the color depth to use. ColorDepth=-32 -- 32 bit greyscale output ColorDepth=-16 -- 16 bit greyscale output ColorDepth=-8 -- 8 bit greyscale output ColorDepth=1 -- 1 bit color indexed output ColorDepth=4 -- 4 bit color indexed output ColorDepth=8 -- 8 bit color indexed output

		ColorDepth=24 -- 24 bit true color output ColorDepth=32 -- 32 bit true color with alpha output
ColorReductionMethod	1, 2, 3, 4, 5	Specifies the color reduction method to use. ColorReductionMethod=1 -- Use the Ordered Dither color reduction method ColorReductionMethod=2 -- Use the Diffused Dither color reduction method ColorReductionMethod=3 -- Use the Popularity color reduction method ColorReductionMethod=4 -- Use the MedianCut555 color reduction method ColorReductionMethod=5 -- Use the MedianCut888 color reduction method
HDPI	integer	Sets the number of horizontal pixels in the exported image that corresponds to a distance one horizontal inch. Either a Width or an HDPI option may be specified, but not both. The default HDPI is the resolution of the display device driver on your computer (this is 96 dots per inch on many Windows display devices, but number can very from device to device).
Height	integer	Sets the height of the exported image in pixels. Either a Height or a VDPI option may be specified, but not both
KeepAspect	0 or 1	If this is set to 1, for any of the settings Height, Width, HDPI or VDPI you only need to set one option and Voxler will automatically calculate the other parameters in line with the aspect ratio of the input image. KeepAspect=0 -- Do not retain the aspect ratio KeepAspect=1 -- Do retain the aspect ratio
KeepPixelSize	0 or 1	Locks the pixel dimension width and height. Changes to the *VDPI* and *HDPI* will affect the document size, but not the pixel size, when this is set to 1. Changes to the *VDPI* and *HDPI* will affect the pixel dimensions, but not the document size, when this is set to 0. Note that the *KeepPixelSize* option is ignored if the either the *Width* or *Height* option is specified.
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default
VDPI	integer	Sets the number of vertical pixels in the exported image that corresponds to a distance of one vertical inch. Either a Height or a VDPI option may be specified, but not both. The default VDPI is the resolution of the display device driver on your computer (this is 96 dots

		per inch on many Windows display devices, but number can very from device to device)
Width	integer	Sets the width of the exported image in pixels. Either a Width or an HDPI option may be specified, but not both.

Example

This example shows how to export an RGBA file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "sgi")
CommandApi.Option("Options", "HDPI=96; Height=418; VDPI=96; Width=576")
CommandApi.Option("Path", "C:/Path/To/Export/file.rgba")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "0")
CommandApi.Do()
```

Used by: Export

Sun Image File Format .RAS, .SUN Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	SUN
ColorDepth	-32, -16, -8, 1, 4, 8, 24, 32	Specifies the color depth to use. ColorDepth=-32 -- 32 bit greyscale output ColorDepth=-16 -- 16 bit greyscale output ColorDepth=-8 -- 8 bit greyscale output ColorDepth=1 -- 1 bit color indexed output ColorDepth=4 -- 4 bit color indexed output ColorDepth=8 -- 8 bit color indexed output ColorDepth=24 -- 24 bit true color output ColorDepth=32 -- 32 bit true color with alpha output
ColorReductionMethod	1, 2, 3, 4, 5	Specifies the color reduction method to use. ColorReductionMethod=1 -- Use the Ordered Dither color reduction method ColorReductionMethod=2 -- Use the Diffused Dither color reduction method

		ColorReductionMethod=3 -- Use the Popularity color reduction method ColorReductionMethod=4 -- Use the MedianCut555 color reduction method ColorReductionMethod=5 -- Use the MedianCut888 color reduction method
HDPI	integer	Sets the number of horizontal pixels in the exported image that corresponds to a distance one horizontal inch. Either a Width or an HDPI option may be specified, but not both. The default HDPI is the resolution of the display device driver on your computer (this is 96 dots per inch on many Windows display devices, but number can very from device to device).
Height	integer	Sets the height of the exported image in pixels. Either a Height or a VDPI option may be specified, but not both
KeepAspect	0 or 1	If this is set to 1, for any of the settings Height, Width, HDPI or VDPI you only need to set one option and Voxler will automatically calculate the other parameters in line with the aspect ratio of the input image. KeepAspect=0 -- Do not retain the aspect ratio KeepAspect=1 -- Do retain the aspect ratio
KeepPixelSize	0 or 1	Locks the pixel dimension width and height. Changes to the *VDPI* and *HDPI* will affect the document size, but not the pixel size, when this is set to 1. Changes to the *VDPI* and *HDPI* will affect the pixel dimensions, but not the document size, when this is set to 0. Note that the *KeepPixelSize* option is ignored if the either the *Width* or *Height* option is specified.
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default
VDPI	integer	Sets the number of vertical pixels in the exported image that corresponds to a distance of one vertical inch. Either a Height or a VDPI option may be specified, but not both. The default VDPI is the resolution of the display device driver on your computer (this is 96 dots per inch on many Windows display devices, but number can very from device to device)
Width	integer	Sets the width of the exported image in pixels. Either a Width or an HDPI option may be specified, but not both.

Example

This example shows how to export an RAS file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "sun")
CommandApi.Option("Options", "HDPI=96; Height=418; VDPI=96; Width=576")
CommandApi.Option("Path", "C:/Path/To/Export/file.ras")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

Golden Software Surfer 6 ASCII Grid File Format .GRD Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	sgrd6a
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default

Example

This example shows how to export an GRD file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "sgrd6a")
CommandApi.Option("Path", "C:/Path/To/Export/file.grd")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

Golden Software Surfer 6 Binary Grid File Format .GRD Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	sgrd6b
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default

Example

This example shows how to export an GRD file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "sgrd6b")
CommandApi.Option("Path", "C:/Path/To/Export/file.grd")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

Golden Software Surfer 7 Binary Grid File Format .GRD Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	sgrd7b
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default

Example

This example shows how to export an GRD file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "sgrd7b")
CommandApi.Option("Path", "C:/Path/To/Export/file.grd")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

SYLK Spreadsheet File Format .SLK Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	SLK
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default

Example

This example shows how to export an SLK file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "slk")
CommandApi.Option("Path", "C:/Path/To/Export/file.slk")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

Targa True Vision Image File Format .TGA Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	TGA
ColorDepth	-32, -16, -8, 1, 4, 8, 24, 32	Specifies the color depth to use. ColorDepth=-32 -- 32 bit greyscale output ColorDepth=-16 -- 16 bit greyscale output ColorDepth=-8 -- 8 bit greyscale output ColorDepth=1 -- 1 bit color indexed output ColorDepth=4 -- 4 bit color indexed output ColorDepth=8 -- 8 bit color indexed output ColorDepth=24 -- 24 bit true color output ColorDepth=32 -- 32 bit true color with alpha output
ColorReductionMethod	1, 2, 3, 4, 5	Specifies the color reduction method to use. ColorReductionMethod=1 -- Use the Ordered Dither color reduction method ColorReductionMethod=2 -- Use the Diffused Dither color reduction method ColorReductionMethod=3 -- Use the Popularity color reduction method ColorReductionMethod=4 -- Use the MedianCut555 color reduction method ColorReductionMethod=5 -- Use the MedianCut888 color reduction method
HDPI	integer	Sets the number of horizontal pixels in the exported image that corresponds to a distance one horizontal inch. Either a Width or an HDPI option may be specified, but not both. The default HDPI is the resolution of the display device driver on your computer (this is 96 dots per inch on many Windows display devices, but number can very from device to device).
Height	integer	Sets the height of the exported image in pixels. Either a Height or a VDPI option may be specified, but not both
KeepAspect	0 or 1	If this is set to 1, for any of the settings Height, Width, HDPI or VDPI you only need to set one option and Voxler will automatically calculate the other parameters in line with the aspect ratio of the input image. KeepAspect=0 -- Do not retain the aspect ratio KeepAspect=1 -- Do retain the aspect ratio
KeepPixelSize	0 or 1	Locks the pixel dimension width and height. Changes to the *VDPI* and *HDPI* will affect the document size, but not the pixel size, when this is set to 1. Changes to the *VDPI* and *HDPI*

		will affect the pixel dimensions, but not the document size, when this is set to 0. Note that the *KeepPixelSize* option is ignored if the either the *Width* or *Height* option is specified.
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default
VDPI	integer	Sets the number of vertical pixels in the exported image that corresponds to a distance of one vertical inch. Either a Height or a VDPI option may be specified, but not both. The default VDPI is the resolution of the display device driver on your computer (this is 96 dots per inch on many Windows display devices, but number can very from device to device)
Width	integer	Sets the width of the exported image in pixels. Either a Width or an HDPI option may be specified, but not both.

Example

This example shows how to export an TGA file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "tga")
CommandApi.Option("Options", "HDPI=96; Height=418; VDPI=96; Width=576")
CommandApi.Option("Path", "C:/Path/To/Export/file.tga")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

Tagged Image File Format .TIF, .TIFF Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	TIF
ColorDepth	-32, -16, -8, 1, 4,	Specifies the color depth to use. ColorDepth=-32 -- 32 bit greyscale output

	8, 24, 32	ColorDepth=-16 -- 16 bit greyscale output ColorDepth=-8 -- 8 bit greyscale output ColorDepth=1 -- 1 bit color indexed output ColorDepth=4 -- 4 bit color indexed output ColorDepth=8 -- 8 bit color indexed output ColorDepth=24 -- 24 bit true color output ColorDepth=32 -- 32 bit true color with alpha output
ColorReductionMethod	1, 2, 3, 4, 5	Specifies the color reduction method to use. ColorReductionMethod=1 -- Use the Ordered Dither color reduction method ColorReductionMethod=2 -- Use the Diffused Dither color reduction method ColorReductionMethod=3 -- Use the Popularity color reduction method ColorReductionMethod=4 -- Use the MedianCut555 color reduction method ColorReductionMethod=5 -- Use the MedianCut888 color reduction method
Compress	0, 1, 2	Tells Voxler whether to compress the output tiff. 0 = Use no compression 1 = Use the packbits compression 2 = Use the deflate compression
Format	0, 1, 2, 3, 4, 5	Tells Voxler what kind of tiff export format. 0 = monolithic 1 = one row per strip 2 = 16 rows per strip 3 = 64 rows per strip 4 = 64x64 tiles 5 = 256x256 tiles
HDPI	integer	Sets the number of horizontal pixels in the exported image that corresponds to a distance one horizontal inch. Either a Width or an HDPI option may be specified, but not both. The default HDPI is the resolution of the display device driver on your computer (this is 96 dots per inch on many Windows display devices, but number can very from device to device).
Height	integer	Sets the height of the exported image in pixels. Either a Height or a VDPI option may be specified, but not both
KeepAspect	0 or 1	If this is set to 1, for any of the settings Height, Width, HDPI or VDPI you only need to set one option and Voxler will automatically calculate the other parameters in line with the aspect ratio of the input image. KeepAspect=0 -- Do not retain the aspect ratio

		KeepAspect=1 -- Do retain the aspect ratio
KeepPixelSize	0 or 1	Locks the pixel dimension width and height. Changes to the *VDPI* and *HDPI* will affect the document size, but not the pixel size, when this is set to 1. Changes to the *VDPI* and *HDPI* will affect the pixel dimensions, but not the document size, when this is set to 0. Note that the *KeepPixelSize* option is ignored if the either the *Width* or *Height* option is specified.
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default
UseTransparency	0 or 1	Use application background transparency for export. If set to 1, background is transparent.
VDPI	integer	Sets the number of vertical pixels in the exported image that corresponds to a distance of one vertical inch. Either a Height or a VDPI option may be specified, but not both. The default VDPI is the resolution of the display device driver on your computer (this is 96 dots per inch on many Windows display devices, but number can very from device to device)
Width	integer	Sets the width of the exported image in pixels. Either a Width or an HDPI option may be specified, but not both.

Example

This example shows how to export an TIF file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "tif")
CommandApi.Option("Options", "HDPI=96; Height=418; VDPI=96; Width=576; Format=3;
Compress=2; UseTransparency=1")
CommandApi.Option("Path", "C:/Path/To/Export/file.tga")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

Text Data File Format .TXT Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Delimiter	comma, tab, space, semicolon	The delimiter is the separator between columns.
Filter	string	TXT
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default
TextQualifier	doublequote, singlequote, none	The text qualifier is the mark around text, keeping text in one cell.

Example

This example shows how to export an TXT file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "txt")
CommandApi.Option("Options", "Delimiter=comma; TextQualifier=doublequote")
CommandApi.Option("Path", "C:/Path/To/Export/file.txt")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

Golden Software Voxler Data File Format .VDAT Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	VDAT
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default

Example

This example shows how to export an VDAT file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "vdat")
CommandApi.Option("Path", "C:/Path/To/Export/file.vdat")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

Visualization Toolkit File Format .VTK Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	VTK
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default

Example
This example shows how to export an VTK file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "vtk")
CommandApi.Option("Path", "C:/Path/To/Export/file.vtk")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

Microsoft Excel Spreadsheet File Format .XLS Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
FileFormat	BIFF8, BIFF5	Specifies the Excel format compatibility for the export file. BIFF8 = Excel 97 BIFF5 = Excel 95
Filter	string	XLS
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window 2=default

Example

This example shows how to export an XLS file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "xls")
CommandApi.Option("Options", "FileFormat=BIFF8")
CommandApi.Option("Path", "C:/Path/To/Export/file.xls")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

XYZ Grid File Format .DAT Export Automation Options

When the program is driven from an automation script, an options string can be specified in the script that acts as the export options dialogs do in the program. The string consists of comma-separated parameters, which specify the behavior of the various export options.

Parameter	Type	Description
Filter	string	XYZ
ModuleID	integer	Numeric ID of the module to be exported. 0=entire viewer window

		2=default

Example
This example shows how to export an DAT file with the various options.

```
CommandApi.Construct("Export")
CommandApi.Option("GuiEnabled", "False")
CommandApi.Option("ClearOptions", "False")
CommandApi.Option("Filter", "xyz")
CommandApi.Option("Path", "C:/Path/To/Export/file.dat")
CommandApi.Option("PersistOptions", "True")
CommandApi.Option("ModuleId", "2")
CommandApi.Do()
```

Used by: Export

Appendix B - Mathematical Functions

The *Math Module*, *Exclusion Filter Module*, *Function Lattice Module*, and *Transform Module* all utilize mathematical functions, as well as the **Data | Transform** command in the worksheet window.

Data Types

The expression evaluator supports 32-bit signed integer numbers, double-precision floating-point numbers, a boolean value, a text string of 0 to 256 characters, and time stamp values.

Variable Names

Variable names must begin with a column letter (i.e. A), row number (i.e. _1), or cell location (i.e. A2), which may be followed by other letters, numbers, or underscores (_), up to a maximum of 256 characters per variable name.

The variable names are not case sensitive. For example, **sum(a..z)**, **sum(A..z)**, and **sum(A..Z)** all refer to the same variable.

Precedence

The mathematical expression can consist of constants, variables (such as column letters), or functions (outlined below). The formulas follow standard precedence rules. Spaces are used in the equation for clarity.

Formulas are specified using standard precedence rules. Operators, in order of decreasing precedence, are:

( )	parentheses
-	minus (or negative sign)
* /	multiplication and division
+ -	addition and subtraction

The expression evaluator treats operators with the following precedence:

1. !, NOT, ~
2. *, /, %
3. +, -
4. <<, >>
5. <, >, <=, >=
6. ==,!=,<>
7. &
8. ^, XOR
9. |
10. &&, AND
11. ||, OR
12. ?:
13. IF

Operators of equal precedence are evaluated from left to right within the equation. Parentheses are used to override precedence, and expressions with the parentheses are performed first.

The following built-in functions are supported.

Trigonometric Functions

All trigonometric functions are carried out in radians. If the data are in degrees, use the d2r(x) conversion function (in the *Miscellaneous Functions* section below) to convert degree data to radians and then use the trigonometric functions.

sin(x)	sine of angle x
cos(x)	cosine of angle x
tan(x)	tangent of angle x, the value of x must not be an odd multiple of π/2
asin(x)	Arcsine in the range -π/2 to π/2, x must be between -1 and 1
acos(x)	Arccosine in the range 0 to π, x must be between -1 and 1
atan(x)	Arctangent in the range -π/2 to π/2
atan2(y,x)	Arctangent of y/x in the range -π to π

Bessel Functions

j0(x) j1(x) jn(n,x)	Bessel functions of the first kind at x of orders 0, 1, and n, respectively
y0(x) y1(x) yn(n,x)	Return the Bessel functions of the second kind at x, of orders 0, 1, and n, respectively. For y0, y1, and yn, the value of x must not be negative.

Exponential Functions

exp(x)	exponential function of x (e to the x)
sinh(x)	hyperbolic sine of angle x
cosh(x)	hyperbolic cosine of angle x
tanh(x)	hyperbolic tangent of angle x
ln(x)	natural logarithm (base *e*) of x, x must be positive
log10(x)	base 10 logarithm of x, x must be positive
pow(x,y)	x raised to the y^{th} power Error conditions result if: x is zero and y is negative or zero, x is negative and y is not an integer, an overflow results.

Miscellaneous Functions

min(x,y)	smaller of x and y
max(x,y)	larger of x and y
randn(x,y)	an approximately normally (Gaussian) distributed real random number with mean x and standard deviation y
randu(x)	a uniformly distributed real random number with mean of x from the interval [0,x]
row()	returns the row number

ceil(x)	next whole number greater than or equal to x
floor(x)	next whole number less than or equal to x
pi()	returns the value of Pi. To limit to a specific number of digits, use Round(Pi(),y) where Y is the number of digits after the decimal point
round(x, y)	X rounded to a the nearest number with Y digits after the decimal point
sqrt(x)	square root of x, x must not be negative
fabs(x)	absolute value of x
fmod(x,y)	floating point remainder of x/y, if y is zero, fmod returns zero
d2r(x)	convert angle x from degrees to radians. Example: sin(d2r(30)) computes the sine of 30 degrees. Sin(30) computes the sine of 30 radians
r2d(x)	convert angle x from radians to degrees.

Statistical Functions of Intervals

sum(a..z)	calculates the sum of a range of columns in a row
sum(_1.._5)	calculates the sum of a range of rows in a column
avg(a..z)	calculates the average of a range of columns in a row
avg(_1.._5)	calculates the average of a range of rows in a column
std(a..z)	calculates the (population) standard deviation of a range of columns in a row
std(_1.._5)	calculates the population standard deviation of a range of rows in a column
rowmin(a..z)	finds the minimum value of a range of columns in a row
rowmin(_1.._5)	finds the minimum value of a range of rows in a column
rowmax(a..z)	finds the maximum value of a range of columns in a row
rowmax(_1.._5)	finds the maximum value of a range of rows in a column

- The *Statistical Functions of Intervals* functions operate row-wise on an interval of columns or column-wise on an interval of rows.
- The *Statistical Functions of Intervals* do not operate on cells. The equation B1 = avg(A1..A6) will result in only the average of row 1, not the average of the specified cells.
- For example, SUM(A..Z) computes the sum of the twenty-six columns A, B, C, ..., Z. It does this for each row separately.
- Replace 'a..z' by any valid interval of columns or rows, such as W..AC or _4.._612. There must be exactly two periods between the column or row labels. The labels may be given in reverse order, such as SUM(Z..A).

String Comparison

atof(x)	convert a string x to floating point value
atoi(x)	convert a string x to an integer value
strlen(x)	length of string x in characters
strcmp(x,y)	compare string x with y and return 1 if x>y, -1 if x<y, or 0 if x=y

stricmp(x,y)	compare string x with y without regard to the case of any letters in the strings
strncmp(x,y,z)	compare the first z characters of string x with y
strnicmp(x,y,z)	compare the first z characters of string x with y without regard to the case of any letters in the strings

String comparison functions work with strings, not numbers. Any rows or columns containing numbers result in blanks. In each of the string comparison functions, 1 is returned if string x is greater than string y, -1 is returned if string x is less than string y, and 0 if string x = string y. In the three-parameter comparison functions, the third parameter, z, specifies the number of characters to compare. For example, a z value of 3 compares the x and y strings' first three characters and ignores any characters after the third.

The comparisons are based on the standard ASCII table:

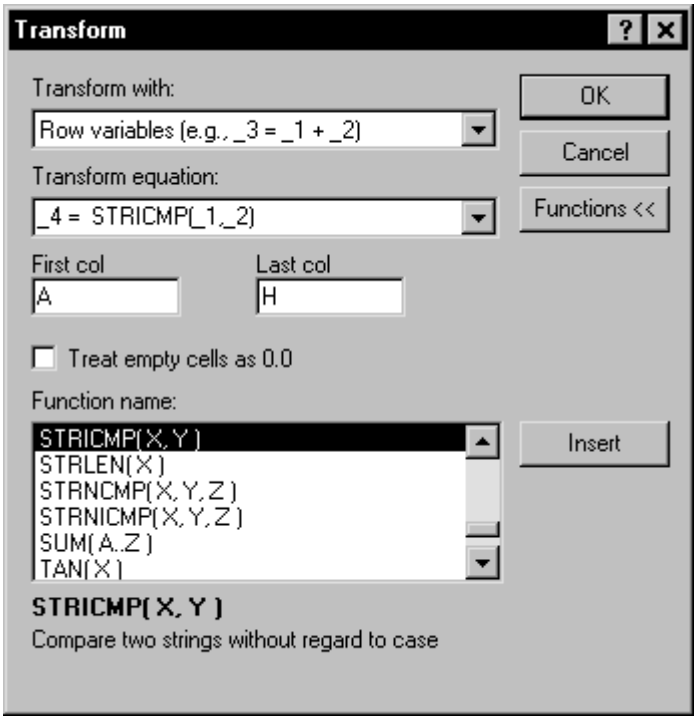
1. numeric values (disregarded in string comparisons as mentioned above)
2. cells starting with a space character
3. common punctuation
4. numeric text (numbers entered as text)
5. less common punctuation
6. uppercase letters
7. even less common punctuation
8. lower case letters
9. uncommon punctuation
10. blank cells (disregarded in string comparisons)

0	1	2	3	4	5	6	7	8	9
space	!	"	#	$	%	&	'	(	)
*	+	,	-	.	/	"0"	"1"	"2"	"3"
"4"	"5"	"6"	"7"	"8"	"9"	:	;	<	=
>	?	@	A	B	C	D	E	F	G
H	I	J	K	L	M	N	O	P	Q
R	S	T	U	V	W	X	Y	Z	[
\	]	^	_	`	a	b	c	d	e
f	g	h	i	j	k	l	m	n	o
p	q	r	s	t	u	v	w	x	y
z	{	|	}	~	blank				

This is the ASCII table order. The table is read left to right, top to bottom. Items appearing toward the upper left corner are less than the items appearing toward the lower left corner.

Example of a String Comparison

The following example shows a comparison of two strings in rows 1 and 2 with the results displayed in row 4. STRCMP returns a 1 if row 1 is greater than row 2; -1 if row 1 is less than row 2, and 0 if row 1 is equal to row 2.



This equation compares the strings in rows 1 and 2 and places the results in row 4.

	A	B	C	D	E	F	G
1	1	abc	abc	*abc	ABC	abc	abc
2	A	zyx	zyx	abc	abc		abc
3							
4		-1	1	-1	-1		0

This table includes results of the string comparisons based on the ASCII table. Cells containing numbers and blanks are ignored.

Boolean Expressions

Boolean expressions include:

- **logical Operators** (and, or, xor, not),
- **Comparison Operators** (=, <>, <, >, <=, >=),
- the **IF Function -** IF(condition, true_value, false_value)

The words AND, OR, XOR, NOT, and IF are reserved keywords and may not be used as variable names.

Logical Operators (and, or, xor, not)

SYMBOL	NAME	DESCRIPTION
AND	AND	The result is true if both operands are true.
&&	AND	The result is true if both operands are true.

!	Logical NOT	Inverts the boolean value. True becomes false, false becomes true.
NOT	Logical NOT	Inverts the boolean value. True becomes false, false becomes true.
&	AND	The result is true if both operands are true.
|	OR	The result is true if either of the two operands is true
^	Exclusive-OR (XOR)	
XOR	Exclusive-OR (XOR)	Same as ^
||	OR	The result is true if either of the two operands is true.
OR	OR	The result is true if either of the two operands is true.

Comparison Operators (=, <>, <, >, <=, >=)

SYMBOL	NAME	DESCRIPTION
~	Bitwise NOT	Inverts the bits in an integer
*	Multiply	Multiplies the two operands
/	Divide	Divides the first operand by the second
%	Remainder	Integer remainder of the first operand divided by the second.
+	Add	Adds the two operands
-	Subtract	Subtracts the second operand from the first
<<	Shift Left	Shifts the operand to the left
>>	Shift Right	Shifts the operand to the right
<	Less Than	In the example, A1 < B1, the result is true if A1 is less than B1.
<=	Less Than or Equal To	Result is true if the ordinal value of p1 is less than or equal to p2
>	Greater Than	Result is true if the ordinal value of p1 is greater than p2
>=	Greater Than or Equal To	Result is true if the ordinal value of p1 is greater than or equal to p2
==	Equal To	Result is true if the operands have identical values
!=	Not Equal To	Result is true if the operands do not have identical values
<>	Not Equal To	Result is true if the operands do not have identical values

IF Function - IF(condition, true_value, false_value)

SYMBOL	NAME	EXAMPLE	DESCRIPTION

IF	Conditional Evaluation	IF(p1,p2,p3)	IF(condition,true_value,false_value) i.e. If p1 is true, the result will be p2. If p1 is false, the result will be p3.
IF	Conditional Evaluation	p1?p2:p3	condition?true_value:false_value i.e. If p1 is true, the result will be p2. If p1 is false, the result will be p3.

Examples

The following are examples of mathematical function syntax used in **Surfer**. With **Grid | Function** and **Grid | Math** use X,Y, and Z for the variables. If you use **Data | Transform** in the worksheet, replace X,Y, and Z with column letters (A), row numbers (remember the underscore, _1 is row 1), or cell locations (A1).

Equation	Mathematical Function Syntax
x^2	pow(x,2)
$\ln(x)$	ln(x)
$\log_{10} x$	log10(x)
$1-e^{-x}$	(1-exp(-X))
$1-e^{-x^2}$	(1-exp(-X*X))
$1-\frac{\sin(x)}{x}$	1-(sin(x)/x)
$\frac{x^2}{1+x^2}$	pow(x,2)/(1+(pow(x,2)))
$2x-x^2$	(2 * X)- pow(x,2)
$(x^2+y^2)(\sin(8\times\tan^{-1}xy))$	(pow(x,2)+ pow(y,2))*(sin(8*atan(x*y)))

Index

?10g, 645
2D lattice, 91, 203, 664, 667, 674, 688, 697, 721, 730
3D lattice, 91, 135, 648, 681, 695, 696, 702, 708, 716, 730, 731
3D lines, 140
3D objects, 703
3D scene, 384
3D view, 33
3D viewer window, 22
3-minute tour, 6, 583
A Note About Transparency - Tutorial, 57
abort import, 15
about Voxler windows, 8
absorption of light, 341
accdb import options automation, 803
ACR-NEMA, 669
add a module, 153
add graphics, 221, 247
add labels, 50
add modules to network, 20
add multiple grids, 196
add multiple lattices, 196
Adding a Bounding Box - Tutorial, 48
Adding Contours - Tutorial, 60
adding wells, 118, 375
ADF, 647
ADF ESRI ArcInfo Binary Grid File Description, 647
ADF Export Options, 862
ADF Import Options, 803
adjust width, 561
Adobe Portable Document Format, 711
advanced tutorial, 45
ADX, 661
AGR, 650
AIG, 650
aliasing, 341
align objects in 3D space, 384
all modules, 22
alpha channel, 732
alpha channels, 715
AM, 648
AM export options, 863
AM Import Options, 804
ambient color, 82
ambient intensity, 82
ambient lighting, 82
Amira, 648
AMIRA export options dialog, 776
Amira mesh, 648
Amira stacked images, 652
AN?, 669, 742
AN? ACR-NEMA medical image, 649
AN1, 649, 669
AN2, 649, 669
analyze, 696
Analyze 7.5 Medical Image, 788
Analyze Export Options, 864
Analyze Import Options, 805
animate, 33
animating draw style, 388
animation, 377, 550
anisotropy, 183
annotation, 221, 247, 248
antialias, 248, 341
antialias text, 248
antialiasing, 657
Application Object, 605
Application Property, 606
apply to subsequent uniform lattices, 538, 738
ArcInfo, 650
Array Variables, 596
arrow scale, 315
as is, 385
ASC, 650
ASC Export Options, 864
ASC Import Options, 806
ASCII database, 664
ASCII Grid, 650
ASCII grid file format, 686
ASI, 652
asterisk, 15
auotmation objects, 605
auto connect, 511
auto update, 26, 37
AutoCAD DXF, 675, 749, 751
auto-calc step interval, 309
auto-hide, 10
automatic update, 26
automation, 581
automation example, 631
Automation Examples, 619
automation model, 605
automation properties, 606, 608
AutoZoom, 33, 511
average, 162
AVI, 550
AVS, 681, 732
AVS export options, 866
AVS Field, 784
AVS field files, 681
AVS Import Options, 807
AVSX export options, 866
AVSX files, 732
AVSX Import Options, 808
axes, 221, 226, 229, 232, 247, 309

axial, 206, 269, 274, 295, 327, 330, 333
axis, 309
axis triad, 87, 380
azimuth, 364
Background, 262, 277, 287, 301, 321, 336, 344, 355, 561
background color, 82
bad component value, 96, 525
basic feature tutorial, 45
BASIC language, 590
Bessel, 915
BIN, 718, 795
binary grid, 647
binary lattice import options, 763
bitmap, 657, 699, 704, 710, 715, 717, 719, 729
blank a lattice, 348
blank handling, 162
blanked data, 91
blanking, 91
blanking data, 761, 793
BLN, 655
BLN Export Options, 869
BLN Import Options, 809
blue connection line, 30
BMP, 657
BMP Export Options, 870
BMP Import Options, 809
BNA Export Options, 872
BND Export Options, 872
BND Import Options, 810
bold text, 3
Boolean expressions, 915
Boolean functions, 158
boundary objects, 709
bounding, 162
bounding box (no depth), 385, 388
bounding box labels, 239
BoundingBox, 221, 247
break connection, 27
brightness, 573
brightness & contrast, 162
BW, 719
camera, 33, 86, 87, 379, 402
camera properties, 86, 402
can I create a gridder lattice using the nearest neighbor method?, 191
can I create an ASCII XYZC lattice file?, 127
cancel in-progress connection, 30
cancel in-progress module connection, 30
capture video, 550
Capturing a Video - Tutorial, 65
CDX, 664
center of rotation, 33
center position, 309
central moment, 162
change the input file, 96
change the scale of one of the axes, 235
change the size of a window, 14
change the type of the data components, 154
change x axis, 226
ChangeType, 153, 154
changing a window's location, 9
changing from docked to floating windows, 10
Changing Symbol Colors - Tutorial, 50
Changing the Bounding Box Properties - Tutorial, 51
changing the isosurface properties - tutorial, 55
Changing the Transparency - Tutorial, 62
changing the window layout, 10
changing well display, 118, 375
check box, 24, 37
choosing gridding method, 186
clear undo history, 566
clip plane, 240
clip plane normal, 240
clip planes, 240
ClipPlane, 221, 240, 247
closing a window, 9
CLR, 573
coefficient of variation, 162
COL, 648
collapse folder, 20
color, 573
color blending, 393
color graph, 573
color mapping, 573
color method, 255, 283, 309, 315
color palette, 568
colormap, 341, 572, 573
colormap editor, 573
colors, 341, 515, 568, 572, 573
CommandApi Object, 607
CommandApi Property, 608
commands, 2
Comments, 591
compare data, 139
component, 538, 738
component 1 limits, 75
component limits, 102, 122
component type, 75, 102, 122
components to filter, 162
composition, 341
compuserve, 683
computational, 22, 24, 153
computational modules, 17, 73, 89, 151, 153, 219, 245
compute paths, 114
connect, 27
connect modules, 27
connected module, 30
connecting modules, 24, 30
connection pads, 24
connection style, 511
connections, 731
ConnectModules, 614
constant time interval, 309

Construct Method, 608
context menus, 24
contour map, 281
contours, 269
contrast stretch, 154
control definition of duplicate point, 156
controls, 37
convolution kernel, 162
coordinate limits, 122
coordinate loading, 657
copy module, 566
copy snapshot, 550
Copying a Snapshot - Tutorial, 64
coronal, 206, 269, 274, 295, 327, 330, 333
CPS, 661
CPS Import Options, 812
CPS-3 Grid, 661
create a new lattice, 133
create an isosurface from well data, 119, 374
create lattice, 191
create lattice data, 135
CreateModule, 611
Creating a Scatter Plot - Tutorial, 48
Creating an Isosurface - Tutorial, 54
Creating Dialogs, 600
creating stacked and tabbed windows, 10
CSV, 662
CSV Export Options, 873
CSV Import Options, 813
cube, 141
cube render, 351
curvilinear, 91
curvilinear lattice, 91, 538, 648, 681, 695, 708, 716, 730, 731, 738
custom, 568
custom colors, 568
custom labels, 262, 277, 287, 301, 321, 336, 344, 355
custom lighting, 75
custom mouse settings, 516
custom text, 248
custom view dialog, 86, 402
customize mouse buttons, 33
cutting plane, 203, 206, 269, 274, 330, 333
cylinder, 141
dabase, 664
DAT, 661, 662, 673, 734
DAT Export Options, 874
DAT Import Options, 814
data, 793
data duplication, 156
data export options dialog, 772
data flow model, 22
data from logs, 755
data geometry, 132
data import options, 96, 525
data import options dialog, 99, 735
data max, 266, 281, 291, 573
data min, 266, 281, 291, 573
data point exclusion, 158
data sets, 22
data source, 133
data source modules, 17, 73, 89, 151, 219, 245
data types, 91
data view, 139
data-dependent graphics, 221, 247
dataset type, 75, 102, 122
date, 248, 251
DBF, 664
DBF Import Options, 816
DCM, 649, 669, 742
DDF, 664, 665
Debugging Scripts, 587
default camera, 384
default margins, 554
default path, 511
default z, 538, 738
defined views, 381
Defining Object Properties and Methods, 599
definition of a duplicate point, 156
deflate, 797
delete, 567
delete all modules, 567
delete module, 566
deleting wells, 118, 375
delimiters, 99, 735
DEM, 306, 645, 664, 667, 778, 797
DEM Export Options, 875
DEM Import Options, 816
demo, 45
density, 255
deselect module, 24
dialog, 126, 540
dialogs, 2
DIC, 649, 669
DICOM, 649, 669, 742
DICOM Import Options, 817
difference between gridding methods, 186
difference between lattices, 125
different direction weighting while gridding, 183
digital imaging, 649
digital line graph, 670
digital line graph import, 744
digital terrain elevation data, 674
dimensions, 75, 102, 122
dip well, 364
direct volume rendering, 341
direction, 309
direction information, 364
directional data, 364
directional light, 79, 87, 379
disable graphics, 385
disconnect, 27
disconnect modules, 30
display cubes, 351
display labels, 50

display labels on scatterplot, 259
display legend, 262, 277, 287, 301, 321, 336, 344, 355
display module information, 75
display wells, 115, 361, 371
display worksheet data, 139
displaying log values, 68
displaying logs, 68
displaying wells, 67
distance, 162
distance from center, 240
distance 'inf norm', 162
distribution of velocity, 309
DLG, 670, 744, 748
DLG Import Options, 818
Do Method, 618
docked windows, 9, 10
docking mechanism, 9, 10
document tab, 14
documentation, 3
DoOnce Method, 618
DOS, 673
Double Quotes and Text, 591
dragger, 79, 203, 206, 211, 240, 269, 274, 309, 330, 333, 377, 403
draw border, 269
draw lines, 255
draw style, 33, 128, 283, 295, 385, 388
drawing contours, 281
drop-down list, 37
DT0, 674
DT1, 674
DT2, 674
DTED files, 674
DTED Import Options, 821
duplicate filter, 156
duplicate points - tutorial, 57
DuplicateFilter, 153, 156
DXF, 675, 749, 751
DXF Import Options, 821
E00, 678, 752
E00 Import Options, 822
ecw, 679
ecw dialog, 753
ecw file import, 679
ecw import, 753
edge detection, 162
edge handling, 162
edit boxes, 37
edit menu commands, 565, 566
elevation data, 645, 664, 667, 674
elevation grid, 688
enable AutoZoom, 511
Encapsulated PostScript, 680, 782
encoding, 761, 793
endianess, 763, 795
enhanced compression wavelet, 679
EPS, 680, 782
EPS Export Options, 877
er mapper, 679
ER Mapper Grid, 680
ERS, 680
ERS Export Options, 879
ERS Import Options, 824
ESRI, 647
ESRI ArcInfo Export Format, 678, 752
ESRI Binary Float Grid, 682
ESRI shapefile, 722
ETOPO5 Import Options, 824
example math module, 196
example of heightfield, 305
example of wellrender, 118, 375
example script, 599, 602, 619, 623, 627
example scripts, 619
examples, 20, 22
Excel, 770, 771, 800
Excel files, 733
exclude data points, 158
ExclusionFilter, 158
expand folder, 20
expanded, 162
exponential functions, 915
export, 546, 772, 789, 793, 795, 797
Export Automation, 616
export formats, 861
export options dialog, 772, 789, 793, 795, 797
extents of gridder, 179
extract data from well, 160
extract points, 160
extract points well data, 119, 374
extractpoints, 160
eye dropper, 568
face render geometry, 354
facerender, 351
facerender geometry, 354
FAQ, 567
field format, 99, 735
File Info, 748
fill, 162
fill between two heightfields, 70
filter, 153, 162
filter data points, 156
filter references, 162
filter type, 162
filtering data - tutorial, 57
fit to window, 377
FLD, 681, 784
float, 688, 716, 721
floating windows, 9, 10
Flow Control, 591
FLT, 682
FLT Export Options, 880
FLT Import Options, 825
flush undo history, 566
fog, 82
folder behavior, 20
foreshortening, 384

format document for printing, 554
formation tops, 293
FPS, 15
frame rate, 550
frequency, 315
function, 135
function expression, 133
function lattice geometry, 138
FunctionLattice, 133
functions, 915
gamma correction, 162
Gauss, 162
Gaussian, 162, 915
general, 75, 511
general modules, 17, 73, 75, 89, 151, 219, 245
general preferences, 511
generate lattices, 133
GEO Import Options, 825
geographix, 734
geologic model, 70, 293
geologic surfaces, 292
geometric data, 713
geometric figures, 648, 731
geometric shapes, 713
geometry, 91
geometry facerender, 354
geometry for combined lattices, 199, 202
geometry for gridder module, 179
geometry for merge, 199, 202
geometry function lattice, 138
geometry info, 78
geometry lattice, 124
geometry math module, 195
geometry of lattice, 173
geometry points, 132
geometry properties info module, 78
geometry properties lattice module, 124
geometry source modules, 128
geometry subset, 210
geometry test lattice, 138
Geosoft binary grid, 687
Geosoft GRD Export Options, 881
geotiff, 729
getting help from the help menu, 2
GGF, 687
GI, 687
GIF, 683
GIF Export Options, 882
GIF Import Options, 826
global land one-km base elevation project, 645
Global Variables, 597
Globe DEM, 645
GLOBE Import Options, 827
Golden Software Blanking, 655
Golden Software Boundary, 692
Golden Software Interchange, 693
Golden Software PlotCall, 712
gradient, 153, 173
gradient field, 173
graphics, 221, 247
graphics interchange file, 683
graphics output, 22, 221, 247
graphics output modules, 17, 73, 89, 151, 219, 245
grayed out text, 37
GRD, 306, 650, 661, 686, 687, 688, 689, 690
green indicator LED, 24
Grid Exchange, 693, 786
grid file, 689, 690
grid format, 647, 761
grid surface, 292
gridder, 153, 186
gridder anisotropy, 183
gridder example, 190
gridder extents, 179
gridder geometry, 179
gridder lattice, 191
gridder search, 181
gridder weights, 183
gridding anisotropy, 183
gridding data, 190
Gridding Data - Tutorial, 53
gridding example, 190
gridding method, 186
gridding well data, 119, 374
GSB, 692
GSB Import Options, 828
GSI, 693
GSI Import Options, 828
GTOPO-30 files, 697
GTOPO30 Import Options, 829
GXF, 693, 786
GXF Export Options, 884
GXF Import Options, 830
hardware, 1
haze, 82
HDF, 695, 787
HDF Export Options, 885
HDF Import Options, 830
HDR, 682, 696, 697, 788
header bytes, 795
headlight, 33, 87, 379
HeightField, 69, 221, 247, 295, 306
heightfield example, 305
heightfield overlay, 298
help, 2
help menu, 2
help messages, 15
help topics, 2
hidden, 2, 162, 341
hidden line, 385
hide a window, 9
hide status bar, 15
hiding windows, 10
hierarchical data format, 695, 787
hierarchy, 605

home, 402
home position, 33, 86, 402
how can I calculate volume?, 291, 359
how can I change the data min and max in the legend?, 266, 281, 291
how can I create a lattice from an XYZC data file and save the lattice data?, 191
how can I create a surface from my XYZ data file, 192, 306
how can I display 3D lines?, 140
how can I display geometric shapes?, 141
how can I model geologic surfaces?, 292
how data is handled, 153
how do I blank a lattice above an irregular topography surface?, 348
how do I create an isosurface with different colors above and below a GRD surface?, 292
how do i create streamlines, 309
how do I use interval data for formation tops or ore grade?, 293
how does the transform scale change the xyz coordinates, 211
how to display wells, 67, 115, 371
how to grid data, 190
how to use heightfield, 305
HSL, 573
hue, 568, 573
I attached a HeightField to a 2D curvilinear lattice and when I change the scale from 0, the surface is distored., 306
Idrisi raster image, 720
if function, 158
ignore, 162
image, 96, 525, 657, 699
image coordinate, 657
image export options, 789, 797
image overlay, 298
image quality, 561
image stacks, 96, 525
IMG, 696, 720, 788
import abort, 15
Import Automation, 609
import downhole data, 755
import ecw, 753
import ecw file, 679
import ecw image, 679
import er mapper image, 753
Import File Types, 801
import grid file, 69
import las, 755
import lattice, 538, 738
import options dialog, 99, 538, 540, 735, 738, 763, 766
import progress, 15
import well data, 755
import welldata, 115, 371
importing additional wells, 118, 375
importing log data, 67
importing well collars, 66
importing well log data, 67
importing well path, 67
increasing number of nodes, 125
indicator LED, 24, 26
indicies, 126
info, 75, 702
info module, 78
info module geometry, 78
inhomogeneity, 341
input, 27
input component, 173, 269
input range, 154
input type, 154
integration of the vector field, 309
intensity, 573
interp, 573
interpolate, 330
interquartile range, 162
interval data, 68
introduction, 1
introduction to modules, 17, 73, 89, 151, 219, 245
introduction to scripter, 581
introduction to tutorials, 45
introduction to Voxler, 1
invalid data, 96, 525
Inventor files, 703
inverse, 162
Iris Explorer, 791
Iris Explorer files, 708
irregular topography, 348
isosurface, 221, 247, 283, 292
isovalue, 283
italic text, 3
IV, 128, 140, 141, 703
IV Export Options, 886
IV Import Options, 831
ix, 126
jack, 135
JPEG, 704
JPEG image export options dialog, 789
JPG, 704, 789
JPG Export Options, 886
JPG Import Options, 831
justification, 248, 251
jy, 126
keep, 156
kernel size, 162
key, 262, 277, 287, 301, 321, 336, 344, 355
keyboard commands, 20, 24, 37
knotted rope, 135
knotted torus, 135
kz, 126
label, 248
label format, 261
labels, 255
labels bounding box, 239
labels scatter plot, 259

landscape, 554
Laplacian, 162
Laplacian edge detect, 162
LAS, 706, 755
LAS automation import, 832
LAS files, 706
LAS import, 755
LAS import options, 832
LAT, 708, 791
LAT Export Options, 890
LAT Import Options, 835
lattice, 96, 127, 133, 135, 173, 191, 348, 525, 648, 664, 667, 669, 674, 681, 688, 695, 696, 697, 702, 708, 716, 721, 729, 730, 731
lattice geometry, 124, 138
lattice import, 538, 738
lattice size, 125
lattice source modules, 122
lattice view dialog, 126
LatticeMath, 154
lattices, 91, 122
Layer Name, 751
layout, 10, 37
LCS, 702
leave alone, 162
LED, 24, 26
legend, 255, 262, 266, 269, 277, 281, 283, 287, 291, 301, 309, 315, 321, 327, 330, 336, 341, 344, 355, 572
LEICA, 702
Leica confocal scanning files, 702
Leica Import Options, 835
level ends, 573
level method, 269
levels of undo, 511
LGO, 744, 748
LGS, 744, 748
light, 79, 87, 327, 379
light type, 79
lighting, 327, 330
limits subset module, 210
Line Continuation, 591
linear convolution, 162
LizardTech MrSID, 726, 769
load colormap, 573
load data, 96, 525
load LAS, 706
load LAS data, 706
loading, 96, 525
loading well data logs, 67
log path, 517
low res points (no depth), 388
low res wireframe (no depth), 388
low resolution, 385, 388
lum, 568
luminance, 568, 573
make connection, 27
MapInfo interchange format, 709
mapping method, 327, 330
math, 153
math geometry, 195
math limits, 195
math module, 70, 196
math module examples, 196
math module limits, 195
mathematical functions, 915
maximize window, 15
maximum, 162
maximum dilation, 162
MDB, 710, 740
MDB Import Options, 836
MDX, 664
mean removal, 162
measure distances, 384
median, 162
median difference, 162
medical image, 649
medical image data sets, 669
medican image data sets, 696
memory, 1, 154
memory for lattice, 125
menu bar, 8, 15
menu commands, 15, 565
merge, 153, 197
merge geometry, 199, 202
merge lattice geometry, 199, 202
merge limits, 199, 202
Metamorph, 728
Microsoft Access, 710, 740
Microsoft Excel spreadsheet files, 733
MIF, 709
MIF Import Options, 837
minimum, 162
minimum erosion, 162
miscellaneous functions, 915
model, 292
model chart, 605
modifying wells, 118, 375
ModifyModule, 613
module, 20, 24, 30, 75, 79, 82, 96, 102, 122, 126, 128, 133, 135, 139, 153, 154, 156, 158, 162, 173, 197, 200, 203, 206, 208, 211, 221, 240, 247, 248, 251, 255, 269, 274, 283, 295, 309, 315, 327, 330, 333, 341, 915
module connection, 27
module disconnection, 27
module library, 8, 20, 22
module name, 567
Module Properties, 598
Module Types, 597
modules, 17, 20, 22, 24, 30, 73, 89, 151, 219, 245, 731
moment statistics, 162
motion, 33
mouse, 516
mouse wheel, 377

move an existing connection, 30
move an existing module, 30
move module, 24
movie, 550
MPR, 330
MrSID, 726, 769
multiline text, 248
multi-planar reconstruction, 330
multiple, 96, 525
multiple input connections, 30
multiple output connections, 30
name of module, 567
NDX, 664
nearest neighbor, 191, 341
NEMA, 669
NEMA Import Options, 837
network window, 8, 24, 27
new features, 42
new name, 567
no texture, 385, 388
no wells appear, 115, 371
NOAA, 645
node spacing, 795
nodes, 133
nonlinear, 162
normal, 240
normal direction, 206, 274, 309, 333
note about documentation, 3
null, 96, 525
number of components, 75, 102, 122
number of digits, 261
number of images, 75
number of levels, 269
number of lines, 75
number of points, 75
number of seeds, 309, 313
number of text, 75
number of triangles, 75
Nx, 133, 135
Ny, 133, 135
Nz, 133, 135
Object Variables, 596
ObliqueImage, 221, 247, 330
offset, 154
offset from center, 206, 274, 330, 333
offset the plane, 206, 274, 333, 403
online help, 2
online tutorial, 45
opacity, 33, 295, 327, 330, 341, 393, 715
opacity mapping, 573
opaque, 573
open dialog, 536
open Inventor, 703
open network, 536
opening and closing a window, 9
operating system, 1
operators, 158, 591
Optional Arguments, 592
order statistics, 162
ore grade, 293
orientation, 126, 206, 240, 269, 274, 295, 315, 327, 330, 333, 554
origin, 248
original data, 96
orthographic, 384
orthographic projection, 384
OrthoImage, 221, 247, 327
other, 162
output, 27
output dimensions, 135
output lattice, 162
output limits, 135
output range, 154
output type, 133, 135, 154
overlap, 561
overlap method, 197
overlay heightfield, 298
overlay image, 298
overlay surface, 298
overview, 7
P3D, 127, 716, 761, 793
P3D Export Options, 891
P3D Import Options, 838
packbits, 797
page setup, 554
pan, 33, 379
parallel projection, 384
paste module, 566
path, 511
path calculation, 114
path method, 114
path well, 370
PBM, 716
PCX, 710
PCX Export Options, 892
PCX Import Options, 839
PDF, 711
PDF Export Options, 893
perspective, 384
perspective projection, 384
PGM, 716
pipeline, 22
plane, 240
Plot-3D, 127, 716
PLOT3D export options dialog, 793
Plot3D import options dialog, 761
PlotCall, 712
PLT, 712
PLY, 141, 713
PLY Import Options, 841
PNG, 715
PNG Export Options, 895
PNG Import Options, 842
PNM, 716
PNM Export Options, 897
PNM Import Options, 842
point data, 648, 695, 730
point import, 96, 525

point label, 255
point light, 79
point set, 731
point sets, 91
point source modules, 102
points, 385, 388
points geometry, 132
Portable Any Map, 716
portable network graphics files, 715
portrait, 554
postfix, 261
PPM, 716
preferences - colors, 515
preferences - general, 511
preferences - mouse, 516
prefix, 261
prevent objects from spinning, 404
prevent spinning, 33
Prewitt, 162
print, 561
printer, 554
prism, 141
Private and Public Definitions, 598
Program Statements, 590
progress, 15
projection, 33, 384
properties, 37, 511
properties window, 8, 37
Q, 127, 716
quality, 327
quartile, 162
radius search, 181
RAM, 1
range, 162
ranges, 133
rank, 162
RAS, 717
raster graphic, 550
RAW, 718, 795
RAW Binary Lattice, 718
RAW Export Options, 899
RAW export options dialog, 795
RAW Import Options, 843
RAW import options dialog, 763
read data from logs, 755
read LAS, 755
read LAS data, 706
read well data, 706
rectangular, 162
rectilinear, 91
rectilinear lattice, 91, 648, 681, 695, 708, 730, 731
red indicator LED, 24
redo, 565, 566
references, 135, 162
reloading data, 96
relocate geometry, 377
remove duplicate data, 156
rename module, 567
render, 33, 388
render geometry, 385
render method, 341
require duplicates points to have same coordinates, 156
requirements, 1
resample, 153, 200
resample method, 197
rescale now, 154
reset camera position, 402
reset rotation, 402
reset undo history, 566
reset zoom, 402
resize window, 14
resizing a window, 9
resolution, 133, 330, 763
restart required, 511
restore window, 15
reverse action, 565
reverse orientation, 315
RGB, 568, 573, 719
RGBA, 719
right-click menu, 24
rotatable trackball, 206, 274, 333
rotate graphics, 377
rotate object, 33
rotate the dragger, 403
Rotating Graphics - Tutorial, 52
rotation, 215, 377, 393
rotation axis, 206, 274, 333
RST, 720
RST Import Options, 845
runge-kutta algorith, 309
Runge-Kutta algorithm, 309
Running Scripts, 585
Running Scripts from the Command Line, 586
sagittal, 206, 269, 274, 295, 327, 330, 333
same as still, 388
sat, 568
saturation, 568, 573
save colormap, 573
save data, 542, 793
save lattice data, 191
save memory, 154, 566
save network, 541
save network as, 541
Saving a Graphic - Tutorial, 64
Saving a Network - Tutorial, 63
Saving Data - Tutorial, 64
Scale, 154, 211, 215, 295, 306
scale factors, 309
scale method, 315
scale size of trackball, 206, 274, 333, 403
scaling transformation equation, 154
scatter plot labels, 50
ScatterPlot, 221, 247, 255
scatterplot labels, 259
scene, 22, 381
screen color, 82

screen size, 15, 550
script example, 599, 602, 619, 623, 627
script examples, 619
Scripter, 581
scripter windows, 581
SDTS, 667
SDTS Digital Elevation Model, 667
SDTS files, 664
SDTS Topological, 665
SDTS Topological Vector Profile, 742
SDTSDEM Import Options, 846
SDTSTVP Import Options, 846
search, 181
search for gridding, 181
seed location, 313
seed method, 309, 313
seed plane, 309
seeds streamlines, 313
SEGY, 721, 766
SEGY Import Options, 847
SEGY import options dialog, 766
SEG-Y seismic data log files, 721
seismic data log files, 721
select data columns dialog, 102
select format dialog, 540
select module, 24
select slices, 797
set function lattice limits, 138
set home, 402
set label properties bounding box, 239
set seeds for streamlines, 313
set y axis properties, 229
set z axis properties, 232
setting the search options while gridding, 181
SGI, 719
SGI Export Options, 899
SGI Import Options, 849
SGI Open Inventor files, 703
SGRD Import Options, 849
SGY, 721, 766
shapes, 713
show a window, 9
show all modules, 22
show axes triad, 511
show dragger, 79, 206, 240, 269, 274, 309, 330, 333, 403
show frames per second, 511
show labels bounding box, 239
show legend, 262, 277, 287, 301, 321, 336, 344, 355
show property help, 511
show status bar, 15
showing all modules, 20
showing windows, 10
SHP, 722
SHP Import Options, 850
SID, 726, 769
SID Import Options, 851
side to draw, 283
Silicon Graphics RGB image files, 719
size, 554
Size and Color Page, 773
size of window screen, 554
skip leading spaces, 99, 735
slice, 153, 203, 797
slice number, 295, 327
slider, 37
SLK, 727
SLK Import Options, 853
smoke, 82
smooth, 393
smooth normals, 283
Sobel max, 162
Sobel norm, 162
Solaris, 717
source, 554
source data, 731
source modules, 96, 102, 122, 128
source of raw data, 133
spacing, 75, 102, 122
specify data columns, 96, 525
sphere, 141
spherical, 162
spherical density, 135
spin, 33, 377
spinning, 33
spinning graphics, 33, 388
split multi-plane lattices, 797
spot light, 79
spreadsheet, 727
Stacked Import Options, 853
stacked windows, 10
standard colors, 568
standard deviation, 162
Stanford polygon files, 713
start import at row, 99, 735
starting path, 511
static text, 37
statistical functions, 915
status bar, 8, 15
step interval, 309
steps, 309
still draw style, 385
STK, 728
STK Import Options, 854
stop spinning, 33
streamline seeds, 313
StreamLines, 221, 247, 309
string of text, 248
Subroutines and Functions, 593
subset, 153, 208
subset geometry, 210
subset limits, 210
suffix, 261
Suggested Reading - Scripter, 604
SUM, 915
SUN, 717
SUN Export Options, 901

Sun Import Options, 854
Sun raster files, 717
surface, 192, 306
surfaces, 292
Surfer 6 grid file format, 689
Surfer 7 grid file format, 690
Surfer ASCII grid, 686
Surfer GRD Export Options, 903, 904
Surfer grid files, 688
swap clip direction, 240
SYLK Export Options, 905
SYLK spreadsheet files, 727
symbol, 255
system requirements, 1
tab document, 14
tabbed documents, 14
tabbed windows, 10
tabs, 14
tabular data, 727
tagged image file format, 729
target position, 86, 402
test lattice geometry, 138
testing, 135
TestLattice, 133, 135
testlattice geometry, 138
TestLattice references, 135
text, 221, 247, 248, 251
text qualifiers, 99, 735
TGA, 728
TGA Export Options, 905
TGA Import Options, 855
The '#Uses Line, 598
the module library, 20
the network window, 24
the properties window, 37
the viewer window, 33
the visualization network, 22
three-dimensional lattice, 96, 525
three-minute tour, 6, 583
threshold averaging, 162
threshold crossing, 162
TIF, 729, 797
TIF Export Options, 907
TIF Import Options, 856
TIFF, 729, 797
TIFF image export options, 797
time, 248, 251
title bar, 8, 15
tolerance, 156
toolbars, 8
topographic surface, 348
tops, 293
tour, 6, 583
trackball, 33, 206, 274, 333, 377, 403
transfer graphics, 550
transform, 153, 211
transform rotation, 215
transform scale, 215
transform translation, 215
translation, 215
transparency, 33, 295, 327, 330, 341, 393, 573, 715
transparency type, 393
transparent, 573
treat consecutive delimiters as one, 99, 735
trigonometric functions, 915
trilinear, 341
trim excess white space from non-numeric values, 727
Truevision Targa, 728
Turk 1998, 713
tutorial, 45, 66, 67, 68
Tutorial Completion, 72
TVP, 665, 742
TXT, 662
TXT Export Options, 909
TXT Import Options, 856
Type Library References, 590
types of data, 91
UI look, 511
unconnected module, 30
undo, 511, 565, 566
undo levels, 511
uniform lattice, 91, 538, 648, 664, 667, 669, 674, 681, 688, 695, 696, 697, 702, 708, 721, 729, 730, 731, 738
uniform lattice import, 69
uniform lattice import options, 538, 738
UNIX, 717, 732
unrecognized, 540
unsaved changes, 15
unsharp mask, 162
update network, 26
update now, 37
update the network, 96
updating network, 22
use context submenus, 511
user interface, 7, 8
user-defined rotation axis, 206, 274, 333
User-Defined Types, 597
UserDialog Items, 601
USGS, 667, 670, 744
USGS DEM files, 667
USGS Digital Elevation Model, 778
USGS Digital Line Graph, 748
USGS ETOPO5, 673
using scripter help, 603
using the wellrender module, 118, 375
Variables, 595
variance, 162
VBA, 604
VDAT, 730
VDAT Export Options, 910
VDAT Import Options, 858
vector, 315
vector style, 315
vector vortex, 135
VectorPlot, 221, 247, 315

vectors, 315
velocity, 309
video, 550
Video Capture - Advanced Suggestions, 552
video card, 1
view data, 75, 102, 126, 139
viewer window, 8, 33, 82
views, 381
visibility, 377
vista, 554
visual basic, 604
visual tool kit, 799
visualization, 1
visualization network, 22, 24, 33
Visualization Toolkit files, 731
VolRender, 221, 247, 341
volume, 291, 359
volume rendering, 341
VOXB, 536, 731
voxels, 341
Voxler data files, 730
Voxler documentation, 3
Voxler help, 2
Voxler network files, 731
Voxler user interface overview, 7
Voxler windows, 10, 20, 24, 33, 37
VTK, 731, 799
VTK Export Options, 911
VTK Import Options, 858
well azimuth, 364
well data, 114, 120, 706
well data direction, 364
well data directional information, 364
well data gridder, 119, 374
well data path, 370
well data points, 160
well data to isosurface, 119, 374
well dip, 364
well display, 361
well path, 67, 370
well path color, 370
well paths, 114
well trace, 370
well trajectory, 67
welldata, 115, 371
welldata automation example, 631
welldata example, 631
wellrender, 115, 361, 371
wellrender example, 118, 375
what's new, 42
wheel mouse, 378, 516
when I attach a HeightField to a DEM or GRD file, how do I set the Scale so the HeightField matches the Z axis?, 306
who uses Voxler, 1
wiffle cube, 135
window background, 82
window layout and control, 9, 10, 14, 15
window size, 15, 550, 554
windows, 10
wireframe, 385, 388
wireframe overlay, 385
working in the network window, 24
working with scripts, 583
working with Voxler windows, 9
worksheets, 733
world axis triad, 33, 87, 380
writing scripts, 584
WS, 554
X, 732
X axis, 226
x axis settings, 226
x limits, 75
Xbase, 664
XIMG, 732
XLS, 733, 771, 800
XLS Export Options, 911
XLS Import Options, 859
XLSX, 770
XLSX Import Options, 860
x-ray image, 649
XYZ, 716, 761, 793
XYZ Export Options, 912
XYZ Grid, 734
XYZC, 91
Y axis, 229
Y axis properties, 229
y limits, 75
yellow connection line, 30
yellow indicator LED, 24
YUV, 729
Z axis, 232
z axis properties, 232
z dimension, 688
z limits, 75
zero crossing, 162
ZMAP, 734
ZMAP Import Options, 860
zones, 761, 793
zoom, 378
zoom in, 378
zoom in/out, 378
zoom out, 378
ZSoft/Paintbrush bitmap files, 710

Before calling, please check the following available resources as your question may already be answered.

Registration: Register online at www.goldensoftware.com or fax to the number below.
Knowledge Base: www.GoldenSoftware.com/knowledge-base or in the **Voxler** program using the **Help | Golden Software on the Web | Knowledge Base** command
Forums: www.goldensoftware.com/forum or in the **Voxler** program using the **Forums** button in the Online Help.
Frequently Asked Questions: In the **Voxler** program using the **Help| Golden Software on the Web | Frequently Asked Questions** command
Tutorial: Complete the tutorial section in this quick start guide or in the **Voxler** program using the **Help | Tutorial** command
Online Help: In the **Voxler** program using the **Help | Contents** command
Support Videos: www.GoldenSoftware.com/products/voxler#training-videos for **Voxler** specific training videos and webinars

Business Hours

Technical Support:

Monday through Friday, 8:00 AM - 5:00 PM, Mountain Time

Product Sales:

Online orders available 24 hours, 7 days a week

Golden Software Contact Information

www.GoldenSoftware.com
VoxlerSupport@goldensoftware.com
phone: 303-279-1021
fax: 303-279-0909

www.goldensoftware.com

Golden Software, LLC
809 14th Street
Golden, Colorado 80401 USA

Phone +1 303 279 1021
Fax +1 303 279 0909

www.goldensoftware.com

© Copyright 2015 Golden Software, LLC. All Rights Reserved. Golden Software®, the Golden Software® mark, Voxler® and the Voxler® logo marks are owned by Golden Software and may be registered. All other trademarks and logos are the property of their respective companies. Software developed in the USA.