IDL

What's New in IDL 7.0

Contents

Chapter 1
Overview of New Features in IDL 7.0

This chapter contains the following topics:

IDL Workbench

The *IDL Workbench* is a new graphical interface to IDL that replaces the IDL Development Environment included with IDL versions prior to 7.0. The IDL Workbench 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.

Platform Support Changes

IDL 7.0 is available for Windows, Macintosh OS X, Solaris, and Linux platforms. See for complete details.

The following enhancements have been made to IDL's platform support for the 7.0 release:

Java Import Bridge: 64-Bit Linux and Windows Support

The Java Import Bridge is now supported on all IDL platforms.

Java Import and Export Bridges: JDK/JRE Version Update

When using the Java Import and Export bridges, ensure that your application uses version 1.5 or later of the Java Runtime Environment (JRE). When creating Java objects for use by IDL, use the Java Software Development Kit (JDK) version 1.5 or later.

Note

On Macintosh machines, the version of Java installed along with the operating system should be sufficient, whatever its version number.

Language Enhancements

The following enhancements have been made to the core language for the 7.0 release:

- GET_LOGIN_INFO
- MAKE_RT
- ROUTINE_FILEPATH
- IDL_START_DIR Preference

GET_LOGIN_INFO

The GET_LOGIN_INFO function returns the name of the computer running the IDL process, as well as the user's login name. See GET_LOGIN_INFO for details.

MAKE_RT

The MAKE_RT procedure creates a stand-alone IDL runtime distribution for one or more platforms. In addition to the IDL binary and resource files required to execute an IDL program in runtime mode, MAKE_RT creates platform-specific launching scripts that allow an end-user to easily launch a runtime application. See MAKE_RT for details.

ROUTINE_FILEPATH

The ROUTINE_FILEPATH function returns the full path to a currently-compiled procedure or function (or a list of procedures and functions). See ROUTINE_FILEPATH for details.

IDL_START_DIR Preference

The IDL_START_DIR system preference replaces two obsolete system preferences: IDL_MDE_START_DIR and IDL_WDE_START_DIR. The value of this preference determines IDL's initial current working directory when you are using the IDL Workbench.

Note ───
The IDL_START_DIR preference is ignored when running IDL in command-line mode; in this situation, IDL's initial current working directory is the directory from which IDL was started.
───

See IDL_START_DIR for details. See "Obsolete IDE Preferences" on page 18 for information system preferences that have been removed.

Documentation Enhancements

In addition to documentation for new and enhanced IDL features, the IDL online help system has been redesigned to use the IDL Workbench's native Eclipse-based help system. The new system replaces the IDL Assistant-based help system used in IDL versions 6.2-6.4, and provides better integration between the help content and IDL itself. As a result, the IDL help content can now contain "active" hyperlinks that execute IDL code or make changes in the IDL Workbench interface.

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

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.

GET_LOGIN_INFO — Returns the name of the computer running the IDL process, as well as the user's login name.

MAKE_RT — Creates a stand-alone IDL runtime distribution for one or more platforms.

ROUTINE_FILEPATH — Returns the full path to a currently-compiled procedure or function (or a list of procedures and functions).

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.

PROFILER — The PROFILER procedure has the following new keyword:

- FILENAME specifies the path to a text file that will contain the profiler results.

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.

IDLgrBuffer::QueryRequiredTiles
IDLgrClipboard::QueryRequiredTiles
IDLgrPrinter::QueryRequiredTiles
IDLgrWindow::QueryRequiredTiles — These methods have the following new keyword:

- PATH specifies the path in the graphics hierarchy used to compute the visible tiles. Each path object reference specified with this keyword must contain an alias. Tile visibility is computed for the version of the image object falling within that path. If this keyword is not set, the PARENT properties determine the path from the current object to the top of the graphics hierarchy and no alias paths are pursued.

Backward Compatibility

In the transition from the pre-version 7.0 development environments to the IDL Workbench, we have tried to replicate or enhance the features you've come to expect from IDL. In some cases, however, we have altered or deprecated existing functionality. See "Obsolete Features" on page 16 for details.

This section describes features whose behavior has been altered since IDL 6.4. If your IDL code relies on any of the features whose behavior has changed, you may need to modify your existing code to run it under IDL 7.0.

Changes to the ONLINE_HELP Procedure

Beginning in IDL 7.0, IDL's online help system uses the cross-platform help viewer support provided byt he Eclipse-based IDL Workbench. As a result, the behavior of the ONLINE_HELP procedure has changed in the following ways:

- No support is provided for displaying user-created topics in IDL's default help viewer. Existing calls to ONLINE_HELP to display topics in the IDL Assistant help viewer (IDL's default help viewer for versions 6.2 - 6.4), a web browser, PDF viewer, or Windows help viewer will all work as they did in previous releases.

    Note ───

    We hope to incorporate support for user-created content in the default IDL help viewer in a future release.

    ───

- The behavior of ONLINE_HELP when the BOOK keyword is not specified has been modified. Previously, IDL would search the directories specified by the !HELP_PATH system variable for files that matched the argument to ONLINE_HELP. Beginning in IDL 7.0, IDL will only search in IDL's own help system for matching topics.

- The behavior of ONLINE_HELP when the BOOK keyword is specified has changed in the following way: if no file extension is provided to the value of the BOOK keyword, ONLINE_HELP will no longer find topics from IDL's own help system. The behavior with respect to user-created content is unchanged.

Chapter 2
Features Obsoleted in IDL 7.0

The following features were present in IDL Version 6.4 but became obsolete in Version 7.0. Obsoleted features should not be used in new IDL code.

Obsolete Features

In the transition from the pre-version 7.0 development environments to the IDL Workbench, we have tried to replicate or enhance the features you've come to expect from IDL. In some cases, however, we have altered or deprecated existing functionality. Specifically:

IDLDE Macros

The IDL Development Environment macro capability is no longer available. In some limited cases, code templates may provide a substitute.

GUI Builder

The IDL GUI Builder is no longer included. The GUI Builder was a Windows-only feature that did not interoperate with the IDL Workbench.

Projects

The pre-version 7.0 *IDL Projects* mechanism has been replaced by a new IDL Workbench project mechanism based on Eclipse features. See the *Working with Projects* tutorial in the IDL online help for additional information.

Runtime Distribution Export

The ability to export an IDL runtime distribution using the project mechanism is no longer available. While we hope to add the ability to create an IDL runtime distribution to the IDL Workbench in a future release, we have added a utility routine, MAKE_RT, to allow you to create runtime distributions from the IDL command line.

Preferences and Preference Migration

IDL system preferences that affected the Windows and Motif IDL Development Environment are obsolete. See "Obsolete IDE Preferences" on page 18 for details.

Beginning with IDL 7.0, PREF_MIGRATE is no longer called automatically the first time IDL is run. You can still use PREF_MIGRATE to manually migrate IDL system preferences. IDL Workbench preferences an managed separately, by the workbench itself. For more on the difference between system and workbench preferences, see "About IDL System Preferences" on page 5530.

X Window System Resources

Previous versions of IDL used the values of numerous X Window System resources to modify the appearance and behavior of the Motif IDL Development Environment. The relevant resources were prefaced with the identifier `Idl*idlde`; they were defined in a system-wide resource file

```
<IDL_DIR>/resource/X11/lib/app-defaults/Idl
```

and in individual users' `~/.idlde` files.

With the switch to the Eclipse-based IDL Workbench, all of the Motif IDLDE X resources are obsolete, and have been removed from the `Idl` resource file. Individual users' `~/.idlde` files will no longer be read when IDL starts up. Flags specifying the values of X resources are no longer accepted at the UNIX command line when launching IDL.

X resources that apply to IDL widgets and graphics windows are still used. These resources are prefaced with the identifier `Idl*` (no `idlde`). The system-wide `Idl` resource file is still present, and individual users can modify the `~/.Xdefaults` file to contain resource definitions for the `Idl*` resources.

Obsolete IDE Preferences

The following IDL system preferences are no longer supported. Most of the obsolete preferences are replaced by IDL Workbench preferences. As a result, these values can be set within the IDL Workbench interface via the **Preferences** dialog, but not from within IDL using the IDL system preferences mechanism.

All of the obsolete IDE preferences have the prefix IDL_MDE (Motif Development Environment) or IDL_WDE (Windows Development Environment). The table below lists all of the obsolete IDE preferences along with the IDL Workbench equivalents, if any. Note that although an IDE preference may have only been available for one of the two Development Environments, the IDL Workbench equivalents are valid on all IDL platforms.

Obsolete IDE Preference	IDL Workbench Equivalent
IDL_MDE_EDIT_BACKUP IDL_WDE_EDIT_BACKUP	The local history of a file is maintained when you create or modify a file. Each time you edit and save a file, a copy of it is saved. This allows you to replace the file with a previous state.
IDL_WDE_EDIT_[B|F]COLOR_*	The background and foreground colors for various categories of text are set in the **Preferences** dialog, in the **IDL > Syntax Coloring** section.
IDL_WDE_EDIT_CHROMACODE	None.
IDL_MDE_EDIT_COMPILE_OPTION IDL_WDE_EDIT_COMPILE_OPTION	1. In the **Preferences** dialog, select **IDL** > **Editor**. 2. Select **Enable Compile on Save** to automatically compile files when you save them.

Obsolete IDE Preference	IDL Workbench Equivalent
IDL_MDE_EDIT_CWD IDL_WDE_EDIT_CWD	To change the IDL process' current working directory to the directory containing the file just opened: 1. In the **Preferences** dialog, select **IDL > Editor**. 2. Select **Change directory on file open**. 3. Click **OK**.
IDL_WDE_EDIT_FONT	The general editor font is set in the **Colors and Fonts** section of the **Preferences** dialog. To set the general editor font: 1. In the **Preferences** dialog, select **General > Appearance > Colors and Fonts**. 2. Open the **Basic** folder and select **Text Font**. 3. Click **Change**. 4. On the **Font** dialog, choose the font options. 5. Click **OK** on both dialogs.
IDL_WDE_EDIT_OPEN_ON_ DEBUG	None. **Note -** When execution halts due to an error, the file and line number at which the error occurred are shown in the Console view as a hyperlink. Clicking the link opens the file automatically.
IDL_MDE_EDIT_READONLY IDL_WDE_EDIT_READONLY	1. In the Project Explorer view, right-click a file and select **Properties**. 2. On the **Properties** dialog, select **Read only**. 3. Click **OK**.

Obsolete IDE Preference	IDL Workbench Equivalent
IDL_WDE_EDIT_TAB_ENAB LE	To determine whether a tab character or spaces are inserted when pressing the **Tab** key in the IDL editor: 1. In the **Preferences** dialog, select **IDL > Editor**. 2. If you don't want to use tab characters, select **Use spaces instead of tabs**. 3. Click **OK**.
IDL_WDE_EDIT_TAB_SP_ON _SAVE	None.
IDL_WDE_EDIT_TAB_WIDT H	To set the number of spaces used to display a tab character: 1. In the **Preferences** dialog, select **IDL > Editor**. 2. Change the **Displayed tab width** value. 3. Click **OK**.
IDL_MDE_EXIT_CONFIRM IDL_WDE_EXIT_CONFIRM	To exit the Workbench, select **File > Exit** from the menu bar or close the workbench with the window close button (x). When the latter option is used a prompt will ask if you really wish to exit the Workbench. To enable or disable the exit prompt: 1. Select **Window > Preferences**. 2. On the **Preferences** dialog, select **General > Startup and Shutdown**. 3. Select **Confirm exit when closing last window**. 4. Click **OK**.

Obsolete IDE Preference	IDL Workbench Equivalent
IDL_WDE_INPUT_FONT	To set the command line font: 1. Select **Window > Preferences**. 2. On the **Preferences** dialog, select **General > Appearance > Colors and Fonts**. 3. Open the **IDL** folder and select **Command Line font**. 4. Click **Change**. 5. On the **Font** dialog, choose the font options. 6. Click **OK** on both dialogs.
IDL_WDE_LOG_FONT	To set the Console view font: 1. Select **Window > Preferences**. 2. On the **Preferences** dialog, select **General > Appearance > Colors and Fonts**. 3. Open the **Debug** folder and select **Console font**. 4. Click **Change**. 5. On the **Font** dialog, choose the font options. 6. Click **OK** on both dialogs.
IDL_MDE_LOG_LINES IDL_WDE_LOG_LINES	You cannot control the number of lines written to the Console view, but you can set the maximum number of characters written: 1. Select **Window > Preferences**. 2. On the **Preferences** dialog, select **Run/Debug > Console**. 3. Select **Limit console output**. 4. Enter a **Console buffer size** (in characters). 5. Click **OK**.

Obsolete IDE Preference	**IDL Workbench Equivalent**
IDL_MDE_LOG_TRIM IDL_WDE_LOG_TRIM	None.
IDL_MDE_SPLASHSCREEN IDL_WDE_SPLASHSCREEN	None.
IDL_MDE_START_DIR IDL_WDE_START_DIR	These preferences are replaced by the IDL_START_DIR preference. To set the IDL_START_DIR preference, either use the PREF_SET routine or the IDL Workbench Prererences dialog: 1. In the **Preferences** dialog, select **IDL**. 2. Type a directory in the **Initial working directory** box, or browse for a directory. 3. Click **OK**.

Chapter 3
Requirements for This Release

This section describes supported platforms and software requirements.

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, Vista
		Intel/AMD x86_64 64-bit	Windows	XP, Vista
Macintosh [a]	Apple	PowerMac G4, G5 32-bit	OS X	10.4, 10.5
	Apple	Intel Core Duo 32-bit	OS X	10.4[b], 10.5
UNIX [a,d]	SUN	SPARC 32-bit	Solaris	10
	SUN	SPARC 64-bit	Solaris	10
	various	Intel/AMD x86 32-bit	Linux [c]	Kernel version 2.6.9 glibc version 2.3.4 gtk2 version 2.4.13
	various	Intel/AMD x86_64 64-bit	Linux [c, e]	Kernel version 2.6.9 glibc version 2.3.4 gtk2 version 2.4.13

Table 3-1: 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] 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.

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

[d] 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.

[e] IDL may not install correctly on a pure 64-bit Linux system. If installation fails, install your platform's 32-bit compatibility libraries.

32-bit and 64-bit Versions

On UNIX 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
```

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

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 the graphics features in IDL.

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:

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

Table 3-2: Software Requirements for IDL

Feature Support by Operating System

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

Feature	Windows		OS X	Linux		Solaris	
	32-bit	**64-bit**	**32-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	●		●[b]	●		●	
DICOM Read/Write (IDLffDicomEx object)	●		●[b]	●		●	
DICOM Read/Write (IDLffDicom object)	●		●	●	●	●	●
DXF file format (IDLffDXF object)	●			●		●	
IDL Analyst (IMSL Numerical Library)	●		●	●	●	●	●
MrSID (IDLffMrSID)	●						
Flexible single-user license, student license	●	●	●	●	●	●	
Remote Procedure Calls (RPCs)			●	●	●	●	●
tty-based interface			●	●	●	●	●

Table 3-3: Feature Support: IDL 7.0

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

[b] 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.