IDL

What’s New in IDL 6.4

IDL Version 6.4
April 2007 Edition

0407IDL64WN

Restricted Rights Notice

The IDL®, ION Script™, ION Java™, IDL Analyst™, ENVI®, and ENVI Zoom™ software programs and the accompanying procedures, functions, and documentation described herein are sold under license agreement. Their use, duplication, and disclosure are subject to the restrictions stated in the license agreement. ITT Visual Information Solutions reserves the right to make changes to this document at any time and without notice.

Limitation of Warranty

ITT Visual Information Solutions makes no warranties, either express or implied, as to any matter not expressly set forth in the license agreement, including without limitation the condition of the software, merchantability, or fitness for any particular purpose.

ITT Visual Information Solutions shall not be liable for any direct, consequential, or other damages suffered by the Licensee or any others resulting from use of the software packages or their documentation.

Permission to Reproduce this Manual

If you are a licensed user of these products, ITT Visual Information Solutions grants you a limited, nontransferable license to reproduce this particular document provided such copies are for your use only and are not sold or distributed to third parties. All such copies must contain the title page and this notice page in their entirety.

Export Control Information

This software and its associated documentation are subject to the controls of the Export Administration Regulations (EAR). It has been determined that this software is classified as EAR99 under U.S. Export Control laws and regulations, and may not be re-transferred to any destination expressly prohibited by U.S. laws and regulations. The recipient is responsible for ensuring compliance to all applicable U.S. Export Control laws and regulations.

Acknowledgments

ENVI® and IDL® are registered trademarks of ITT Corporation, registered in the United States Patent and Trademark Office. ION™, ION Script™, ION Java™, and ENVI Zoom™ are trademarks of ITT Visual Information Solutions.

Numerical Recipes™ is a trademark of Numerical Recipes Software. Numerical Recipes routines are used by permission.

GRG2™ is a trademark of Windward Technologies, Inc. The GRG2 software for nonlinear optimization is used by permission.

NCSA Hierarchical Data Format (HDF) Software Library and Utilities. Copyright © 1988–2001, The Board of Trustees of the University of Illinois. All rights reserved.

NCSA HDF5 (Hierarchical Data Format 5) Software Library and Utilities. Copyright © 1998–2002, by the Board of Trustees of the University of Illinois. All rights reserved.

CDF Library. Copyright © 2002, National Space Science Data Center, NASA/Goddard Space Flight Center.

NetCDF Library. Copyright © 1993–1999, University Corporation for Atmospheric Research/Unidata.

HDF EOS Library. Copyright © 1996, Hughes and Applied Research Corporation.

SMACC. Copyright © 2000-2004, Spectral Sciences, Inc. and ITT Visual Information Solutions. All rights reserved.

This software is based in part on the work of the Independent JPEG Group.

Portions of this software are copyrighted by DataDirect Technologies, © 1991–2003.

BandMax®. Copyright © 2003, The Galileo Group Inc.

Portions of this computer program are copyright © 1995–1999, LizardTech, Inc. All rights reserved. MrSID is protected by U.S. Patent No. 5,710,835. Foreign Patents Pending.

Portions of this software were developed using Unisearch's Kakadu software, for which ITT has a commercial license. Kakadu Software. Copyright © 2001. The University of New South Wales, UNSW, Sydney NSW 2052, Australia, and Unisearch Ltd, Australia.

This product includes software developed by the Apache Software Foundation (http://www.apache.org/).

MODTRAN is licensed from the United States of America under U.S. Patent No. 5,315,513 and U.S. Patent No. 5,884,226.

FLAASH is licensed from Spectral Sciences, Inc. under a U.S. Patent Pending.

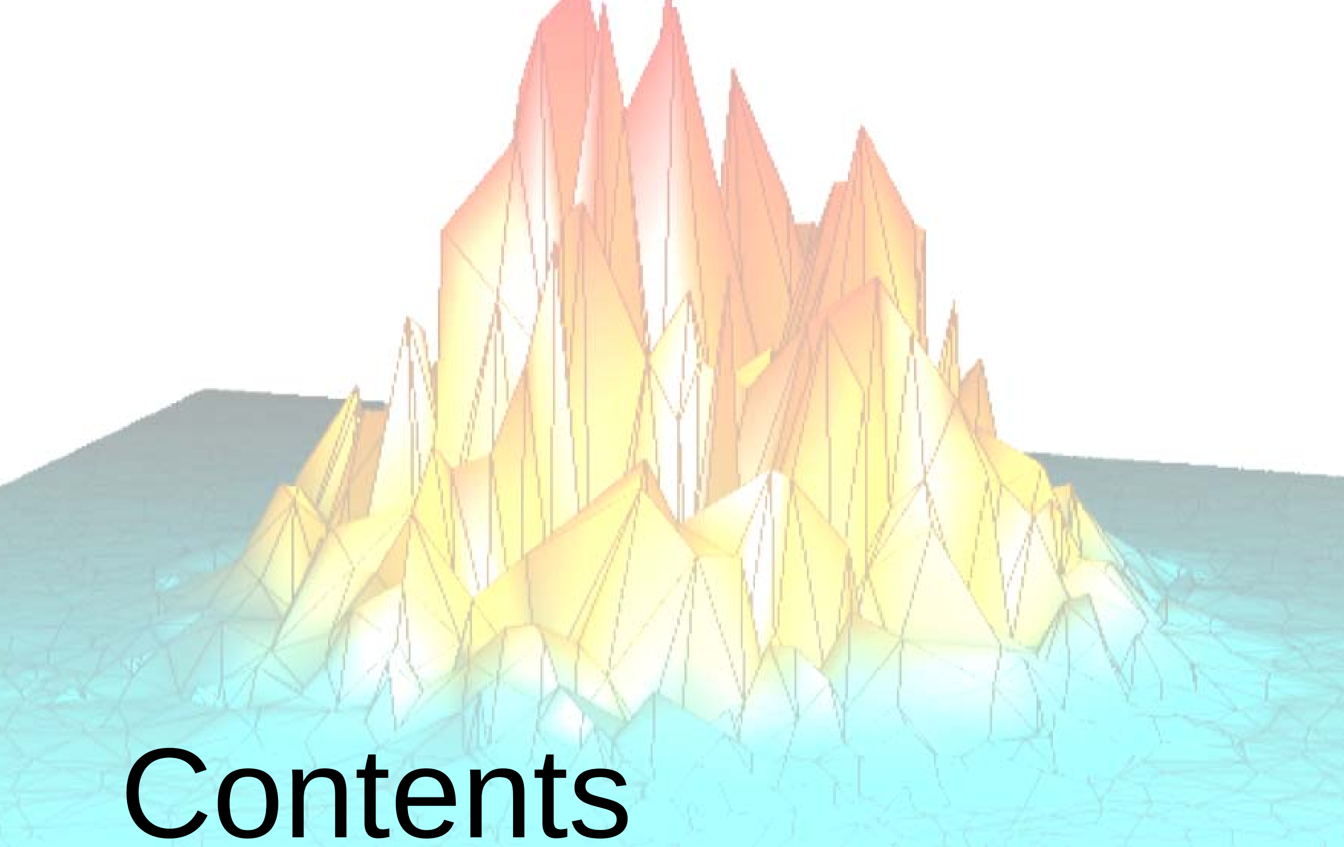
Portions of this software are copyrighted by Merge Technologies Incorporated.

Support Vector Machine (SVM) is based on the LIBSVM library written by Chih-Chung Chang and Chih-Jen Lin (http://www.csie.ntu.edu.tw/~cjlin/libsvm), adapted by ITT Visual Information Solutions for remote sensing image supervised classification purposes.

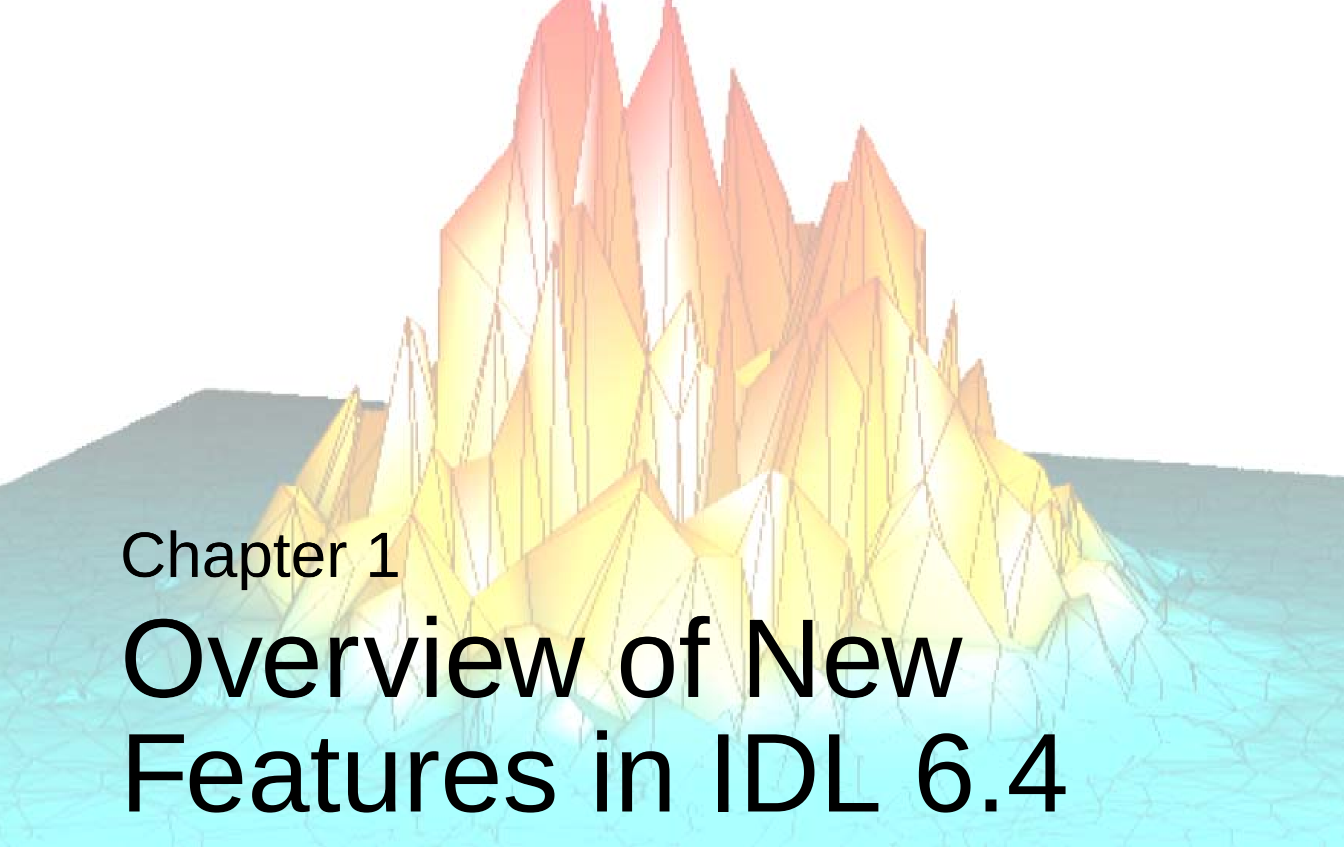
IDL Wavelet Toolkit Copyright © 2002, Christopher Torrence.

IMSL is a trademark of Visual Numerics, Inc. Copyright © 1970–2006 by Visual Numerics, Inc. All Rights Reserved.

Other trademarks and registered trademarks are the property of the respective trademark holders.



Contents



Chapter 1

Overview of New Features in IDL 6.4

This chapter contains the following topics:

Platform Support Changes

The following enhancements have been made to IDL's platform support for the 6.4 release:

Microsoft Windows Vista Support

IDL 6.4 supports Windows Vista. See "Requirements for This Release" on page 41 for additional details.

Macintosh OS X on Intel Processor Support

IDL 6.4 supports Macintosh OS X on Intel processors. (Support was added to IDL 6.3 in a special release several months after the release of IDL 6.3 for other platforms.) See "Requirements for This Release" on page 41 for additional details.

iTool Enhancements

The IDL Intelligent Tools (iTools) are a set of interactive utilities that combine data analysis and visualization with the task of producing presentation quality graphics. Introduced in IDL 6.0, the iTools are designed to help you get the most out of your data with minimal effort. They allow you to benefit from the control of a programming language, while accelerating your data analysis through the use of interactive utilities.

For details on these additions and other enhancements that have been made to the IDL iTools system for the 6.4 release, see the following topic:

- New and Enhanced iTools Keywords
- Image Plane Location Control
- TrueType Font Availability
- Z Values for iMap’s Graphic Layers

New and Enhanced iTools Keywords

The following are new or enhanced keywords to the iTool launch routines:

- ANISTROPIC_SCALE_2D — This keyword indicates the ratio of the Y dimension to the X dimension for two-dimensional images. Applies to: all iTools.
- ANISTROPIC_SCALE_3D — This keyword indicates the ratio of the Z dimension to the X and Y dimensions for three-dimensional images. Applies to: all iTools.
- FIT_TO_VIEW — This keyword automatically scales the newly-created visualization so that it fills the current view. Applies to: all iTools.
- GEOTIFF — This keyword allows GeoTIFF structures to be passed into iMap. Applies to: IMAP.
- INSERT_COLORBAR — This new keyword allows you to display a colorbar automatically in your iTool. Applies to: all iTools.
- INSERT_LEGEND — This keyword automatically inserts a legend into your visualization. Applies to: ICONTOUR, IPLOT, and ISURFACE.
- MAX_VALUE — This keyword indicates the maximum value to be plotted. Applies to: ICONTOUR and ISURFACE.

- MIN_VALUE — This keyword indicates the minimum value to be plotted.Applies to: ICONTOUR and ISURFACE.
- RGB_TABLE — This keyword has been enhanced to accept an IDL colorstyle number. Applies to: all iTools.
- RENDERER — This keyword overrides the value specified by the IDL_GR_WIN_RENDERER (Windows) or IDL_GR_X_RENDERER (UNIX) preference for the iTool. Applies to: all iTools.
- SCALE_ISOTROPIC — This keyword indicates the scaling method to be used for the dataspace. Applies to: all iTools.
- SYM_COLOR — This keyword specifies the color for the missing-point symbols. Applies to: IVECTOR.
- SYM_INDEX — This keyword specifies the symbol to be used for missing points. Applies to: IVECTOR.
- SYM_OBJECT — Provides the ability to define and select user-defined plot symbols. Applies to: IPLOT.
- SYM_SIZE — This keyword specifies the size of the missing-point symbols. Applies to: IVECTOR.
- TRANSPARENCY — This keyword specifies the percent transparency of the visualization. Applies to: ICONTOUR, IIMAGE, IPLOT, ISURFACE, and IVECTOR.
- USE_DEFAULT_COLOR — This keyword specifies the use of the color of the vectors for the missing-point symbols (this is the default). Applies to: IVECTOR.
- VIEW_ZOOM — This keyword sets the initial view zoom factor. Applies to: all iTools.
- [XYZ]LOG — These keywords specifies a logarithmic axis. Applies to: ICONTOUR, ISURFACE, and IVECTOR.

Image Plane Location Control

A new Image Plane Location property now allows you to specify the exact data location of the image plane.

TrueType Font Availability

iTools can now use any TrueType font available on your system. For more information, see [XYZ]TICKFONT_INDEX under any of the iTools Routines.

Z Values for iMap’s Graphic Layers

Map Gridlines, Shapefile Polygons, Polylines, and Points visualizations now have a new Z value. If multiple images are displayed in a stack you can now specify the corresponding Z value for the graphics so that any or all of the images can have graphics drawn on them.

Visualization Enhancements

The following enhancements have been made to IDL's visualization functionality for the 6.4 release:

- 24-bit Z-Buffer Device
- Support for OpenGL Shading Language (GLSL)

24-bit Z-Buffer Device

IDL's Z-Buffer device has been enhanced to support a 24-bit (TrueColor) frame buffer. By default, the Z-Buffer uses an 8-bit frame buffer; the new SET_PIXEL_DEPTH keyword to the DEVICE procedure allows you to specify the pixel depth of the buffer. The new GET_PIXEL_DEPTH keyword allows you to retrieve the current value from the Z-Buffer device.

See "The Z-Buffer Device" (Appendix A, *IDL Reference Guide*) for complete details.

Support for OpenGL Shading Language (GLSL)

The *shader* functionality implemented in IDL object graphics provides access to the advantages of the hardware-based OpenGL Shading Language (GLSL) features that are available on modern graphics cards. Using a shader program, computationally intensive image processing operations can be off-loaded to a graphics card that supports the OpenGL 2.0 interface. Additionally, the OpenGL Shading Language greatly expands on the capabilities of the fixed OpenGL rendering pipeline to produce advanced visual effects. A shader program can perform image filtering operations, complex lighting and shading calculations, interactive blending of multiple textures, animation of object vertices, or any combination of these.

The IDLgrShader object provides a way to associate a shader program with atomic graphic objects including IDLgrImage, IDLgrPolygon, and IDLgrSurface. In addition to the shader object, the IDL distribution also includes a small library of pre-built shader objects. The IDLgrShaderBytscl and IDLgrShaderConvol3 objects let you quickly add byte-scaling or convolution filtering shader functionality to an IDL application without having to write any shader code. Also, the new IDLgrFilterChain object lets you apply a series of image filtering shaders to an image. See "New IDL Object Classes" on page 29 for more information. Also refer to Chapter 14, "Advanced Rendering Using Shader Objects" (*Object Programming*) for information about and examples of using shaders in IDL applications.

Analysis Enhancements

The following enhancements have been made to IDL's data-analysis functionality for the 6.4 release:

- Color Mapping Functionality
- Edge Detection Filters
- Noise Functions
- Statistics Enhancements to COMFIT

Color Mapping Functionality

Five new color mapping functions have been added to IDL 6.4. These are:

- COLOR_EXCHANGE
- COLOR_RANGE_MAP
- COLORIZE_SAMPLE
- COLORMAP_GRADIENT
- COLORMAP_ROTATION

Additionally, COLOR_CONVERT has been enhanced to include more color spaces.

For more information, see "New IDL Routines" on page 20 and "IDL Routine Enhancements" on page 22.

Edge Detection Filters

Five new filters in IDL enhance edge detection capabilities. These are:

- EDGE_DOG
- EMBOSS
- LAPLACIAN
- PREWITT
- SHIFT_DIFF

For more information, see "New IDL Routines" on page 20.

Noise Functions

Four new noise functions have been added to IDL. These are:

- NOISE_HURL
- NOISE_PICK
- NOISE_SCATTER
- NOISE_SLUR

For more information, see “New IDL Routines” on page 20.

Statistics Enhancements to COMFIT

This release contains new keywords to COMFIT allowing for enhanced output statistics. For more information on these new keywords, see “IDL Routine Enhancements” on page 22.

Language Enhancements

The following enhancements have been made to the core language for the 6.4 release:

- OBJ_HASMETHOD Routine
- Export Bridge Object Licensing
- Mouse Wheel Events in Exported IDL Objects
- New IDLnetURL Class

OBJ_HASMETHOD Routine

The OBJ_HASMETHOD function determines whether an object class implements or inherits a specified method or methods. The function returns true for a given class and method combination if the class or any of its superclasses defines the method.

Export Bridge Object Licensing

In IDL 6.3, IDL objects exported via the Java or COM Export Bridge required a full IDL development license to run. In IDL 6.4 and later, developers of exported IDL objects can control the licensing requirements for their objects, allowing Java and COM programs that rely on them to run in IDL runtime mode or IDL Virtual Machine mode as well as in full development mode.

See “IDL Licensing Modes” (*IDL Connectivity Bridges*) for additional details.

Mouse Wheel Events in Exported IDL Objects

Exported IDL object classes wrapped by Java or COM objects can now respond to events generated by the rolling of a mouse’s scroll wheel. See “Event Handling” (Chapter 8, *IDL Connectivity Bridges*) for details on using wheel events in COM objects. See “Event Handling” (Chapter 9, *IDL Connectivity Bridges*) for details on using wheel events in Java objects.

New IDLnetURL Class

A new IDLnetURL class has been added that allows IDL to act as a client to an HTTP or FTP server. The object handles:

- Opening a connection to a remote HTTP or FTP server.
- Creating and sending http requests to the remote HTTP or FTP server.

- Receiving responses from the remote HTTP or FTP server.
- Receiving and writing large data files to disk at a user-specified location.

File Access Enhancements

The following enhancements have been made to IDL’s file-access capabilities in the IDL 6.4 release:

- Transparency Support for GIF Files
- Local Colormap Support for GIF Files
- Open Geospatial Consortium Server Access
- XML Read and Write Enhancements

Transparency Support for GIF Files

IDL now supports transparency for GIF files through new keywords to both READ_GIF and WRITE_GIF. For more information, see “IDL Routine Enhancements” on page 22.

Local Colormap Support for GIF Files

IDL now supports local colormaps for GIF files that contain multiple images.

Open Geospatial Consortium Server Access

The Open Geospatial Consortium (OGC) is an open-standards organization devoted to developing standards that govern web-based delivery of geospatial data including geographical information systems (GIS) data combined with location, elevation or other types of structure data. This release of IDL features two new network objects, IDLnetOGCWMS and IDLnetOGCWCS, that let you access Web Map Service servers and Web Coverage Service servers respectively. For more information, see “New IDL Object Classes” on page 29.

XML Read and Write Enhancements

The Init, Load, and Save methods to the IDLffXMLDOMDocument object now support reading and writing XML data from and to IDL string variables, avoiding the need for file input/output.

Similarly, the ParseFile method to the IDLffXMLSAX object allows XML data to be read from a string variable or a remote URL.

For more information, see “IDL Object Method Enhancements” on page 35.

User Interface Toolkit Enhancements

The following enhancements have been made to the IDL's graphical user interface toolkit in the IDL 6.4 release:

- Title Bar Icons for Base Widgets
- Draw Widget Wheel Events on Motif Platforms

Title Bar Icons for Base Widgets

IDL now supports (for Windows platforms only) the ability to add title bar icons to base widgets. The new BITMAP and MASK keywords to WIDGET_BASE allow this functionality. For more information, see "IDL Routine Enhancements" on page 22.

Draw Widget Wheel Events on Motif Platforms

IDL on Motif platforms now supports the WHEEL_EVENTS keyword to WIDGET_DRAW. Wheel events have been supported on Windows platforms since IDL version 6.2.

Documentation Enhancements

In addition to documentation for new and enhanced IDL features, the following enhancements to the IDL documentation set are included in the 6.4 release:

- IDL Analyst Manual
- Manual Reorganization

IDL Analyst Manual

IDL Analyst combines the power of IDL with the IMSL C Numerical Library provided by Visual Numerics, Inc. The addition of the IMSL library gives IDL users access to an extensive and powerful set of mathematical and statistical analysis routines via the standard IDL programmer's interface.

IDL Analyst was first released after the IDL 6.3 release; as a result, users who purchased that release received only the Adobe Acrobat PDF version of the new *IDL Analyst Reference Guide*. Although IDL Analyst is a separately licensed module, documentation is now included as part of the IDL help system.

The *IDL Analyst Reference Guide* appears in the *Guides for Add-on Products* section of the IDL help system, and in the Adobe Acrobat PDF documentation set.

Manual Reorganization

In an effort to better delineate topics in the IDL help system, we have made the following changes to the IDL documentation set:

- The manual titled *Building IDL Applications* has been renamed *Application Programming*.
- The manual titled *iTool Developer's Guide* has been renamed *iTool Programming*.
- Material on user interface programming using IDL widgets that previously appeared in Building IDL Applications and elsewhere is now contained in *User Interface Programming*.

All three manuals are included in the *Programmer's Guides* section of the IDL help system, and in the Adobe Acrobat PDF documentation set.

New IDL Routines

The following new functions and procedures were added to IDL in this release. See the following topics in the *IDL Reference Guide* for complete reference information unless otherwise noted.

CDF_EPOCH_COMPARE — The CDF_EPOCH_COMPARE function compares two epoch (date and time) values and returns an integer value of 1, 0, or -1.

CDF_EPOCH_DIFF — The CDF_EPOCH_DIFF function compares two epoch (date and time) values and returns the difference in milliseconds (default) or microseconds.

COLOR_EXCHANGE — The COLOR_EXCHANGE procedure replaces image pixels of a given color with pixels of a new color.

COLOR_RANGE_MAP — The COLOR_RANGE_MAP function maps all the pixels of an image to another set of pixels, using source and target ranges to control the mapping.

COLORIZE_SAMPLE — The COLORIZE_SAMPLE function colorizes a grayscale image by matching luminance levels with an RGB sample table.

COLORMAP_GRADIENT — The COLORMAP_GRADIENT function maps an image into a specified luminance-based gradient.

COLORMAP_ROTATION — The COLORMAP_ROTATION function maps pixels within a given hue range to another hue range, using the HSV hue component.

EDGE_DOG — The EDGE_DOG function applies the Difference of Gaussians filter to a 2D image array.

EMBOSS — Applies an “embossed” convolution operator to a 2D image array.

LAPLACIAN — Applies a Laplacian operator to a 2D image array.

NOISE_HURL — The NOISE_HURL function introduces noise into an image by changing randomly selected pixels to random colors.

NOISE_PICK — The NOISE_PICK function introduces noise into an image by picking randomly selected pixels to be replaced by a neighboring pixel from a random direction.

NOISE_SCATTER — The NOISE_SCATTER function introduces noise into an image by applying normally distributed noise to the entire image.

NOISE_SLUR — The NOISE_SLUR function introduces noise into an image by selecting random pixels to be replaced by a neighboring pixel from a random location in the row above.

OBJ_HASMETHOD — Determines whether an object class implements or inherits a specified method or methods.

PREWITT — The PREWITT function returns an approximation to the Prewitt edge enhancement operator for images.

SHIFT_DIFF — The SHIFT_DIFF function applies a shift-difference filter to a 2D image array.

IDL Routine Enhancements

The following IDL routines have updated keywords, arguments, or return values in this release. See the following topics in the *IDL Reference Guide* for complete reference information unless otherwise noted.

COLOR_CONVERT — The COLOR_CONVERT function can now convert multi-channel image arrays and has the following new keywords:

- YUV_RGB is set to convert from YUV to RGB.
- YIQ_RGB is set to convert from YIQ to RGB.
- YPBPR_RGB is set to convert from YPbPr to RGB.
- YCBCR_RGB is set to convert from YCbCr to RGB.
- RGB_YUV is set to convert from RGB to YUV.
- RGB_YIQ is set to convert from RGB to YIQ.
- RGB_YPBPR is set to convert from RGB to YPbRp.
- RGB_YCBCR is set to convert from RGB to YCbCr.
- INTERLEAVE is set to the value that corresponds to the image array's interleave format.

COMFIT — The COMFIT function has the following new keywords:

- CHISQ returns the value of the reduced chi-square goodness-of-fit statistic.
- DOUBLE is set to force the computation to be done in double-precision arithmetic.
- ITER returns the number of iterations performed.
- ITMAX is set to specify the maximum number of iterations.
- STATUS sets the status of the computation.
- TOL sets the desired convergence tolerance.
- YERROR returns the standard error between YFIT and Y.

CW_ANIMATE — The CW_ANIMATE function has the following new keywords:

- MPEG_BITRATE specifies the MPEG movie bit rate.
- MPEG_FORMAT specified the MPEG encoding type.

- MPEG_QUALITY specifies the quality at which the MPEG stream is to be stored.

DEVICE — The DEVICE procedure has the following new keywords and enhancements related to the 24-bit Z-Buffer Device:

- DECOMPOSED and GET_DECOMPOSED keywords now apply to the Z-buffer device.
- GET_PIXEL_DEPTH returns the pixel depth of the Z-buffer device.
- SET_PIXEL_DEPTH controls the pixel depth of the Z-buffer device. Allowed values are 8 and 24.
- GET_WRITE_MASK and SET_WRITE_MASK now apply to the Z-buffer device.

ICONTOUR — The ICONTOUR function has the following new keywords:

- ANISTROPIC_SCALE_2D indicates the ratio of the Y dimension to the X dimension for two-dimensional images.
- ANISTROPIC_SCALE_3D indicates the ratio of the Z dimension to the X and Y dimensions for three-dimensional images.
- FIT_TO_VIEW automatically scales the newly-created visualization so that it fills the current view.
- INSERT_COLORBAR inserts a colorbar.
- INSERT_LEGEND inserts a legend.
- MAX_VALUE indicates the maximum value to be plotted.
- MIN_VALUE indicates the minimum value to be plotted.
- RENDERER overrides the value specified by the IDL_GR_WIN_RENDERER (Windows) or IDL_GR_X_RENDERER (UNIX) preference for the iTool.
- SCALE_ISOTROPIC indicates the scaling method to be used for the dataspace.
- TRANSPARENCY specifies the percent transparency of the visualization.
- VIEW_ZOOM sets the initial view zoom factor.
- [XYZ]LOG specifies a logarithmic axis.

IDLITSYS_CREATETOOL — The IDLITSYS_CREATETOOL function has the following new keywords:

- ANISTROPIC_SCALE_2D indicates the ratio of the Y dimension to the X dimension for two-dimensional images.
- ANISTROPIC_SCALE_3D indicates the ratio of the Z dimension to the X and Y dimensions for three-dimensional images.
- RENDERER overrides the value specified by the IDL_GR_WIN_RENDERER (Windows) or IDL_GR_X_RENDERER (UNIX) preference for the iTool.
- SCALE_ISOTROPIC indicates the scaling method to be used for the dataspace.

IIMAGE — The IIMAGE function has the following new keywords:

- ANISTROPIC_SCALE_2D indicates the ratio of the Y dimension to the X dimension for two-dimensional images.
- ANISTROPIC_SCALE_3D indicates the ratio of the Z dimension to the X and Y dimensions for three-dimensional images.
- FIT_TO_VIEW automatically scales the newly-created visualization so that it fills the current view.
- INSERT_COLORBAR inserts a colorbar.
- RENDERER overrides the value specified by the IDL_GR_WIN_RENDERER (Windows) or IDL_GR_X_RENDERER (UNIX) preference for the iTool.
- SCALE_ISOTROPIC indicates the scaling method to be used for the dataspace.
- TRANSPARENCY specifies the percent transparency of the visualization.
- VIEW_ZOOM sets the initial view zoom factor.

IMAP — The IMAP function has the following new keywords:

- ANISTROPIC_SCALE_2D indicates the ratio of the Y dimension to the X dimension for two-dimensional images.
- ANISTROPIC_SCALE_3D indicates the ratio of the Z dimension to the X and Y dimensions for three-dimensional images.
- FIT_TO_VIEW automatically scales the newly-created visualization so that it fills the current view.
- GEOTIFF specifies a structure containing the GeoTIFF tags from a GeoTIFF file.

- RENDERER overrides the value specified by the IDL_GR_WIN_RENDERER (Windows) or IDL_GR_X_RENDERER (UNIX) preference for the iTool.
- SCALE_ISOTROPIC indicates the scaling method to be used for the dataspace.
- VIEW_ZOOM sets the initial view zoom factor.

IPLOT — The IPLOT function has the following new keywords:

- ANISTROPIC_SCALE_2D indicates the ratio of the Y dimension to the X dimension for two-dimensional images.
- ANISTROPIC_SCALE_3D indicates the ratio of the Z dimension to the X and Y dimensions for three-dimensional images.
- FIT_TO_VIEW automatically scales the newly-created visualization so that it fills the current view.
- INSERT_COLORBAR inserts a colorbar.
- INSERT_LEGEND inserts a legend.
- RENDERER overrides the value specified by the IDL_GR_WIN_RENDERER (Windows) or IDL_GR_X_RENDERER (UNIX) preference for the iTool.
- SCALE_ISOTROPIC indicates the scaling method to be used for the dataspace.
- SYM_OBJECT specifies an object reference to be used for the plotting symbol.
- TRANSPARENCY specifies the percent transparency of the visualization.
- VIEW_ZOOM sets the initial view zoom factor.

ISURFACE — The ISURFACE function has the following new keywords:

- ANISTROPIC_SCALE_2D indicates the ratio of the Y dimension to the X dimension for two-dimensional images.
- ANISTROPIC_SCALE_3D indicates the ratio of the Z dimension to the X and Y dimensions for three-dimensional images.
- FIT_TO_VIEW automatically scales the newly-created visualization so that it fills the current view.
- INSERT_COLORBAR inserts a colorbar.
- INSERT_LEGEND inserts a legend.

- MAX_VALUE indicates the maximum value to be plotted.
- MIN_VALUE indicates the minimum value to be plotted.
- RENDERER overrides the value specified by the IDL_GR_WIN_RENDERER (Windows) or IDL_GR_X_RENDERER (UNIX) preference for the iTool.
- SCALE_ISOTROPIC indicates the scaling method to be used for the dataspace.
- TRANSPARENCY specifies the percent transparency of the visualization.
- VIEW_ZOOM sets the initial view zoom factor.
- [XYZ]LOG specifies a logarithmic axis.

IVECTOR — The IVECTOR function has the following new keywords:

- ANISTROPIC_SCALE_2D indicates the ratio of the Y dimension to the X dimension for two-dimensional images.
- ANISTROPIC_SCALE_3D indicates the ratio of the Z dimension to the X and Y dimensions for three-dimensional images.
- FIT_TO_VIEW automatically scales the newly-created visualization so that it fills the current view.
- INSERT_COLORBAR inserts a colorbar.
- RENDERER overrides the value specified by the IDL_GR_WIN_RENDERER (Windows) or IDL_GR_X_RENDERER (UNIX) preference for the iTool.
- SCALE_ISOTROPIC indicates the scaling method to be used for the dataspace.
- SYM_COLOR specifies the color for the missing-point symbols.
- SYM_INDEX specifies the symbol to be used for missing points.
- SYM_SIZE specifies the size of the missing-point symbols.
- TRANSPARENCY specifies the percent transparency of the visualization.
- USE_DEFAULT_COLOR specifies the use of the color of the vectors for the missing-point symbols (this is the default).
- VIEW_ZOOM sets the initial view zoom factor.
- [XYZ]LOG specifies a logarithmic axis.

IVOLUME — The IVOLUME function has the following new keywords:

- ANISTROPIC_SCALE_2D indicates the ratio of the Y dimension to the X dimension for two-dimensional images.
- ANISTROPIC_SCALE_3D indicates the ratio of the Z dimension to the X and Y dimensions for three-dimensional images.
- FIT_TO_VIEW automatically scales the newly-created visualization so that it fills the current view.
- INSERT_COLORBAR inserts a colorbar.
- RENDERER overrides the value specified by the IDL_GR_WIN_RENDERER (Windows) or IDL_GR_X_RENDERER (UNIX) preference for the iTool.
- SCALE_ISOTROPIC indicates the scaling method to be used for the dataspace.
- VIEW_ZOOM sets the initial view zoom factor.

LOADCT — The LOADCT function has the following new keyword.

- RGB_TABLE returns the desired color table as an [NCOLORS, *3*] array.

READ_GIF — The READ_GIF function has the following new keywords:

- BACKGROUND_COLOR returns the index of the background color within the global color table.
- DELAY_TIME returns the delay in hundredths (1/100) of a second that the decoder should wait after displaying the current image.
- DISPOSAL_METHOD returns the disposal method that the decoder should use after displaying the current image.
- REPEAT_COUNT returns the repeat count for the animation within the file.
- TRANSPARENT sets the byte value giving the index within the color table to be designated as the transparent color.
- USER_INPUT returns the user input flag for the current image.

WIDGET_BASE — The WIDGET_BASE function has the following new keywords:

- BITMAP specifies a custom top level base icon that will be used in the title bar, the task bar and the Windows task switching window.
- MASK creates (along with the BITMAP keyword) a top level base icon that has transparency.

WRITE_GIF — The WRITE_GIF function has the following new keywords:

- BACKGROUND_COLOR sets a byte value giving the index within the global color table to be designated as the background.
- DELAY_TIME sets an integer giving the delay in hundredths (1/100) of a second after the decoder displays the current image (for use with multiple images).
- DISPOSAL_METHOD sets an integer giving the method that the decoder should use for disposing the current image after display (for use with multiple images).
- REPEAT_COUNT sets an integer giving the number of times that the decoder should repeat the animation (for use with multiple images).
- TRANSPARENT sets a byte value giving the index within the color table to be designated as the transparent color.
- USER_INPUT sets a flag indicating whether the decoder should require user input before continuing processing.

XINTERANIMATE — The XINTERANIMATE function has the following new keyword:

- MPEG_FORMAT specifies the type of MPEG encoding to use.

New IDL Object Classes

The following new object classes were added to IDL in this release. See the following topics in the *IDL Reference Guide* for complete reference information unless otherwise noted.

IDLgrFilterChain — This object lets you apply a series of image filtering shaders to an image object. You can add IDLgrShader objects or objects subclassing from IDLgrShader to this specialized filter chain container. When the sequence of shaders are executed, the output from the first shader is passed to each subsequent shader in the chain until the last shader is reached at which point the result is drawn to the destination device.

IDLgrShader — This object exposes the hardware-based OpenGL Shader Language (GLSL) code within an IDL object graphics application, taking advantage of the flexibility and power offered by completing computationally intensive tasks on a suitable graphics card processing unit instead of the CPU.

IDLgrShaderBytscl — This object is a subclass of IDLgrShader that can be associated with an IDLgrImage, letting you quickly perform a BYTSCL operation on the image data as it is drawn to the screen. This object takes advantage of the advanced processing power of the hardware-based OpenGL Shading Language (GLSL) if an appropriate graphics card is available, or provides a software-based alternative if suitable hardware is not found.

IDLgrShaderConvol3 — This convolution shader object is a subclass of IDLgrShader that can be associated with an IDLgrImage object via the SHADER property to perform general 3x3 convolution operations such as smoothing and edge detection on the image data as it is drawn to the screen. If hardware shader support is available, IDL will use it. Otherwise, this object performs the operation using a software-based alternative.

IDLnetOGCWCS — This object lets you access a Web Coverage Service (WCS) server that supports the Open Geospatial Consortium (OGC) standards. Using this object, you can return unrendered geospatial data containing sensor data, hyperspectral data, elevation data, or imagery from a WCS server. Given a valid URL to an OGS WCS server, this object handles the details of establishing a connection to the server, making requests, receiving responses (and errors, when applicable), and parsing the responses into structures that can be accessed in IDL.

IDLnetOGCWMS — This object lets you access a Web Map Service (WMS) server that supports the Open Geospatial Consortium (OGC) standards. Using this object, you can return one or more layers of data as a rendered map image. In addition, you can request information on map features if supported by the server and if available.

IDLnetURL — This object allows IDL to act as a client to an HTTP or FTP server. The object handles:

- Opening a connection to a remote HTTP or FTP server.
- Creating and sending HTTP requests to the remote HTTP or FTP server.
- Receiving responses from the remote HTTP or FTP server.
- Receiving and writing large data files to disk at a user-specified location.

New IDL Object Properties

The following IDL object classes have new properties in this release. See the following topics in the *IDL Reference Guide* for complete reference information.

IDLgrAxis — This object includes the following new property:

- SHADER associates an IDLgrShader object (or object subclassing from IDLgrShader) with this object.

IDLgrContour — This object includes the following new property:

- SHADER associates an IDLgrShader object (or object subclassing from IDLgrShader) with this object.

IDLgrImage — This object includes the following new properties:

- IMAGE_1D lets you indicate the image data is one-dimensional (useful for creating lookup table texture maps for use in fragment shaders).
- INTERNAL_DATA_TYPE lets you describe the data format to use when storing image in the graphics hardware as a texture map.
- SHADER associates an IDLgrShader object, an object subclassing from IDLgrShader, or an IDLgrFilterChain object with this object.
- TILE_BORDER_SIZE defines the size of a tile border, which is useful when a convolution filter is applied using a shader program. Set this property to a value that will give the convolution filter access to all of the pixels required (including those belonging to adjacent tiles) for filtering.

IDLgrLight — This object includes the following new properties:

- LIGHT_INDEX distinguishes between up to eight individual, non-ambient light objects when your shader program performs lighting calculations.
- SHADER associates an IDLgrShader object (or object subclassing from IDLgrShader) with this object.

IDLgrPlot — This object includes the following new property:

- SHADER lets you associate an IDLgrShader object (and its shader program) with this object.

IDLgrPolygon — This object includes the following new property:

- SHADER associates an IDLgrShader object (or object subclassing from IDLgrShader) with this object.

IDLgrPolyline — This object includes the following new property:

- SHADER associates an IDLgrShader object (or object subclassing from IDLgrShader) with this object.

IDLgrROI — This object includes the following new property:

- SHADER associates an IDLgrShader object (or object subclassing from IDLgrShader) with this object.

IDLgrROIGroup — This object includes the following new property:

- SHADER associates an IDLgrShader object (or object subclassing from IDLgrShader) with this object.

IDLgrText — This object includes the following new property:

- SHADER associates an IDLgrShader object (or object subclassing from IDLgrShader) with this object.

IDLgrVolume — This object includes the following new property:

- SHADER associates an IDLgrShader object (or object subclassing from IDLgrShader) with this object.

New IDL Object Methods

The following IDL object classes have new methods in this release. See the following topics in the *IDL Reference Guide* for complete reference information.

IDLgrImage::ReadFilteredData — This method returns the image data after the application of one or more image filtering IDLgrShader programs to the image.

IDLgrImage::TileDataLoaded — This method returns a boolean value indicating whether the specified tile already has data loaded.

IDLgrPlot::GetVertexAttributeData — This method retrieves the data associated with a named attribute variable containing per-vertex attribute data.

IDLgrPlot::SetVertexAttributeData — This method passes a set of per-vertex attribute data to a vertex shader program, which modifies each individual vertex with a vertex's worth of attribute data at a time.

IDLgrPolygon::GetMultiTextureCoord — This method retrieves the texture coordinates associated with a given Unit.

IDLgrPolygon::GetVertexAttributeData — This method retrieves the data associated with a named attribute variable containing per-vertex attribute data.

IDLgrPolygon::SetMultiTextureCoord — This method associates a Unit with a given set of texture coordinates. This lets you layer textures with different texture coordinates onto a polygon when using a shader program.

IDLgrPolygon::SetVertexAttributeData — This method passes a set of per-vertex attribute data to a vertex shader program, which modifies each individual vertex with a vertex's worth of attribute data at a time.

IDLgrPolyline::GetVertexAttributeData — This method retrieves the data associated with a named attribute variable containing per-vertex attribute data.

IDLgrPolyline::SetVertexAttributeData — This method passes a set of per-vertex attribute data to a vertex shader program, which modifies each individual vertex with a vertex's worth of attribute data at a time.

IDLgrROIGroup::Remove — This method removes an object from the ROI group.

IDLgrSurface::GetMultiTextureCoord — This method retrieves the texture coordinates associated with a given Unit.

IDLgrSurface::GetVertexAttributeData — This method retrieves the data associated with a named attribute variable containing per-vertex attribute data.

IDLgrSurface::SetMultiTextureCoord — This method associates a Unit with a given set of texture coordinates. This lets you layer textures with different texture coordinates onto a surface when using a shader program.

IDLgrSurface::SetVertexAttributeData — This method passes a set of per-vertex attribute data to a vertex shader program, which modifies each individual vertex with a vertex's worth of attribute data at a time.

IDLgrWindow::OnWheel — This method handles notification (from the native window device) that a mouse wheel event has occurred. (Useful only with the IDL Export Bridge.)

IDLitDirectWindow::OnWheel — This method handles notification (from the native window device) that a mouse wheel event has occurred. (Useful only with the IDL Export Bridge.)

IDLitManipulator::OnWheel — This method handles notification (from the native window device) that a mouse wheel event has occurred. (Useful only with the IDL Export Bridge.)

IDLitManipulatorContainer::OnWheel — This method handles notification (from the native window device) that a mouse wheel event has occurred. (Useful only with the IDL Export Bridge.)

IDLitWindow::OnWheel — This method handles notification (from the native window device) that a mouse wheel event has occurred. (Useful only with the IDL Export Bridge.)

IDL Object Method Enhancements

The following IDL object classes have enhanced methods in this release. See the following topics in the *IDL Reference Guide* for complete reference information.

IDLffXMLDOMDocument::Init — This method features the following new keywords:

- STRING specifies a scalar string containing the XML Document text.

IDLffXMLDOMDocument::Load — This method features the following new keywords:

- STRING specifies a scalar string containing the XML Document text.

IDLffXMLDOMDocument::Save — This method features the following new keywords:

- STRING specifies a named variable that will contain the XML string data representing the XML DOM document currently represented in this object.

IDLffXMLSAX::ParseFile — This method features the following new keywords:

- URL specifies that the input value is a Uniform Resource Locator that provides the XML data to be parsed.
- XML_STRING specifies that the input value is a string array containing the XML data to be parsed.

IDLgrClipboard::Draw — This method features the following new keywords:

- ISOLATIN1 indicates that the clipboard object should use Adobe ISO Latin 1 font encoding with any font that supports such coding. Use of this keyword allows access to many commonly-used foreign characters. This keyword is ignored if the generated output is not PostScript.

IDLgrWindow::GetDeviceInfo — This method features the following new keywords:

- FRAME_OBJECT_EXTENSION indicates whether the graphics card and OpenGL driver support the GLSL framebuffer_object_extension, which enables filter shader chaining.
- MAX_FRAGMENT_UNIFORM_CONSTANT contains the maximum number of components, or storage units, declared by active uniform variables in a fragment shader program.
- MAX_TEXTURE_IMAGE_UNITS contains the number of texture units available to the fragment shader program.

- MAX_VERTEX_ATTRIBUTES contains the maximum number of positions available for GLSL vertex attribute variables associated with a shader program.
- MAX_VERTEX_TEXTURE_IMAGE_UNITS contains the number of texture units available to the vertex shader program.
- MAX_VERTEX_UNIFORM_CONSTANTS contains the maximum number of components, or storage units, declared by active uniform variables in a vertex shader program.
- SHADING_LANGUAGE_VERSION contains a string indicating the supported OpenGL shading language (GLSL) version number supported by the system's graphic card.

IDLitComponent::GetProperty — This method features the following new keyword:

- When retrieving the UVALUE property, the NO_COPY keyword specifies that the data contained in the property should be transferred to the specified variable rather than being copied, leaving the UVALUE property itself undefined. The NO_COPY keyword has no effect when retrieving other properties.

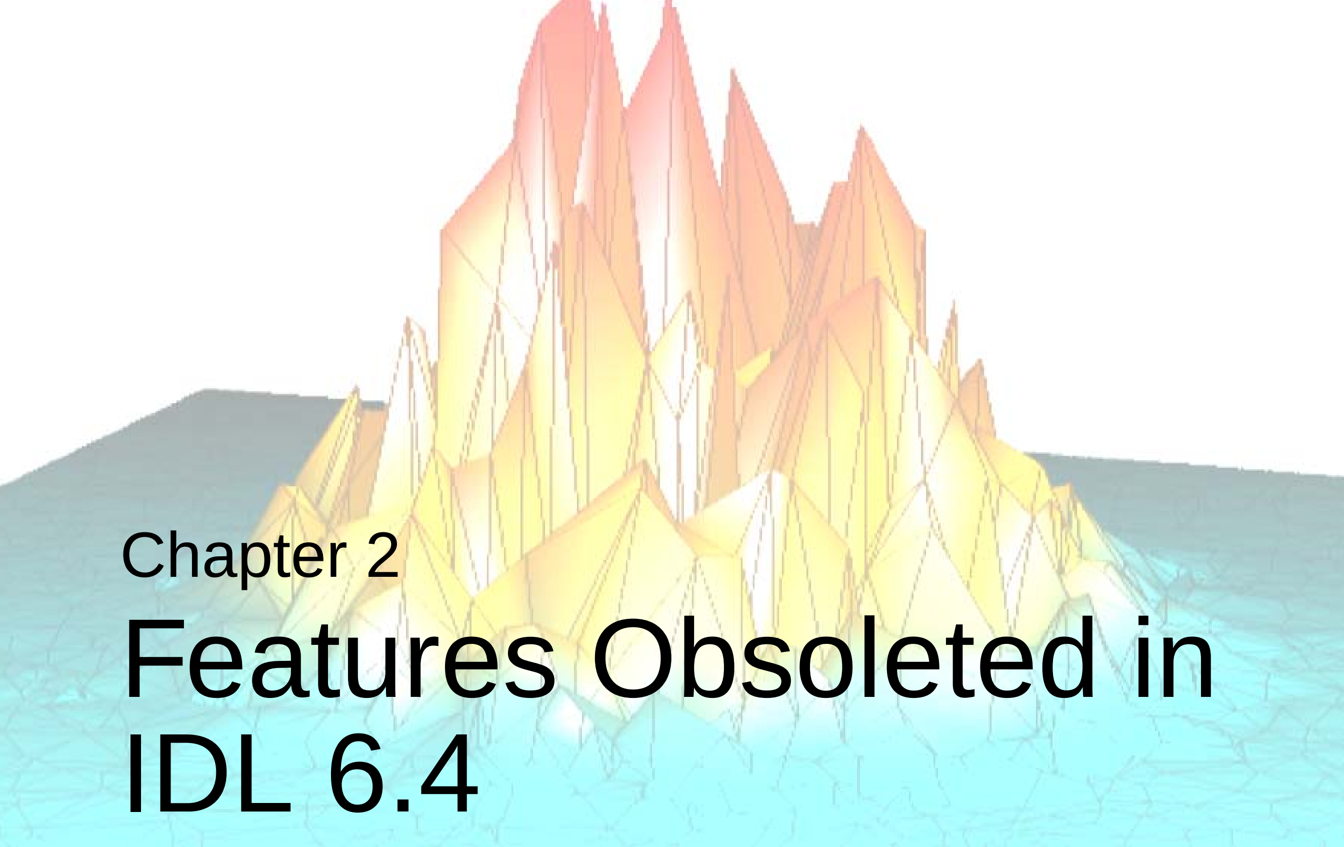
IDLitComponent::SetProperty — This method features the following new keyword:

- When setting the UVALUE property, the NO_COPY keyword specifies that the data contained in the specified variable should be transferred to the property rather than being copied, leaving the original variable undefined. The NO_COPY keyword has no effect when setting other properties.

IDLitWindow::AddWindowEventObserver — This method now supports the observation of events generated by the mouse scroll wheel (wheel events).

IDLitWindow::GetEventMask — This method now supports the observation of events generated by the mouse scroll wheel (wheel events).

IDLitWindow::SetEventMask — This method now supports the observation of events generated by the mouse scroll wheel (wheel events).



Chapter 2

Features Obsoleted in IDL 6.4

The following features were present in IDL Version 6.3 but became obsolete in Version 6.4. Obsoleted features should not be used in new IDL code.

Obsolete Features

The IDLDrawWidget ActiveX control is obsolete. New applications that need to create a drawable COM canvas should use the COM Export Bridge technology described in *IDL Connectivity Bridges*.

The `idldrawx3.ocx` file is still included in the IDL distribution, so applications that use the obsolete control will continue to function. All example code has been removed from the distribution, and documentation has been moved into the *Obsolete Features* manual.

Obsolete Arguments or Keywords

The arguments or keywords to the following routines have been removed:

Routine	Argument or Keyword
IVECTOR	MARK_POINTS keyword

Table 2-1: Obsolete Keywords

Chapter 3

Requirements for This Release

This section describes supported platforms and software requirements.

IDL and ENVI

Hardware and Operating System Requirements

A network interface card (NIC or Ethernet) is required for software-based node-locked and floating licenses.

The following table describes the supported platforms and operating systems for IDL and ENVI. ENVI Zoom is not supported on AIX and IRIX platforms.

Platform	Vendor	Hardware	Operating System	Supported Versions
Windows	Microsoft	Intel/AMD x86 32-bit	Windows	2000 [a], XP, Vista
		Intel/AMD x86_64 64-bit	Windows	XP, Vista
Macintosh [b]	Apple	PowerMac G4, G5 32-bit	OS X	10.4
	Apple	Intel Core Duo 32-bit	OS X	10.4 [c]
UNIX [b]	IBM	RS/6000 32-bit	AIX	5.1
	IBM	RS/6000 64-bit	AIX	5.1
	SGI	Mips 32-bit	IRIX	6.5.1
	SGI	Mips 64-bit	IRIX	6.5.1
	SUN	SPARC 32-bit	Solaris	9, 10
	SUN	SPARC 64-bit	Solaris	9, 10
	various	Intel/AMD x86 32-bit	Linux [d]	Kernel version 2.4 Kernel version 2.6 glibc version 2.3
	various	Intel/AMD x86_64 64-bit	Linux [d]	Kernel version 2.4 Kernel version 2.6 glibc version 2.3

Table 3-1: Hardware Requirements for IDL and ENVI

[a] Windows 2000 Service Pack 4 is required.

[b] For UNIX and Mac OS X, the supported versions indicate that IDL and ENVI were either built on (the lowest version listed) or tested on that version. You can install and run IDL and ENVI on other versions that are binary compatible with those listed.

[c] For Mac OS X on Intel-based computers, version 10.4.8 or later and the X11 Update version 1.1.2 (November 2006) are required.

[d] If your version of Linux is compatible with the listed kernel and glibc versions, you should be able to install and run IDL and ENVI.

32-bit and 64-bit Versions

On UNIX platforms that provide 64-bit support, you can run IDL or ENVI as either a 32-bit or a 64-bit application. When both versions are installed, the 64-bit version is the default. You can run the 32-bit version by specifying the `-32` switch at the command line, as follows:

```
% idl -32
% envi -32
% envizoom -32
```

Under Microsoft Windows, the 32-bit and 64-bit versions are started via separate Start menu entries.

Graphics Hardware

Some IDL and ENVI Zoom features take advantage of graphics hardware that supports the OpenGL 2.0 interface to improve rendering performance, if such hardware is present. Your video card should support OpenGL 2.0 or higher to take advantage of the graphics features in IDL and ENVI Zoom.

Platform Support Questions and Answers

The ITT Visual Information Solutions Technical Support department maintains a list of common platform and operating system support questions and answers. Please see Tech Tip 3635 on the ITT Visual Information Solutions Web site for additional information.

Software Requirements

The following table describes the software requirements for IDL and ENVI:

Platform	Software Requirements
Windows	Internet Explorer 5.0 or higher
Macintosh	Apple X11 X-Windows manager

Table 3-2: Software Requirements for IDL and ENVI

ION

ION (IDL On the Net) is a family of products that allows you to run IDL-driven applications in a networked environment, giving intranet or Internet users access to IDL visualization and analysis.

The ION family consists of two products:

- ION Script
- ION Java

Hardware and Operating System Requirements

ION Script and ION Java can be used separately or together to create interactive Web applications powered by IDL. These two products are available as options during installation of IDL or ENVI for Windows or UNIX.

ION 6.4 works with IDL 6.4 on the following platforms:

- Microsoft Windows (32-bit)
- SGI IRIX (32-bit)
- Sun Solaris (32-bit)
- Linux (32-bit)

Web Server Requirements for ION

In order to use ION, you must install an HTTP Web server. ION has been tested with the following Web server software:

- Apache Web Server version 2.0 for Windows, Linux, and Solaris
- Apache Web Server version 1.3.14 for IRIX (included with the IRIX operating system)

Web Browser Requirements for ION

ION supports the HTTP 1.0 protocol. The following are provided as examples of popular Web browsers that support HTTP 1.0:

- Mozilla Firefox version 1.0 and later
- Netscape Navigator — Versions 4.7 and later
- Microsoft Internet Explorer — Versions 5.5 and later

Browsers differ in their support of HTML features. As with any Web application, you should test your ION Script or Java application using Web browsers that anyone accessing your application is likely to be using.

Java Virtual Machine Requirements for ION

The following are provided as examples of popular Web browsers that are shipped with the required JVMs:

- Mozilla Firefox version 1.0 and later
- Netscape Navigator versions 4.7 and later
- Microsoft Internet Explorer versions 5.5 and later

Feature Support by Operating System

IDL Features

IDL technologies not listed in this table are assumed to work on all supported platforms.

Feature	Windows		OS X	Linux		Solaris		AIX		IRIX	
	32-bit	64-bit	32-bit	32-bit	64-bit	32-bit	64-bit	32-bit	64-bit	32-bit	64-bit
ActiveX: WIDGET_ACTIVEX (IDLcomActiveX object)	•										
COM Object –Export (via Export Bridge Assistant)	•										
COM Object –Import (IDLcomIDispatch object)	•										
DataMiner	•			•[a]		•		•		•	
DICOM Network Services	•		•[d]	•		•					
DICOM Read/Write (IDLffDicomEx object)	•		•[d]	•		•					
DXF file format (IDLffDXF object)	•			•		•		•		•	
GUIBuilder	•	•									
IDL_IDLBridge	•	•	•	•	•	•	•				
ION	•			•		•				•	
Java Object –Export (via Export Bridge Assistant)	•	•	•[c]	•	•	•	•				

Table 3-3: Feature Support: IDL 6.4

Feature	Windows		OS X	Linux		Solaris		AIX		IRIX	
	32-bit	64-bit	32-bit	32-bit	64-bit	32-bit	64-bit	32-bit	64-bit	32-bit	64-bit
Java Object –Import (IDLjavaObject)[b]	•		•	•		•	•			•	
Motion JPEG2000 (IDLffMJPEG2000)	•	•	•	•	•	•	•				
MrSID (IDLffMrSID)	•										
Network access support: IDLnetURL IDLnetOGCWCS IDLnetOGCWMS	•	•	•	•	•	•	•				
Flexible single-user license	•	•	•	•	•						
Remote Procedure Calls (RPCs)			•	•	•	•	•	•	•	•	•
Semaphores	•	•	•	•	•	•	•			•	•
tty-based interface			•	•	•	•	•	•	•	•	•

Table 3-3: Feature Support: IDL 6.4 (Continued)

[a] DataMiner is not supported on Red Hat Enterprise 4 or Fedora Core 3.

[b] Java Virtual Machine Requirements for the IDL-Java Bridge — IDL supports version 1.3.1 and greater on all platforms except for Macintosh (the supported version is 1.3.x) and SUN SPARC 64-bit (the supported version is 1.4.x and greater).

[c] Graphical Java objects cannot be exported under Macintosh OS X.

[d] DICOM Network Services and DICOM Read/Write functionality are supported with the PowerPC Macintosh IDL binary version, but not with the Intel-based Macintosh binary version. The IDLffDICOM object is supported on all platforms except 64-bit Windows.

ENVI Features

The following ENVI features are provided in addition to the features listed above for IDL. ENVI technologies not listed in this table are assumed to work on all supported platforms.

Feature	Windows		OS X	Linux		Solaris		AIX		IRIX	
	32-bit	**64-bit**	**32-bit**	**32-bit**	**64-bit**	**32-bit**	**64-bit**	**32-bit**	**64-bit**	**32-bit**	**64-bit**
ENVI – FLAASH module [e]	•			•	•	•	•			•	•
ENVI – tape utilities [e]	•		•	•	•	•	•	•	•	•	•
ENVI – reading ECW files [e]	•										
ENVI – reading/writing ESRI GRID files [e]	•										
ENVI – GPS Link [e]	•										
ENVI – Reading MrSID files [e]	•										
ENVI – RemoteView link	•					•	•				
ENVI – Intelligent Digitizer	•	•	•	•	•	•	•				
ENVI Zoom	•	•	•	•	•	•	•				
ENVI Zoom – JPIP/IAS [e]	•		•	•			•				
ENVI Zoom – OGC	•	•	•	•	•	•	•				
NITF in ENVI and ENVI Zoom	•	•				•	•				

Table 3-4: Feature Support: ENVI 4.4

[e] Windows users: these ENVI functions only run in 32-bit mode. If you have a 64-bit Windows PC and you want to use these functions, run ENVI in 32-bit mode by selecting **Start** → **Program Files** → **ENVI *x.x*** → **32-bit** → **ENVI** or **ENVI + IDL**.