



What's New in IDL 7.1

IDL Version 7.1

May 2009 Edition

Copyright © ITT Visual Information Solutions
All Rights Reserved.

Restricted Rights Notice

The IDL®, IDL Advanced Math and Stats™, 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

The software and associated documentation are subject to U.S. export controls including the United States Export Administration Regulations. The recipient is responsible for ensuring compliance with all applicable U.S. export control laws and regulations. These laws include restrictions on destinations, end users, and end use.

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.

ESRI®, ArcGIS®, ArcView®, and ArcInfo® are registered trademarks of ESRI.

Portions of this work are Copyright © 2008 ESRI. All rights reserved.

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 (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.

QUAC and FLAASH are licensed from Spectral Sciences, Inc. under U.S. Patent No. 6,909,815 and U.S. Patent No. 7,046,859 B2.

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 (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 7.1	7
Platform Support Changes	8
64-bit IDL for Intel-based Macintosh	8
IDL for 64-bit Solaris X86 Systems	8
IDLffDicomEx on 64-bit Windows	8
64-bit COM Export Bridge Functionality	9
IDL Workbench Enhancements	10
New Visualize Perspective	10
Automatic Data Import	11
Automatic Path Management Enhancements	11
IDLdoc Integration	13
Command Line View Enhancements	14
Variables View Enhancements	14
Miscellaneous IDL Workbench Features	15

Command Line IDL for Windows	18
iTool Enhancements	19
IOPEN Routine for Automatic Data Import	19
Programmatic iTool Interface	19
Enhancements to the iTool Save/Export Process	20
Switch to IDL Command Line from an iTool	20
IIMAGE and IMAP Accept a Filename Argument	20
Visualization Enhancements	21
WRITE_TIFF Enhancements	21
Analysis Enhancements	22
New Image Processing Filter Routines	22
CENTER keyword to FFT	22
Language Enhancements	24
24-Bit Color PostScript Plotting	24
IDL_BASE64 Function	24
MAKE_RT Enhancements	25
Scientific Data Format Libraries	25
CDF_SET_MD5CHECKSUM Procedure	25
READONLY Keyword to CDF_OPEN	26
Checking Class Names with OBJ_ISA	26
Platform-Specific Sharable Library Naming	26
GLOBAL_SYMBOLS Enhancement to DLM Definition	26
IDL-Java Bridge Reset	26
File Access Enhancements	28
Read and Write CSV Files	28
IDLffDicomEx on 64-bit Windows	28
User Interface Toolkit Enhancements	29
Flat Button Appearance	29
Documentation Enhancements	30
Better Ranking of Search Results	30
Active Example Code	31
F1 Help	31
New IDL Routines	32
Filter Routines	32
iTools Procedural API	32
IDL Routine Enhancements	34

New IDL Object Classes	37
New IDL Object Properties	38
New IDL Object Methods	39

Chapter 2

Requirements for This Release	41
IDL	42
Feature Support by Operating System	44



Chapter 1

Overview of New Features in IDL 7.1

This chapter contains the following topics:

Platform Support Changes	8	User Interface Toolkit Enhancements	29
IDL Workbench Enhancements	10	Documentation Enhancements	30
Command Line IDL for Windows	18	New IDL Routines	32
iTool Enhancements	19	IDL Routine Enhancements	34
Visualization Enhancements	21	New IDL Object Classes	37
Analysis Enhancements	22	New IDL Object Properties	38
Language Enhancements	24	New IDL Object Methods	39
File Access Enhancements	28		

Platform Support Changes

IDL 7.1 is available for Windows, Macintosh OS X, Solaris, and Linux platforms. See “[Hardware Requirements for IDL](#)” on page 42 for complete details.

The following enhancements have been made to IDL’s platform support for the 7.1 release:

64-bit IDL for Intel-based Macintosh

Note

This enhancement was released to IDL for Macintosh customers as IDL 7.0.4.

Beginning with IDL version 7.0.4, IDL can run as a 64-bit application on Macintoshes that run OS X 10.5.1 (Leopard) and later using an Intel 64-bit CPU. As a 64-bit application, IDL can address up to four terabytes of physical memory.

Note that IDL can still run as a 32-bit application on a 64-bit Macintosh if requested. See “[Feature Support by Operating System](#)” on page 44 for details of features that are not supported when IDL runs in 64-bit mode.

IDL for 64-bit Solaris X86 Systems

Beginning with IDL version 7.1, IDL can run in command-line only mode on 64-bit Solaris X86 systems. Note that the following are **not** supported:

- The IDL Workbench
- The IDL online help system
- Any form of IDL on 32-bit Solaris X86 systems

IDLffDicomEx on 64-bit Windows

The IDLffDicomEx object and the IDL DICOM Network Services now allow you to access, read from, and write to DICOM files on 64-bit machines running Microsoft Windows.

Note

The IDLffDicomEx feature requires an additional-cost license key to access the functionality. For more information, contact your ITT Visual Information Solutions sales representative or technical support.

64-bit COM Export Bridge Functionality

IDL's Export Bridge now supports the creation of 64-bit COM connector objects.

IDL Workbench Enhancements

The *IDL Workbench* is a graphical interface to IDL that provides powerful code development and debugging tools in a cross-platform interface that looks and behaves the same way on all platforms supported by IDL.

See *Welcome to the IDL Workbench* in the IDL help system for an introduction.

The following enhancements have been made to the IDL Workbench for the 7.1 release:

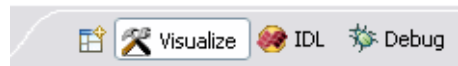
- [New Visualize Perspective](#)
- [Automatic Data Import](#)
- [Automatic Path Management Enhancements](#)
- [IDLdoc Integration](#)
- [Command Line View Enhancements](#)
- [Variables View Enhancements](#)
- [Miscellaneous IDL Workbench Features](#)

New Visualize Perspective

IDL 7.1 introduces the *Visualize Perspective*. The Visualize Perspective is an IDL Workbench perspective that is designed for quick and easy visualization of data contained in files or IDL variables.

Tip

The Visualize Perspective will be shown automatically when you first launch IDL 7.1. To use the IDL perspective familiar from IDL 7.0, simply click the **IDL** button in the upper right corner of the IDL Workbench.



The Visualize Perspective allows you to quickly create visualizations from your data by dragging IDL variables from the **Variables** view to a set of *action bars* contained in the new **Visualization Palette** view.

For more information, see the Visualize Perspective topic in the IDL Online Help.

Automatic Data Import

The IDL Workbench's **File** → **Open File...** menu item has been enhanced to automatically import data from files of known data types. When you select a file by using this menu item, IDL will attempt to read the file and create IDL variables for the data contained therein. Depending on the type of file you select, IDL will either read the contents automatically or prompt you for additional information on how to import the data.

You can use the same automatic data import mechanism programmatically with the new `IOPEN` routine. See [“IOPEN Routine for Automatic Data Import”](#) on page 19 for details.

Enhanced Handling of IDL SAVE Files

In IDL 7.0, opening an IDL SAVE file (`.sav`) using the **File** → **Open File...** menu item (or by double-clicking on the `.sav` file in the Project Explorer) caused IDL to attempt to restore and execute an application contained in the file. Selecting a SAVE file containing variables caused IDL to generate an “Attempt to call undefined procedure or function” error.

Beginning in IDL 7.1, using either the **File** → **Open File...** menu item or the `IOPEN` routine to open a `.sav` file that contains variables will automatically restore the variables in IDL's current scope. Opening a `.sav` file containing routines will continue to attempt to restore and execute an application contained in the file.

Custom Workbench File Readers

If you have a file reader routine designed to import data from files in a custom data format, you can configure the IDL Workbench to recognize those files and perform automatic data import when they are selected. See the Adding Custom File Readers topic in the IDL Online Help for details.

Automatic Path Management Enhancements

Note

This enhancement was released as part of the IDL Workbench 7.0.3 update. If you installed that update, you will not see any additional change in behavior in IDL 7.1.

The introduction of the IDL Workbench and workbench *projects* in IDL 7.0 brought with it the concept of *automatic path management*. Automatic path management means that, for workbench projects that you specify, the workbench will ensure that the project directory is on IDL's search path when the project is open. Automatic path

management is explained in detail in the IDL Path Management topic in the IDL Online Help.

In the IDL 7.0 Workbench, changes to IDL's path were effected by changing the IDL_PATH preference. While this ensured that workbench projects were available both when the IDL Workbench was running and when IDL was used in command line or runtime mode, it had the side effects of resetting the value of the !PATH system variable and the IDL_PATH preference whenever the IDL Workbench updated the path.

Key Changes

The key things to understand about the changes to IDL's automatic path management are:

- The IDL Workbench no longer changes the value of the IDL_PATH preference.
- The IDL Workbench no longer overwrites the value of the !PATH system variable. If you change the value of the !PATH system variable, the IDL Workbench will **add** entries for managed projects at the end of the path specification as described below, but it will not remove any entries.
- If a workbench project has the **Update IDL path when project is opened or closed** preference set, the project is considered to be a *managed project*. For managed projects:
 - If the managed project is open, the IDL Workbench will ensure that the project directory (including any subdirectories) is included in the list of directories specified by the !PATH system variable. If the directory is not present in !PATH, it will be added *at the end* of the path specification.
 - If the managed project is closed, the IDL Workbench will remove the project directory from the list of directories specified by the !PATH system variable *unless* the project directory is also specified by the IDL_PATH preference.

Ensuring your Path Preference is Correct

Prior to the IDL Workbench 7.0.3 update, the workbench modified the value of the IDL_PATH preference with each change to a managed project. As a result, the value of the IDL_PATH preference may include directories you do not wish included. To clean up your IDL_PATH preference, ITT Visual Information Solutions suggests that you do the following after installing IDL 7.1 if you previously installed IDL 7.0 but not the IDL Workbench 7.0.3 update:

1. Open the IDL **Paths** preferences page and **Remove** any managed directories from the list. Note that your IDL_PATH preference should contain, at a minimum, the <IDL_DEFAULT> entry.
2. Apply the changes and exit IDL.
3. Restart IDL and open the IDL **Paths** preferences page to ensure that the IDL_PATH preference contains only directories you have specified.
4. At the IDL Command Line, enter

```
PRINT, !PATH
```

and confirm that the directories for any managed projects appear at the end of the path specification.

What the Changes Mean to You

If you do not manually alter the values of the !PATH system variable or the IDL_PATH preference, or use the contents of managed IDL Workbench project directories from command-line IDL sessions, you should see no change in IDL's behavior.

If you want directories corresponding to managed IDL Workbench projects to be available in IDL's path when you are using IDL in command line mode (that is, when the IDL Workbench is not running), you must add the directories to the IDL_PATH preference manually. You can manage the IDL_PATH preference using either the IDL **Paths** preferences page in the IDL Workbench or the PREF_SET routine.

Similarly, if you want to control the exact location of a managed IDL Workbench project's directory in the !PATH specification, you must add the project directory to the IDL_PATH preference in the desired location. Note that managed project directories added to the IDL_PATH preference are *not* removed from !PATH when the project is closed. If you wish to change the directory's position in the !PATH specification temporarily, you can simply modify the value of the !PATH system variable.

IDLdoc Integration

Note

This enhancement was released as part of the IDL Workbench 7.0.3 update. If you installed that update, you will not see any additional change in behavior in IDL 7.1.

IDLdoc is a third-party, open source text format used within the comment sections of an IDL .pro file. The IDLdoc format allows code comments to be exploited for other

purposes, including generating formatted user documentation (similar to JavaDoc), and creating formatted hover help displays for user-written IDL routines.

Michael Galloy, the creator of IDLdoc, maintains a website to facilitate its exposure and distribution within the IDL community:

<http://idldoc.idldev.com/>

For information on how the IDL Workbench uses IDLdoc-style comments, see the Advanced Code Commenting with IDLDoc topic in the IDL Online Help.

Hover Help

The IDL Workbench already contains the Hover Help feature, in which comments located above an IDL procedure are displayed in a small popup window. IDL's comment extraction utility has been enhanced to convert IDLdoc-style comments into HTML that is optimized for viewing within a small Hover Help popup.

IDLdoc Comment Blocks

The IDL Workbench now includes two default IDLdoc comment blocks that help reduce the work required to use IDLdoc. The Routine template creates a code block for a procedure or function, and the File template creates a code block for an entire .pro file. These templates are accessed through the **IDL → IDLdoc** preferences page, and you can customize them with your own text and variables.

Command Line View Enhancements

The IDL Workbench Command Line View has been enhanced in the following ways:

- A toolbar button on the view toolbar allows you to clear the text from the command line.
- A toolbar button on the view toolbar allows you to perform a reset or full reset of the IDL session (equivalent to issuing the .RESET_SESSION or .FULL_RESET_SESSION executive commands).
- You can drag a variable name from the Variables view directly to the Command Line view.

Variables View Enhancements

The IDL Workbench Variables View has been enhanced in the following ways:

- You can delete one or more variables from the IDL \$MAIN\$ scope by clicking on a toolbar button on the view toolbar, selecting an item from the view menu, or selecting an item from the context menu.
- You can rename a variable by selecting an item from the view menu or from the context menu.
- You can change the value of a scalar variable by selecting an item from the view menu or from the context menu.
- You can drag variable names from the Variables view to the Command Line view, an editor window, or the Visualize Perspective's Tool view.

Miscellaneous IDL Workbench Features

Note

These enhancements were released as part of the IDL Workbench 7.0.3 update. If you installed that update, you will not see any additional change in behavior in IDL 7.1.

Execute Selected Code from the Editor

You can execute a block of `.pro` code that is highlighted in the IDL Workbench editor. After selecting the code block, either select **Run** → **Run Selected Text** or right-click and select **Run Selected Text**.

Open Declarations for Object Self Methods

When using the `self` syntax for referencing object methods, you can immediately display a method declaration inside the `.pro` editor.

Consider the following code block:

```
oTile = self->CreateIterator(/SPATIAL, $
    MASK=oMask, $
    SUB_RECT=subRect, $
    BANDS=bands, TILE_DATA_TYPE=4, $
    OVERLAP=yDim, /INVERSE_MASK)
```

Click inside `CreateIterator`, right-click, and select **Open Declaration** from the context menu. The editor will jump to the class' `CreateIterator()` method declaration.

Note

The self method declaration feature only works if the method is written in .pro code. Methods for built-in classes cannot be queried in this way.

New Formatter Preferences Page

The Formatter preferences page has been created to control the format of IDL code elements for the content assist and code formatting features. Some of these preferences were previously located on the **IDL → Editor** preferences page.

On this preferences page, you can change the appearance of code elements such as reserved words, system routines, keywords, and so on.

Highlight Selected Words in Editor

On the Editor preferences page, you can choose to highlight other occurrences of the currently-selected word or words in the active editor. You can toggle between all the highlighted words by selecting **Occurrences** from the **Next Annotation** or **Previous Annotation** menu on the Workbench toolbar, and clicking **Next Annotation** or **Previous Annotation**.

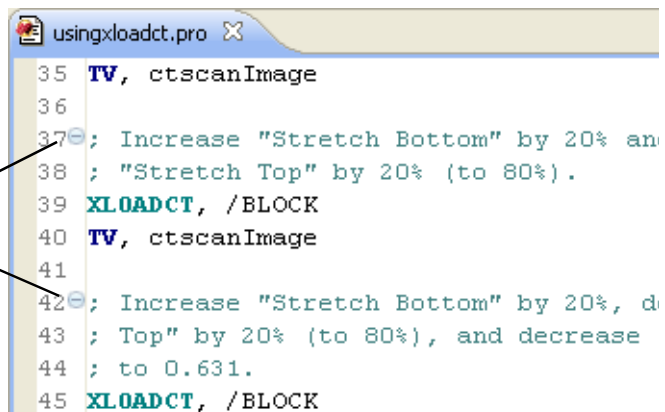
IDL Reserved Words Added to Content Assist

The Content Assist feature now includes IDL reserved words as selectable options.

Expandable Comment Sections

For source code files that are included in an IDL project, the IDL Workbench editor now recognizes comment blocks, and allows you to expand or collapse them with a button in the left margin.

Expand/Collapse
Comment Block Button



Expandable comment sections are enabled by default; to turn them off or back on, right-click in the grey bar at the left edge of the editor window and select **Enable Folding** from the **Folding...** sub-menu.

Command Line Parenthesis/Bracket Matching

In the IDL command line, you can place the cursor to the right of a parenthesis, brace, or bracket, and the matching symbol is highlighted. This new functionality mirrors the parenthesis/bracket matching feature that already exists in the IDL Workbench editor.

Command Line IDL for Windows

IDL 7.1 introduces a command-line-only version of IDL on Microsoft Windows platforms. The command line version allows you to run IDL without the IDL Workbench on Windows systems in the same manner as on Macintosh, Linux, and Solaris systems.

To start the Windows command-line version, do one of the following:

- Select **IDL Command Line** from the IDL 7.1 **Start** menu program group.
- Open a Windows **Command Prompt** window (or other similar shell program) and change directories to the `bin\bin.x86` or `bin\bin.x86_64` directory of your IDL 7.1 installation (usually `C:\Program files\ITT\IDL71`). Type
`idl`
at the command prompt.
- In the Windows Explorer, navigate to the `bin\bin.x86` or `bin\bin.x86_64` directory of your IDL 7.1 installation (usually `C:\Program files\ITT\IDL71`). Double click on `idl.exe` to start IDL.

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....

For details on these additions and other enhancements that have been made to the IDL iTools system for the 7.1 release, see the following topic:

- [IOPEN Routine for Automatic Data Import](#)
- [Programmatic iTool Interface](#)
- [Enhancements to the iTool Save/Export Process](#)
- [Switch to IDL Command Line from an iTool](#)
- [IIMAGE and IMAP Accept a Filename Argument](#)

IOPEN Routine for Automatic Data Import

The IOPEN routine allows you to specify a file to be opened and its contents automatically read into IDL variables.

See [“IOPEN”](#) (*IDL Reference Guide*) for additional details.

Programmatic iTool Interface

Many iTool interactions that relied on mouse action by the user can now be accomplished programmatically. In addition to being able to create iTools (via the existing IPLOT, IIMAGE, and similar routines), you can now programmatically modify existing iTools. New routines allow you to:

- set and retrieve properties of iTool visualizations (ISETPROPERTY, IGETPROPERTY)
- set and retrieve data used to create an iTool visualization (IPUTDATA, IGETDATA)
- create annotations in an existing tool (IELLIPSE, IPOLYGON, IPOLYLINE, ITEXT)
- manipulate visualizations and annotations in an existing tool (IROTATE, ITRANSLATE, ISCALE, IZOOM)

See Using the iTools Procedural Interface in the IDL Online Help for additional details.

Enhancements to the iTool Save/Export Process

The process of saving or exporting the contents of an iTool has been streamlined:

- To save the entire iTool window to a file, select **File** → **Save** in the iTool. Selecting an image file format (.gif, .jpg, etc.) will save the entire contents of the iTool window as an image. Selecting the iTool Save File format (.isv) will save the iTool as a restorable IDL save file.
- To save a bitmap version of a single image contained in an iTool window, select the image and select **Edit** → **Export Image...**

Note

To change the output options for a particular image file format, select **File** → **Preferences...**, then select the image file format from the **File Writers** folder and change the output properties as desired.

- To export data variables used to create the iTool visualization back to IDL, select the visualization and select **Edit** → **Export Data to IDL...**

Switch to IDL Command Line from an iTool

When an iTool window has the focus, you can press **Ctrl-I** or select **Window** → **Show IDL Command Line** to bring the IDL Workbench to the foreground and put the focus in the IDL Command Line view.

IIMAGE and IMAP Accept a Filename Argument

IIMAGE and IMAP now accept a *Filename* argument. Calling these iTools with *Filename* is equivalent to calling **IOPEN** and passing the resulting data into the iTool as the first argument.

Visualization Enhancements

The following enhancements have been made to IDL's visualization functionality for the 7.1 release:

- [WRITE_TIFF Enhancements](#)

WRITE_TIFF Enhancements

The [WRITE_TIFF](#) procedure has the following new keywords:

- **COMPLEX** writes the pixel components as complex floating-point entities (the default is 8-bit).
- **DCOMPLEX** writes the pixel components as double-precision complex floating-point entities (the default is 8-bit).
- **DOUBLE** writes the pixel components as double-precision floating-point entities (the default is 8-bit).
- **L64** writes the pixel components as unsigned 64-bit integer entities (the default is 8-bit). Use the **SIGNED** keyword to write signed integers.
- **SIGNED** if the **SHORT**, **LONG**, or **L64** keywords are set, **SIGNED** writes out signed integers instead of unsigned integers.

Analysis Enhancements

The following enhancements have been made to IDL's data-analysis functionality for the 7.1 release:

- [New Image Processing Filter Routines](#)
- [CENTER keyword to FFT](#)

New Image Processing Filter Routines

Six new image filtering routines have been added to IDL in this release. All are written in the IDL language.

BANDPASS_FILTER — Applies a bandpass filter to a one-channel image.

BANDREJECT_FILTER — Applies a band reject filter on a one-channel image.

ESTIMATOR_FILTER — Applies an order statistic noise-reduction filter to a one-channel image.

LEAST_SQUARES_FILTER — Reduces degradation and noise in an image based on the mean and variance of the degradation and noise.

MEAN_FILTER — Applies mean filter noise reduction on a one- or multi-channel image.

WIENER_FILTER — Reduces degradation and noise in an image by statistically comparing it with an undegraded image.

CENTER keyword to FFT

The **CENTER** keyword to the [FFT](#) procedure calculates a centered Fourier transform. In the forward direction, the resulting Fourier transform has the zero frequency location shifted to the center of the array. In the reverse direction, the input is assumed to be a centered Fourier transform.

This keyword is added as a convenience. The following two examples are equivalent:

Example One: Without the **CENTER** Keyword

```
imageSize = SIZE(image, /DIMENSIONS)
fftUncentered = FFT(image)
center = imageSize/2 + 1
fftCentered = SHIFT(fftUncentered, center)
```

Example Two: With the CENTER Keyword

```
fftCentered = FFT(image, /CENTER)
```

Language Enhancements

The following enhancements have been made to the core language for the 7.1 release:

- [24-Bit Color PostScript Plotting](#)
- [IDL_BASE64 Function](#)
- [MAKE_RT Enhancements](#)
- [Scientific Data Format Libraries](#)
- [CDF_SET_MD5CHECKSUM Procedure](#)
- [READONLY Keyword to CDF_OPEN](#)
- [Checking Class Names with OBJ_ISA](#)
- [Platform-Specific Sharable Library Naming](#)
- [GLOBAL_SYMBOLS Enhancement to DLM Definition](#)
- [IDL-Java Bridge Reset](#)

24-Bit Color PostScript Plotting

The Direct Graphics PostScript device (PS) has been enhanced to allow the use of 24-bit color specifications in plots rendered to a PostScript file. The PS device now accepts the DECOMPOSED keyword to enable 24-bit color plotting. For example, the following code creates a PostScript file that contains a simple plot drawn in blue:

```
SET_PLOT, 'ps'  
DEVICE, /DECOMPOSED, /COLOR  
DEVICE, FILE='c:\Junk\test1.ps'  
PLOT, INDGEN(10), COLOR='ff0000'x  
DEVICE, /CLOSE
```

See “[Color PostScript](#)” (Appendix A, *IDL Reference Guide*) for additional details.

IDL_BASE64 Function

The [IDL_BASE64](#) function uses MIME Base64 encoding to convert a byte array into an Base64 encoded scalar string, or to convert a Base64-encoded scalar string into a byte array.

MAKE_RT Enhancements

The [MAKE_RT](#) routine has been enhanced to create a “generic” launch script if no value is specified for the `SAVEFILE` keyword. This allows you to create a runtime distribution and launch script that is not linked to a specific application; if no `SAVEFILE` is specified, the launch scripts will run IDL in runtime or virtual machine mode and open a file-selection dialog allowing the end-user to select a `.sav` file to restore and run.

Additionally:

- The `OVERWRITE` keyword allows `MAKE_RT` to place files in a non-empty directory
- The `SUNX86_64` keyword creates a distribution for Solaris X86 machines running IDL in 64-bit mode

Scientific Data Format Libraries

Note

The following Scientific Data Format library updates were released as part of the IDL Scientific Data Formats update. If you installed that update to IDL 7.0, you will not see any additional change in behavior in IDL 7.1.

IDL incorporates the following updated third-party Scientific Data Format libraries:

- CDF version 3.2
- HDF version 4.2r3
- HDF5 version 1.6.7

In addition to updating the library versions used by IDL, the following new API features are included:

CDF_SET_MD5CHECKSUM Procedure

The [CDF_SET_MD5CHECKSUM](#) procedure calculates and inserts a MD5 checksum into a CDF file. If you use the `/REMOVE` keyword, a previously-inserted checksum is removed from the file.

READONLY Keyword to CDF_OPEN

Set the READONLY keyword to the [CDF_OPEN](#) function to open the specified CDF file in readonly mode. Opening a file in readonly mode will result in significant performance improvement if there is a large amount of metadata and a large portion of the metadata is read.

Checking Class Names with OBJ_ISA

The [OBJ_ISA](#) function can now test whether an object class is subclassed from another class based on the class name. The previous behavior, whereby OBJ_ISA tests an instance of a class for an “is a” relationship, is unchanged.

Platform-Specific Sharable Library Naming

IDL now attempts to locate a sharable library for the current platform based on a platform-specific file-naming scheme. This allows IDL to differentiate between different versions of a sharable library file created for different operating systems or hardware architectures. A new PLATFORM_EXTENSION keyword to the MAKE_DLL procedure makes it easy to generate sharable library files with platform-specific names.

See “[Dynamically Loadable Modules](#)” (Chapter 15, *External Development Guide*) for additional information.

GLOBAL_SYMBOLS Enhancement to DLM Definition

Including the line GLOBAL_SYMBOLS in a Dynamically Loadable Module’s definition file (.dlm) will cause the shared library to load all of its symbols (functions or procedures) as *globally accessible* rather than locally accessible. If a symbol is globally accessible, then libraries that are loaded later will be able to access the symbol. In practice, adding this line to the DLM file will cause IDL to set the RTLD_GLOBAL flag when calling the `dlopen()` operating system function to load the module.

See “[Dynamically Loadable Modules](#)” (Chapter 15, *External Development Guide*) for additional information.

IDL-Java Bridge Reset

By default, the IDL-Java bridge persists for the duration of the IDL session in which it was created. This means that modifications to items in the Java Classpath are not

loaded until a new IDL session is started. You can modify this policy by modifying the IDL-Java Bridge Configuration file, as described in [“Initializing the IDL-Java Bridge”](#) (Chapter 5, *IDL Connectivity Bridges*).

After you have modified the `Restart Bridge` setting in the bridge configuration file, executing the `.FULL_RESET_SESSION` executive command will also reset the IDL-Java bridge.

File Access Enhancements

The following enhancements have been made to IDL's file-access capabilities in the IDL 7.1 release:

- [Read and Write CSV Files](#)
- [IDLffDicomEx on 64-bit Windows](#)

Read and Write CSV Files

Three new routines allow you to read, write, and query files in comma-separated-values (CSV) format:

- [QUERY_CSV](#)
- [READ_CSV](#)
- [WRITE_CSV](#)

IDLffDicomEx on 64-bit Windows

The IDLffDicomEx object now allows you to access, read from, and write to DICOM files on 64-bit machines running Microsoft Windows.

Note

The IDLffDicomEx feature requires an additional-cost license key to access the functionality. For more information, contact your ITT Visual Information Solutions sales representative or technical support.

User Interface Toolkit Enhancements

The following enhancements have been made to IDL's graphical user interface toolkit in the IDL 7.1 release:

- [Flat Button Appearance](#)

Flat Button Appearance

The [WIDGET_BUTTON](#), [WIDGET_COMBOBOX](#), and [WIDGET_DROPLIST](#) routines have a new FLAT keyword, which creates a button or list widget that appears “flat” rather than with the default three-dimensional appearance.

Note the following:

- For the Combobox and Droplist widgets, the FLAT keyword has no effect on Microsoft Windows platforms.
- For the Button widget, the FLAT keyword has no effect on buttons with text labels.

Documentation Enhancements

The following enhancements have been made to IDL's documentation and online help system in the IDL 7.1 release:

- [Better Ranking of Search Results](#)
- [Active Example Code](#)
- [F1 Help](#)

See *Using IDL Help* in the IDL help system for a complete description of the IDL help interface.

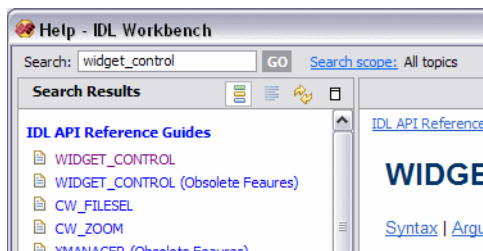
Better Ranking of Search Results


Note

This enhancement was released as part of the IDL Workbench 7.0.3 update. If you installed that update, you will not see any additional change in behavior in IDL 7.1.

In IDL 7.0, the **Search** feature of the IDL Online Help used a default search result ranking algorithm that produced poor results when searching for reference documentation on IDL routines. In particular, many links to reference topics were ranked low in the result listing, while less obviously useful topics appeared at the top.

The search result ranking algorithm has been adjusted so that entering the name of an IDL routine in the **Search** field places the link to that routine's reference page at or near the top of the result list.



Note that the state of the **Show Result Categories** button () can also

influence where in the list of search results an entry appears. Turning the search result categories on can also help you determine what type of topic your search term appears in.

Tip

Entering a ? and the name of an IDL routine at the IDL command line will display the reference documentation for the specified routine directly, without searching.

Active Example Code

Note

This enhancement was released as part of the IDL Workbench 7.0.3 update. If you installed that update, you will not see any additional change in behavior in IDL 7.1.

If you start the IDL Online Help from within the IDL Workbench (as opposed to starting it from an IDL command line session), you will find many blocks of example code that have been highlighted in blue. Clicking on these example blocks will execute the code immediately in IDL.

Note

Not all example code has been made active. The process of making the code blocks active will continue in future releases.

F1 Help

Note

This enhancement was released as part of the IDL Workbench 7.0.3 update. If you installed that update, you will not see any additional change in behavior in IDL 7.1.

You can display information about text inside an editor, the Console view, or the IDL Command Line view by selecting text (or placing the cursor inside a word) and pressing F1. Information about the word or topic is displayed in the IDL Online Help system. You can display the same information by selecting **Help → Help on Selected Item**.

The F1 key binding works on all platforms. You can choose a different key to display help on the selected item by opening the **General → Keys** preference page (or by pressing Ctrl-Shift-L twice), and changing the key binding for the **Help on Selected Item** command.

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_SET_MD5CHECKSUM — Calculates and inserts a MD5 checksum into a CDF file.

IDL_BASE64 — Uses MIME Base64 encoding to convert a byte array into an Base64 encoded scalar string, or to convert a Base64-encoded scalar string into a byte array.

QUERY_CSV — Obtains information about a comma-separated values (CSV) file.

READ_CSV — Reads data from an ASCII file containing comma-delimited data.

WRITE_CSV — Writes comma-separated-values file (CSV).

Filter Routines

BANDPASS_FILTER — Applies a bandpass filter to a one-channel image.

BANDREJECT_FILTER — Applies a band reject filter to a one-channel image.

ESTIMATOR_FILTER — Applies an order statistic noise-reduction filter to a one-channel image.

LEAST_SQUARES_FILTER — Reduces degradation and noise in an image based on the mean and variance of the degradation and noise.

MEAN_FILTER — Applies mean filter noise reduction on a one- or multi-channel image.

WIENER_FILTER — Reduces degradation and noise in an image by statistically comparing it with an undegraded image.

iTools Procedural API

ICONVERTCOORD — Converts between iTool coordinate systems.

IDELETE — Deletes a tool in the IDL Intelligent Tools system. (Note that this routine was previously named ITDELETE.)

IELLIPSE — Creates an ellipse annotation in an existing iTool.

IGETCURRENT — Retrieves the iTool identifier of the current tool in the IDL Intelligent Tools system.

IGETDATA — Retrieves data from a specified iTools visualization.

IGETID — Retrieves the full iTool object identifier string for an iTool visualization item based on a portion of the identifier.

IGETPROPERTY — Retrieves a property value from an iTools visualization.

IOPEN — Opens a file, reads data from the file, and adds one or more variables containing the data to IDL's current scope.

IPOLYGON — Creates a polygon annotation in an existing iTool.

IPOLYLINE — Creates a line annotation in an existing iTool.

IPUTDATA — Replaces the data on a specified iTools visualization without changing other properties of the visualization.

IREGISTER — Registers iTool object classes or other iTool functionality with the IDL Intelligent Tools system. (Note that this routine was previously named ITREGISTER.)

IRESET — Resets the IDL iTools session. (Note that this routine was previously named ITRESET.)

IREOLVE — Resolves all IDL code within the iTools directory, as well as all other IDL code required for the iTools framework. (Note that this routine was previously named ITRESOLVE.)

IROTATE — Rotates an iTools visualization.

ISAVE — Saves the contents of the current iTool as an image file.

ISCALE — Scales an iTools visualization.

ISETCURRENT — Sets the current tool in the IDL Intelligent Tools system. (Note that this routine was previously named ITCURRENT.)

ISETPROPERTY — Sets a property on one or more iTools visualizations.

ITEXT — Creates a text annotation in an existing iTool.

ITRANSLATE — Translates an iTools visualization.

IZOOM — Changes the zoom factor on an iTools visualization.

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.

FFT — The FFT procedure has the following new keywords:

- **CENTER** calculates a centered Fourier transform.

ICONTOUR, **IIMAGE**, **IMAP**, **IPLLOT**, **ISURFACE**, **IVECTOR**, **IVOLUME** — The iTool routines have the following new keywords:

- **CURRENT_ZOOM** is the zoom factor to be used for the current view.
- **DEPTHcue_BRIGHT** is the distance in the Z plane at which the objects in the view begin to fade into the background color.
- **DEPTHcue_DIM** is the distance in the Z plane at which the objects in the view have completely faded into the background color.
- **FONT_COLOR** is the color of the text in the title annotation created by the **TITLE** keyword.
- **FONT_NAME** is the name of the font used for the title annotation created by the **TITLE** keyword.
- **FONT_SIZE** is the point size of the font used for the title annotation created by the **TITLE** keyword.
- **FONT_STYLE** is the style of the font used for the title annotation created by the **TITLE** keyword.
- **STRETCH_TO_FIT** indicates whether the visualizations should be stretched to fit within the view.
- **TITLE** now creates a text annotation in the iTool window rather than specifying the text for the iTool window's title bar.
- **WINDOW_TITLE** creates a title in the iTool window title bar.
- **XMARGIN** is the horizontal margin size between the edge of the view and the edge of the visualization.
- **YMARGIN** is the vertical margin size between the edge of the view and the edge of the visualization.
- **ZOOM_ON_RESIZE** changes the size of visualizations when the window is resized.

IIMAGE — The IIMAGE procedure has the following new arguments and keywords:

- The *Filename* argument is the name of a file that contains image data.

INTERPOL — •The INTERPOL function has the following new arguments:

- *XOUT* contains the abscissa values for the result. (Replaces the *U* argument.)

MAKE_DLL — The MAKE_DLL procedure has the following new keywords:

- PLATFORM_EXTENSION causes MAKE_DLL to create a library file with a platform-specific file name.

MAKE_RT — The MAKE_RT procedure has the following new keywords:

- OVERWRITE overwrites any existing files in the output directory.
- SUNX86_64 creates a distribution for Solaris X86 machines running IDL in 64-bit mode.

MAP_PROJ_INIT — The MAP_PROJ_INIT function has the following new keywords:

- ELLIPSOID sets the spheroid. (Replaces the DATUM keyword.)

MOMENT — The MOMENT function has the following new keywords:

- MAXMOMENT limits the number of moments calculated.

OBJ_ISA — The OBJ_ISA function now accepts string class names in addition to object references when testing an object class for an “is a” relationship.

WIDGET_BUTTON — The WIDGET_BUTTON procedure has the following new keywords:

- FLAT specifies that the button be created with a “flat” appearance rather than the default three-dimensional appearance.

WIDGET_COMBOBOX — The WIDGET_COMBOBOX procedure has the following new keywords:

- FLAT specifies that the combobox be created with a “flat” appearance rather than the default three-dimensional appearance. Note that this keyword has no effect on Microsoft Windows systems.

WIDGET_DROPLIST — The WIDGET_DROPLIST procedure has the following new keywords:

- FLAT specifies that the droplist be created with a “flat” appearance rather than the default three-dimensional appearance. Note that this keyword has no effect on Microsoft Windows systems.

WRITE_TIFF — The WRITE_TIFF procedure has the following new keywords:

- **COMPLEX** writes the pixel components as complex floating-point entities (the default is 8-bit).
- **DCOMPLEX** writes the pixel components as double-precision complex floating-point entities (the default is 8-bit).
- **DOUBLE** writes the pixel components as double-precision floating-point entities (the default is 8-bit).
- **L64** writes the pixel components as unsigned 64-bit integer entities (the default is 8-bit). Use the **SIGNED** keyword to write signed integers.
- **SIGNED** if the **SHORT**, **LONG**, or **L64** keywords are set, **SIGNED** writes out signed integers instead of unsigned integers.

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.

IDLgrTextEdit — The IDLgrTextEdit object class is an extension of the [IDLgrText](#) object class that provides several methods that allow you to delete text from an existing text object, insert text into an existing text object, and move the cursor location within an existing text object. The IDLgrTextEdit object class exposes all of the properties and methods of the IDLgrText superclass.

Consider using an IDLgrTextEdit object if your application will allow interactive editing of text objects.

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.

IDLgrText — This object includes the following new properties:

- **DRAW_CURSOR** indicates whether the insertion cursor is drawn when no text is selected.
- **SELECTION_LENGTH** specifies the length of a text selection, starting from the cursor.
- **SELECTION_START** specifies the character index at which a text selection starts.

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.

IDLgrText::GetIndexAtCoord — This method returns the index of the character in the string closest to a specified screen coordinate.

IDLgrText::GetIndexRelativeTo — This method returns the index of the character in the string closest to a specified character index.

IDLitTool::GetThumbnail — This method returns a thumbnail image of the current iTool.



Chapter 2

Requirements for This Release

This section describes supported platforms and software requirements.

IDL	42	Feature Support by Operating System	44
-----------	----	---	----

IDL

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.

Platform	Vendor	Hardware	Operating System	Supported Versions
Windows	Microsoft	Intel/AMD x86 32-bit	Windows	XP SP2, Vista
		Intel/AMD x86_64 64-bit	Windows	XP SP2, Vista
Macintosh ^a	Apple	PowerPC 32-bit	OS X	10.5.1
	Apple	Intel 32-bit	OS X	10.5.1
	Apple	Intel 64-bit	OS X	10.5.1
UNIX ^a	SUN	SPARC 32-bit	Solaris ^c	10
	SUN	SPARC 64-bit	Solaris ^c	10
	SUN	Intel/AMD x86_64 64-bit	Solaris ^e	10
	various	Intel/AMD x86 32-bit	Linux ^{b, c}	Kernel version 2.6.9 glibc version 2.3.4 gtk2 version 2.4.13
	various	Intel/AMD x86_64 64-bit	Linux ^{b, c, d}	

Table 2-1: Hardware Requirements for IDL

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

^b The LINUX version of IDL is built on RedHat 4. If your version of Linux is compatible with the listed kernel and glibc versions, you should be able to install and run IDL.

^c The IDL Workbench requires the GTK+ library version 2.4 or later. Printing functionality in the Workbench requires GTK+ library version 2.10 or later. If GTK+ version 2.4 or later is not available, IDL will only run in console mode.

^d On 64-bit Linux systems, the IDL Workbench requires that your platform's 32-bit compatibility libraries be installed.

^e The IDL Workbench and Online Help are not supported on Solaris x86 64-bit machines. Documentation is available on the DVD for Solaris x86 platforms.

32-bit and 64-bit Versions

On UNIX and Macintosh Intel platforms that provide 64-bit support, you can run IDL 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
```

or

```
% idlde -32
```

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

On Macintosh Intel 64-bit machines, you can double-click on the 32-bit Macintosh applescript (.app) files.

Graphics Hardware

Some IDL 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 these features. Be sure to update your video card drivers with the most recent version.

Platform Support Questions and Answers

This topic describes the platform support requirements for IDL 7.1. Platform support requirements change over time; for the most current information, visit the ITT Visual Information Solutions web site:

<http://ittvis.com/ProductServices/IDL/PlatformSupport.aspx>.

Software Requirements

The following table describes the software requirements for IDL:

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

Table 2-2: Software Requirements for IDL

Feature Support by Operating System

The following table shows marks indicating which platforms support the corresponding feature. IDL technologies not listed in this table are assumed to work on all supported platforms.

Feature	Windows			OS X			Linux		Solaris		
	Intel 32-bit	Intel 64-bit	PPC 32-bit	Intel 32-bit	Intel 64-bit		Intel 32-bit	Intel 64-bit	SPARC 32-bit	SPARC 64-bit	Intel 64-bit
ActiveX: WIDGET_ACTIVEX (IDLcomActiveX object)	•										
COM Object –Export (via Export Bridge Assistant)	•	•									
COM Object –Import (IDLcomIDispatch object)	•										
DataMiner	•	•					•	•		•	
DICOM Network Services	•	•	•				•		•		
DICOM Read/Write (IDLffDicomEx object)	•	•	•				•		•		
DICOM Read (IDLffDicom object)	•		•	•			•	•	•	•	
DXF file format (IDLffDXF object)	•						•		•		
IDL Advanced Math and Stats (IMSL Numerical Library)	•		•	•			•	•	•	•	
MrSID (IDLffMrSID)	•										
Remote Procedure Calls (RPCs)			•	•			•	•	•	•	•
IDL Workbench and Online Help	•	•	•	•	•		•	•	•	•	

Table 2-3: Feature Support: IDL 7.1