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.


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/).


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 2
Introducing IDL ................................................................................................................. 7
Overview of IDL ...................................................................................................................... 8
Supported File Formats ........................................................................................................... 10
Launching IDL ........................................................................................................................ 13
Launching the iTools ............................................................................................................... 15
Environment Variables Used by IDL ...................................................................................... 18
Command Line Options for IDL Startup .............................................................................. 22
Startup Files ............................................................................................................................ 29
Message of the Day Files ........................................................................................................ 31
Using Your Mouse with IDL ................................................................................................. 32
Using Keyboard Accelerators ............................................................................................... 33
Getting Help with IDL ............................................................................................................ 35
Typographical Conventions ................................................................................................. 45
Quitting IDL ............................................................................................................................ 46
Chapter 3
The IDL Development Environment ............................................. 51
Components of the IDLDE .......................................................... 52
File Menu ................................................................................. 59
Edit Menu .................................................................................. 63
Search Menu ............................................................................. 65
Run Menu .................................................................................. 67
Project Menu ............................................................................. 72
Macros Menu ............................................................................ 73
Window Menu ............................................................................ 75
Help Menu ................................................................................ 78
Printing in IDL .......................................................................... 79
IDL Printer Setup in UNIX or Mac OS X .................................... 80

Chapter 4
Setting IDL Preferences ............................................................. 91
About IDL Preferences .............................................................. 92
Customizing IDL ....................................................................... 93
General Preferences ................................................................. 95
Layout Preferences ................................................................... 98
Graphics Preferences ............................................................... 102
Editor Preferences ................................................................... 105
Startup Preferences ................................................................. 108
Font Preferences ..................................................................... 110
Path Preferences ..................................................................... 113

Chapter 5
Creating Development Environment Macros ......................... 115
What Are Macros? .................................................................. 116
Creating UNIX Macros ........................................................... 117
Creating Windows Macros ....................................................... 120
Command Stream Substitutions .............................................. 122
Building IDL Example Macros ............................................... 123
This chapter includes information about IDL, the IDL documentation set, and how to contact ITT Visual Information Solutions Technical Support. The following topics are covered in this chapter:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview of IDL</td>
<td>8</td>
</tr>
<tr>
<td>Supported File Formats</td>
<td>10</td>
</tr>
<tr>
<td>Launching IDL</td>
<td>13</td>
</tr>
<tr>
<td>Launching the iTools</td>
<td>15</td>
</tr>
<tr>
<td>Environment Variables Used by IDL</td>
<td>18</td>
</tr>
<tr>
<td>Command Line Options for IDL Startup</td>
<td>22</td>
</tr>
<tr>
<td>Startup Files</td>
<td>29</td>
</tr>
<tr>
<td>Message of the Day Files</td>
<td>31</td>
</tr>
<tr>
<td>Using Your Mouse with IDL</td>
<td>32</td>
</tr>
<tr>
<td>Using Keyboard Accelerators</td>
<td>33</td>
</tr>
<tr>
<td>Getting Help with IDL</td>
<td>35</td>
</tr>
<tr>
<td>Typographical Conventions</td>
<td>45</td>
</tr>
<tr>
<td>Quitting IDL</td>
<td>46</td>
</tr>
<tr>
<td>Reporting Problems</td>
<td>47</td>
</tr>
</tbody>
</table>
Overview of IDL

IDL (the Interactive Data Language) is a complete computing environment for the interactive analysis and visualization of data. IDL integrates a powerful, array-oriented language with numerous mathematical analysis and graphical display techniques. Programming in IDL is a time-saving alternative to programming in FORTRAN or C. Using IDL, tasks which require days or weeks of programming with traditional languages can be accomplished in hours. You can explore data interactively using IDL commands and then create complete applications by writing IDL programs.

Analysis advantages include:

- Many numerical and statistical analysis routines—including Numerical Recipes routines—are provided for analysis and simulation of data. Compilation and execution of IDL commands provides instant feedback and hands-on interaction.
- Operators and functions work on entire arrays (without using loops), simplifying interactive analysis and reducing programming time.
- IDL’s flexible input/output facilities allow you to read any type of custom data format. See “Supported File Formats” on page 10 for details.

Visualization advantages include:

- Rapid 2D plotting, multi-dimensional plotting, volume visualization, image display, and animation allow immediate observation of your computation’s results.
- Support for OpenGL-based hardware accelerated graphics.

Application development advantages include:

- IDL is a complete, structured language that can be used interactively and to create sophisticated functions, procedures, and applications.
- IDL’s Intelligent Tools (iTools) can be customized with your own operations or data manipulations.
- IDL widgets can be used to quickly create multi-platform graphical user interfaces to your IDL programs.
- Existing FORTRAN and C routines can be dynamically-linked into IDL to add specialized functionality. Alternatively, C and FORTRAN programs can call IDL routines as a subroutine library or display engine.
Chapter 2: Introducing IDL

- IDL programs run across all supported platforms (UNIX, Macintosh and Microsoft Windows) with little or no modification. This application portability allows you to easily support a variety of computers.
Supported File Formats

IDL supports accessing the following types of file formats.

**Image File Formats**

For specific routine and object information used in IDL to access these type of files, see the “Image Data Formats” category under “Input/Output” (*IDL Quick Reference*).

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMP</td>
<td>Windows Bitmap format</td>
</tr>
<tr>
<td>DICOM</td>
<td>Digital Imaging and Communications in Medicine</td>
</tr>
<tr>
<td>GeoTIFF</td>
<td>TIFF file with tags containing geographic data</td>
</tr>
<tr>
<td>GIF</td>
<td>Graphics Interchange Format</td>
</tr>
<tr>
<td>Interfile</td>
<td>Interfile version 3.3 format</td>
</tr>
<tr>
<td>JPEG</td>
<td>Joint Photographic Experts Group format</td>
</tr>
<tr>
<td>JPEG 2000</td>
<td>JPEG 2000 format</td>
</tr>
<tr>
<td>Motion JPEG2000</td>
<td>Motion JPEG2000 format</td>
</tr>
<tr>
<td>MPEG</td>
<td>Moving Picture Experts Group format</td>
</tr>
<tr>
<td>MrSID</td>
<td>Multi-resolution Seamless Image Database format</td>
</tr>
<tr>
<td>NRIF</td>
<td>NCAR Raster Interchange Format</td>
</tr>
<tr>
<td>PICT</td>
<td>Macintosh version 2 PICT files (bitmap only)</td>
</tr>
<tr>
<td>PNG</td>
<td>Portable Network Graphics format</td>
</tr>
<tr>
<td>PPM</td>
<td>PPM/PGM format</td>
</tr>
<tr>
<td>SRF</td>
<td>Sun Raster File format</td>
</tr>
<tr>
<td>TIFF</td>
<td>8-bit or 24-bit Tagged Image File format</td>
</tr>
<tr>
<td>X11 Bitmap</td>
<td>X11 Bitmap format used for reading bitmaps for</td>
</tr>
<tr>
<td></td>
<td>IDL widget button labels</td>
</tr>
<tr>
<td>XWD</td>
<td>X Windows Dump format</td>
</tr>
</tbody>
</table>

*Table 2-1: IDL-Supported Graphics Standards*
Chapter 2: Introducing IDL

Scientific Data Formats

IDL supports the HDF (Hierarchical Data Format), HDF-EOS (Hierarchical Data Format-Earth Observing System), CDF (Common Data Format), and NetCDF (Network Common Data Format) self-describing, scientific data formats. Collections of built-in routines provide an interface between IDL and these formats. For specific routine and object information used in IDL to access these type of files, see the “Scientific Data Formats” category under “Input/Output” (IDL Quick Reference).

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDF</td>
<td>Common Data Format version 3.1</td>
</tr>
<tr>
<td>HDF</td>
<td>Hierarchical Data Format version 4.1r5</td>
</tr>
<tr>
<td>HDF5</td>
<td>Hierarchical Data Format version 5-1.6.3</td>
</tr>
<tr>
<td>HDF-EOS</td>
<td>Hierarchical Data Format-Earth Observing System version 2.8</td>
</tr>
<tr>
<td>NCDF</td>
<td>Network Common Data Format version 3.5</td>
</tr>
</tbody>
</table>

Table 2-2: IDL-Supported Scientific Data Formats

Other Data Formats

For specific routine and object information used in IDL to access these data types, see the “Other Data Formats” category under “Input/Output” (IDL Quick Reference).

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCII</td>
<td>American Standard Code for Information Interchange</td>
</tr>
<tr>
<td>Binary</td>
<td>Digital data encoded as a sequence of bits</td>
</tr>
<tr>
<td>DXF</td>
<td>Drawing eXchange Format</td>
</tr>
<tr>
<td>ESRI Shapefile</td>
<td>Stores non-topological geometry and attribute information</td>
</tr>
<tr>
<td>SYLK</td>
<td>Symbolic Link Format</td>
</tr>
<tr>
<td>VRML</td>
<td>Virtual Reality Modeling Language</td>
</tr>
</tbody>
</table>

Table 2-3: Other IDL-Supported File Formats
### Chapter 2: Introducing IDL

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WAV</td>
<td>Microsoft Waveform Format</td>
</tr>
<tr>
<td>WAVE</td>
<td>Wavefront Advanced Data Visualizer Format</td>
</tr>
<tr>
<td>XDR</td>
<td>eXternal Data Representation</td>
</tr>
<tr>
<td>XML</td>
<td>eXtensible Markup Language</td>
</tr>
</tbody>
</table>

*Table 2-3: Other IDL-Supported File Formats (Continued)*
Launching IDL

To launch the IDL program, do one of the following:

**On Windows platforms** — Launching IDL means starting the IDL Development Environment application (no command-line mode is available under Windows). The IDL Development Environment is described in detail in Chapter 3, “The IDL Development Environment”. To start IDL, double-click on the IDL icon or select IDL from the Start menu.

**On UNIX platforms** — IDL offers two interfaces:

- In *command-line mode*, IDL uses a text-only interface and sends output to your terminal screen or shell window. (Graphics are displayed in IDL graphics windows.) To start IDL in command-line mode, enter `idl` at the shell prompt.

- In *graphical mode*, IDL displays the IDL Development Environment, an X-windows application that allows you to select options from menus, work with a built-in editor, and more. The IDL Development Environment is described in detail in Chapter 3, “The IDL Development Environment”. To start IDL in graphical mode, enter `idlde` at the shell prompt.

**On the Macintosh MacOS X platform** — IDL is launched in the same way as on UNIX platforms, except that you must explicitly open an X11 Terminal window.

- In *command-line mode*, IDL uses a text-only interface and sends output to your terminal screen or shell window. (Graphics are displayed in IDL graphics windows.) To start IDL in command-line mode, enter `idl` at the X11 Terminal window shell prompt.

- In *graphical mode*, IDL displays the IDL Development Environment, an X-windows application that allows you to select options from menus, work with a built-in editor, and more. The IDL Development Environment is described in detail in Chapter 3, “The IDL Development Environment”. To start IDL in graphical mode, double-click on the IDL icon or enter `idlde` at the X11 Terminal window shell prompt.

**Startup Options**

You can specify options to the command that starts IDL. On UNIX platforms, simply append the option flag after the `idl` or `idlde` command at the shell prompt. On Windows platforms, modify the **Target** field of the properties dialog for the IDL icon to include the option flag. See “Command Line Options for IDL Startup” on page 22 for a listing of the available startup options.
Troubleshooting

When IDL is ready to accept a command, it displays the IDL> prompt. If IDL does not start, take the following action depending upon the operating system you are running:

- **Windows:** Be sure that the path listed in the Properties dialog for the IDL icon accurately reflects the location of the IDL executable file `idlde.exe`.
- **UNIX:** Be sure that your `PATH` environment variable includes the directory that contains IDL.
Chapter 2: Introducing IDL

Launching the iTools

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. Based on the IDL Object Graphics system, the iTools are designed to help you get the most out of your data with minimal effort. They allow you to continue to benefit from the control of a programming language, while enjoying the convenience of a point-and-click environment. Each tool is designed around a specific visualization type:

- Two and three dimensional plots (line, scatter, polar, and histogram style)
- Surface representations
- Contour maps
- Image displays
- Volume visualizations
- Maps
- Vector displays

![Figure 2-1: Black Hole Density Data in the iVolume Tool](image-url)
Chapter 2: Introducing IDL

For detailed information on the new iTools and how to use them, see the * iTool User’s Guide.*

The iTools are built upon an object-oriented framework, or set of object classes, that serve as the building blocks for their interface and functionality. IDL programmers can easily use this framework to create custom data analysis and visualization environments. Such custom Intelligent Tools may be called from within a larger IDL application, or they may serve as the foundation for a complete application in themselves. For more information on creating your own custom iTools, see the * iTool Programming.*

Starting an iTool

To get started using the new IDL iTools, from the IDLDE command line, simply type the name of the tool you wish to call. The tools available are:

- *iContour*
- *iImage*
- *iPlot*
- *iSurface*
- *iVolume*
- *iMap*
- *iVector*

You can also launch an iTool using these other methods:

- From Windows:
  
  **Start → Programs → IDL 6.3 → iTools → iTool Name**

- From the IDLDE:

  **File → New → Visualization → iTool Name**

Loading Data into an iTool

There are multiple options for loading your data into your chosen iTool for visualization:

- **Command Line Argument** — At the IDL Command Line enter:

  ```idl
  mydata = RANDOMU(SEED, 45)
iPlot, mydata
  ```
Chapter 2: Introducing IDL

This option allows you to have control over parameters and keyword options for setting up the way you wish your plot (or other visualization) to appear.

- **File → Open** — The quickest way to create a default visualization of your data.
- **File → Import → IDL variable** — This will invoke the IDL Import wizard.
- **File → Import → From a File** — This also invokes the IDL Import wizard.
- **Insert → Visualization** — This method gives you parameter control similar to using the command line.

**Note**

For more detailed information on loading data into the iTools, see Chapter 2, “Importing and Exporting Data” (iTool User’s Guide).

---

The iTools Data Manager

All data used by any iTool is first loaded into the iTools Data Manager, which keeps track of which data items are associated with an iTool visualization. The Data Manager provides a convenient and structured environment in which to import and view files and variables.

The process of loading data into the Data Manager is entirely automatic if you specify data when launching an iTool at the IDL command line or if you open a data file using the **Open** command from the iTool’s **File** menu. In these cases, the iTool will import the data in the specified file or variable and create a visualization of the default type for the selected data and the iTool you are using.

If you want more control over the process of creating a visualization, you can load data into the Data Manager manually, either from a data file or from one or more variables that exist in your current IDL session. Once a data item is placed in the Data Manager, it is available to all iTools until it is removed.
Environment Variables Used by IDL

When IDL starts, it checks for the presence of a number of environment variables. If one of these environment variables exists, its value is used in one of two ways:

- As the value for a preference
- To configure IDL’s environment in such a way that it can load and run

Preferences

Preferences are internal values that control various aspects of the environment IDL presents to its users. While user preference values are most often retrieved from preference files, the value of any preference can be defined by setting an environment variable of the same name to the appropriate value. For example, to set the value of the IDL_PATH preference, which supplies the initial value of the !PATH system variable, you would define an environment variable named IDL_PATH.

If an environment variable corresponding to a preference exists, its value will be used as the value of that preference unless the value is explicitly overridden with a value set at the command line when invoking IDL. See Appendix E, “IDL Preferences” (IDL Reference Guide) for a detailed description of IDL’s preferences system and the precedence given to different sources for preference values.

Non-Preference Environment Variables

IDL checks the following environment variables at startup, but does not use the values as the values of IDL preferences.

CLASSPATH

The Java Connectivity bridges use the value of the CLASSPATH environment variable to locate IDL-supplied and user-defined Java classes.

DISPLAY

On UNIX platforms, IDL uses the DISPLAY environment variable to choose which X display is used to display graphics.

HOME

IDL uses the value of the HOME environment variable when storing user-specific information in the local file system.
Note
Under Microsoft Windows, the HOME environment variable might not be set in all cases. If it is not set, IDL first attempts to substitute the USERPROFILE environment variable (which usually looks something like C:\Documents and Settings\username where username is the login name of the current user). If USERPROFILE is not set, IDL uses the value of the first of the following it finds: the TEMP environment variable, the TMP environment variable, or the Windows system directory.

**IDL_BRIDGE_DEBUG**

The IDL export bridges check the value of the IDL_BRIDGE_DEBUG environment variable to enable or disable debugging support. Setting this environment variable to an appropriate value causes the Connectivity bridge to send debugging information that would typically appear in the IDL Output log to stdout. In addition, on Windows systems, debugging information is also written with the OutputDebugString() API, whose output can be captured by the Debug Monitor (DBMON.exe) tool (provided in the Platform SDK), Visual Studio, or the WinDbg debugger.

When IDL_BRIDGE_DEBUG is enabled, the following debug information is available:

- Library load errors (on Windows)
- IDL execution errors
- Output from the IDL print command

This table below shows the valid values for the IDL_BRIDGE_DEBUG environment variable and their meanings:

<table>
<thead>
<tr>
<th>Value</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Turn off debug output</td>
</tr>
<tr>
<td>1</td>
<td>Turn on debug output</td>
</tr>
</tbody>
</table>

All other values will be ignored. See either “Debugging” in Chapter 8 (COM) or “Debugging” in Chapter 9 (Java) in the IDL Connectivity Bridges manual for more information.
Chapter 2: Introducing IDL

Environment Variables Used by IDL

**IDLJAVAB_CONFIG**

The IDL-Java bridge uses the value of the IDLJAVAB_CONFIG environment variable to locate the IDL-Java bridge configuration file. See “Initializing the IDL-Java Bridge” (Chapter 5, IDL Connectivity Bridges) for additional details.

**IDLJAVAB_LIB_LOCATION**

The IDL-Java bridge uses the value of the IDLJAVAB_LIB_LOCATION environment variable to determine which JVM shared library within a given Java version to use. See “Initializing the IDL-Java Bridge” (Chapter 5, IDL Connectivity Bridges) for additional details.

**LD_LIBRARY_PATH**

On UNIX systems, the Java Connectivity bridges use the value of the LD_LIBRARY_PATH environment variable to determine where IDL’s shared library files are located.

**LM_LICENSE_FILE**

IDL’s FLEXlm-based license manager uses the value of the LM_LICENSE_FILE environment variable to determine where to search for valid license files. Consult the license manager documentation for details.

**PATH**

When IDL asks for an operating system resource such as a shell, the executable file for that resource must be located in the operating system’s path. While IDL itself does not use the value of the PATH environment variable explicitly, its value does affect IDL’s behavior when attempting to launch other applications.

**TERM**

On UNIX platforms, IDL uses the environment variable TERM to determine the type of terminal in use when IDL is in command-line mode.

**Setting Environment Variables**

The process used to set environment variables varies depending on the operating system you are using.
UNIX and MacOS X Systems

On UNIX systems, environment variables are generally specified in a file read by your shell program at startup. Syntax for setting environment variables varies depending on the shell you are using, as does the file you use to specify the variables. If you are unsure how to set environment variables on your system, consult the system documentation or a system administrator.

For example, to set the environment variable IDL_PATH to the value /usr/local/idl when using a C shell (csh), you would add the following line to your .cshrc file:

```
setenv LM_LICENSE_FILE /usr/local/idl/license/license.dat
```

Similarly, to set the same variable when using a Bourne shell (sh), you would add the following lines to your .profile file:

```
LM_LICENSE_FILE="/usr/local/idl/license/license.dat" \
; export LM_LICENSE_FILE
```

Microsoft Windows Systems

On Microsoft Windows systems, environment variables are set in the Environment Variables dialog, which is accessible from the System Control panel. Some Windows versions allow you to set environment variables either only for the user you logged in as (“user variables”) or for all users (“system variables”). Setting IDL environment variables as user variables means that other users who log on to the computer will not have access to your environment variable values.
# Command Line Options for IDL Startup

You can alter some IDL behaviors by supplying command-line switches along with the command used to invoke IDL. The following table shows the IDL command-line switches and the IDL interfaces to which they apply:

<table>
<thead>
<tr>
<th>Switch</th>
<th>UNIX idl</th>
<th>Windows idlde.exe</th>
<th>Windows idlde.exe</th>
</tr>
</thead>
<tbody>
<tr>
<td>-32</td>
<td>•</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-arg</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>-args</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>-autow</td>
<td></td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>-demo</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>-e</td>
<td></td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>-em</td>
<td></td>
<td></td>
<td>•</td>
</tr>
<tr>
<td>-novm</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>-nw</td>
<td></td>
<td></td>
<td>•</td>
</tr>
<tr>
<td>-pref</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>-queue</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>-quiet</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>-rt</td>
<td></td>
<td></td>
<td>•</td>
</tr>
<tr>
<td>-student</td>
<td></td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>-ulicense</td>
<td></td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>-vm</td>
<td></td>
<td></td>
<td>•</td>
</tr>
<tr>
<td>-w</td>
<td></td>
<td></td>
<td>•</td>
</tr>
</tbody>
</table>

Table 2-4: Command Line Switches
**Preference Switches**

In addition to the switches listed above, you can specify the value of IDL preferences when invoking IDL. See “Specifying Preferences at the Command Line” on page 28 for details.

**X Defaults**

In addition to the switches listed above, there are numerous command-line switches that control the appearance of the IDL Development Environment on UNIX systems. Those options are not listed here, and future versions of the UNIX Development Environment might not continue to support them. See “X Resources at the Command Line” in Chapter 6 for details.

**Batch Mode**

IDL can also be started in non-interactive “batch” mode by specifying the name of a batch file at startup time. See Chapter 3, “Executing Batch Jobs in IDL” (Application Programming) for details.

### Command-Line Switches

The following command line switches can be used when invoking IDL. Unless otherwise noted, switches can be combined and can be specified in any order.

**-32**

**Syntax:** `-32`

Starts IDL in 32-bit mode. If this switch is not set, IDL starts in 64-bit mode by default for those platforms that support 64-bit. If you have not installed the 64-bit version, the 32-bit version will automatically be started. If you have not installed the 32-bit version, this flag will not work.

This switch is only available on UNIX platforms.

**-arg**

**Syntax:** `-arg value`

Specifies a single command line option to be saved for later access via the `COMMAND_LINE_ARGS` function. The `value` string is saved. Multiple `-arg` switches are allowed; the values are saved in the order specified. The `-arg` option can be used to pass program-specific information from the command line to IDL programs.
-args

Syntax: -args value1 value2 ... valueN

Specifies one or more command line options to be saved for later access via the
COMMAND_LINE_ARGS function. When IDL sees the -args option, it takes any
command-line arguments that follow it and saves them all. There can only be one
-args option on an IDL command line, and it is always the final option. The -args
switch can be used with the -arg switch; if both switches are specified, occurrences
of -arg must come first, and the values specified by -args are saved following any
values specified by -arg.

-autow

Syntax: -autow

Starts IDL with the graphical user interface if possible. If for any reason IDL cannot
display the graphical user interface, it starts in command-line mode.

This switch is only available on UNIX platforms.

-demo

Syntax: -demo

Forces IDL to run in seven-minute demo mode.

-e

Syntax: -e IDL_statement

Specifies a single IDL statement to be executed. Once the statement has executed,
IDL waits for any widget applications to exit, and then IDL itself exits. Only the last
-e switch on the command line is honored.

Note

If the IDL statement includes spaces, it must be enclosed in quote marks. Under
UNIX the statement can be enclosed in either single or double quotes, but under
Microsoft Windows the statement must be enclosed in double quotes.

Under UNIX, the -e switch always uses the command line interface (that is, the
idlde command followed by the -e switch behaves like the idl command followed
by the -e switch).

Under Microsoft Windows, the idlde command displays the full development
environment, but the user is not prompted for IDL commands to execute. This mode
is primarily useful because the output log window is visible, and will show any output generated by the IDL statement. The \texttt{idlrt} command also supports the \texttt{-e} option, and in this mode requires a standard IDL license. Since \texttt{idlrt} does not display the output generated by IDL statements, it is primarily of use for widget based applications that provide a graphical user interface to their functionality.

\textbf{Note}

Because the \texttt{-e} switch causes IDL to exit as soon as the statement is complete, if the IDL statement being executed produces graphics, you may wish to delay the exit until the user has a chance to view the graphics. In such a case, you must explicitly cause IDL to wait before exiting. For example, the following will produce a plot of one cycle of a sinusoid:

\begin{verbatim}
idlde -e "PLOT, SIN(FINDGEN(628)/100) & t=DIALOG_MESSAGE('Done')"
\end{verbatim}

The plot will remain on the screen until the user dismisses the dialog, at which point IDL will exit.

\textbf{-em}

Syntax: \texttt{-em=file}

Starts IDL with an embedded license. The \textit{file} argument should be an IDL \texttt{.sav} file that contains an embedded (“unlimited right to distribute”) IDL license. See Chapter 24, “Distributing Runtime Mode Applications” (Application Programming) for details on creating applications with an embedded IDL license.

This switch is accepted on UNIX platforms and by the \texttt{idlrt.exe} application on Microsoft Windows platforms.

\textbf{-novm}

Syntax: \texttt{-novm}

Forces IDL to run in seven-minute demo mode rather than Virtual Machine mode if no license is available. This switch can only be used in conjunction with the \texttt{-rt} switch or the \texttt{idlrt.exe} executable.

If IDL attempts to load and run an IDL application in runtime mode, but finds no license available, it will load the application in Virtual Machine mode by default. Setting the \texttt{-novm} switch prevents the application from running in Virtual Machine mode, and instead causes it to run in demo mode.
Chapter 2: Introducing IDL

Command Line Options for IDL Startup

IDL Interface

- **nw**
  
  Syntax: `-nw`
  
  Starts IDL in command-line mode no matter what. Note that specifying `idlde -nw` at the shell prompt will start IDL in command-line mode.
  
  This switch is only available on UNIX platforms.

- **pref**
  
  Syntax: `-pref=file`
  
  Loads the specified preference file. The `file` argument should be a text file containing IDL preference/value pairs. See Appendix E, “IDL Preferences” (IDL Reference Guide) for a detailed description of IDL’s preferences system, the format of preference files, and the precedence given to different sources for preference values.
  
  This feature is of particular interest to those writing stand-alone applications in IDL, possibly using the runtime or Virtual Machine modes of operation. The use of a command-line preference file allows authors of such applications to control the values of preferences important to their applications in a way that is user-adjustable and not hardwired into the code of their application.

- **queue**
  
  Syntax: `-queue`
  
  Causes IDL to wait for a license to become available before beginning an IDL task such as batch processing. This switch is useful for users of counted floating licenses who need their IDL process to run in licensed mode rather than in seven-minute demo mode.

- **quiet**
  
  Syntax: `-quiet`
  
  Suppresses printing of the IDL announcement and the `motd.txt` file. See “Message of the Day Files” on page 31 for details on message-of-the-day files.

- **rt**
  
  Syntax: `-rt=file`
  
  Starts IDL with a runtime license. If the `file` argument is specified, it should be an IDL `.sav` file. If the `file` argument is not specified, IDL attempts to run a file named
Chapter 2: Introducing IDL

IDL Interface

Command Line Options for IDL Startup

runtime.sav. See Chapter 24, “Distributing Runtime Mode Applications” (Application Programming) for details on creating runtime applications.

This switch is accepted on UNIX platforms and by the idlrt.exe application on Microsoft Windows platforms. It is, however, redundant when using the idlrt.exe application.

-**student**

Syntax: -student

Forces IDL to start in student mode. This switch is useful for testing IDL applications that should run in student mode.

-**ulicense**

Syntax: -ulicense

Check out a unique license even if IDL is already running on the same display. If IDL has checked out a unique license using this flag, the user is allowed to change the DISPLAY environment variable after IDL has started.

-**vm**

Syntax: -vm=file

Starts the IDL Virtual Machine. If the file argument is specified, it should be an IDL .sav file. If the file argument is not specified, IDL displays a file selection dialog. See Chapter 25, “Distributing Virtual Machine Applications” (Application Programming) for details on creating applications that run in the IDL Virtual Machine.

This switch is accepted on UNIX platforms and by the idlrt.exe application on Microsoft Windows platforms.

-**w**

Syntax: -w

Starts IDL with the graphical user interface. This is the same as entering idlde at the command prompt.

This switch is only available on UNIX platforms.
Specifying Preferences at the Command Line

In addition to the command line switches described above, the value of any IDL preference can be specified at the command line using the following syntax:

```
idlcommand -PREFERENCE value
```

where `idlcommand` is the command used to launch IDL (one of `idl`, `idlde`, or `idlrt`), `PREFERENCE` is the name of an IDL preference (note the leading hyphen), and `value` is the value for the preference. For example, to set the value of the IDL_MORE preference to one when launching IDL in command-line mode on a UNIX machine, you would use the following command line:

```
idl -IDL_MORE 1
```

Any number of preference values can be specified the command line. See Appendix E, “IDL Preferences” (IDL Reference Guide) for a detailed description of IDL’s preferences system and the precedence given to different sources for preference values.

Using Switches Under Windows

Under Microsoft Windows, applications can be launched either from the prompt in a Command Window or by double-clicking on the application icon. If you launch IDL from a command prompt, simply specify the switch on after the name of the IDL executable you are using. For example, to start IDL in Virtual Machine mode using the `-vm` switch, you would use the following command line:

```
C:\IDL_DIR\bin\bin.platform\idlrt.exe -vm=file.sav
```

where `IDL_DIR` is the directory where you have installed IDL, `platform` is the platform-specific `bin` directory, and `file.sav` is the name of the SAVE file you wish to restore and run.

If you launch IDL by double-clicking on the application icon, set switches by modifying the `target` specified in the application’s shortcut properties to include the switch.
Startup Files

A startup file is a batch file that is executed automatically each time the IDL is started. The name of the startup file is specified by the IDL_STARTUP preference. (See Appendix E, “IDL Preferences” (IDL Reference Guide) for information on IDL’s preferences system.)

Common uses for startup files include the following:

- Restoring variable data contained in a .sav file or reading in commonly used data
- Setting common keywords to the DEVICE procedure
- Specifying shared or private color maps for PseudoColor devices

Startup files are executed one statement at a time. It is not possible to define program modules (procedures, functions, or main-level programs) in the startup file. For more information on creating batch files, see Chapter 3, “Executing Batch Jobs in IDL” (Application Programming).

Understanding When Startup Files are Not Executed

Startup files are executed only when a command line is present. (Prior to IDL 6.2, IDL would execute the startup file specified by the environment variable IDL_STARTUP even if no IDL command line was present.) Now, however, the startup file is not executed when running the following types of applications:

- IDL Virtual Machine applications
- Runtime applications
- Applications using an IDL Remote Procedure Call server
- ION applications
- Callable IDL applications
- COM and Java Connectivity Bridge applications
- Applications that use the IDL_IDLBridge object

In most cases the new behavior is desirable; you as the developer of an IDL application do not necessarily know whether the end user of your application has an IDL startup file.
If, however, you are creating an application that relies on settings defined in a startup file, you have the following options:

- Use the IDL preferences system, rather than a startup file, to set the appropriate values. This option is only available if the values being defined in the startup file correspond to IDL preferences.
- If your application is not a Virtual Machine or Runtime IDL application, explicitly execute the startup file after your IDL session has begun. See the following section for details.

**Manually Executing a Startup File**

To explicitly execute the startup file after an IDL process has been started (either through the IDL_IDLBridge object or a Connectivity Bridge wrapper object), complete the following steps:

1. Create a string variable containing the “@” character concatenated with the name of the startup file:

   \[
   \text{startup\_file} = '@' + \text{PREF\_GET('IDL\_STARTUP')}
   \]

2. Using the appropriate “execute string” functionality for your application to execute the string variable.
Chapter 2: Introducing IDL

Message of the Day Files

When IDL starts, it displays the contents of the \texttt{motd.txt} file, located in the \texttt{help/motd} subdirectory of the IDL distribution, in the Output Log. You can use this \textit{Message of the Day} file to provide information to IDL users every time IDL starts.

In addition, IDL will display the contents a file with the name \texttt{platform.txt} located in the \texttt{help/motd} subdirectory of the IDL distribution, where \texttt{platform} is a string corresponding to the current operating system platform. For example, on Linux systems, IDL displays a file named \texttt{linux.txt}.

You can determine the correct name for this file on a given platform by using the following IDL command:

\begin{verbatim}
PRINT, !VERSION.OS
\end{verbatim}

and appending the “.txt” extension to the operating system name.

If you do not want to see either the \texttt{motd.txt} file or the platform-specific file each time IDL starts, remove them from the \texttt{help/motd} subdirectory of the IDL distribution.

\textbf{Note}

The \texttt{motd.txt} and platform-specific files are simply an ASCII text files—not IDL programs or batch files. To execute a series of IDL commands, select a startup file as described in “Startup Files” on page 29.
Using Your Mouse with IDL

IDL supports the use of mice with up to three buttons. Because some systems use mice with one or two buttons, IDL provides mechanisms for simulating a three-button mouse.

Using a Two-Button Mouse

Many mice used with Microsoft Windows systems have only two buttons. See your system documentation for information on emulating a middle-button press.

Using a Macintosh (One-Button) Mouse

Many mice used with Macintosh systems have only one button. The X Window System software provided with MacOS X provides multi-button mouse emulation, allowing you to configure the system to generate middle- and right-button press events. See your MacOS X system documentation for details.
Using Keyboard Accelerators

IDL supports the use of keyboard accelerators or shortcuts in three different contexts: in the IDL Development Environment (menu actions), in the IDLDE Editor window, and in IDL widget applications. For information on development environment keyboard shortcuts, see one of the following:

- Chapter 3, “The IDL Development Environment” provides descriptions of each available menu item including keyboard shortcuts
- “Editor Window Keyboard Shortcuts” (Chapter 2, Application Programming) describes keyboard shortcuts specifically designed for use in the Editor window

Keyboard shortcuts can also be defined for individual buttons and menu items in an IDL widget application. Defining shortcut key combinations is the responsibility of the IDL programmer who creates the widget application; if you are using a widget application and are unsure about whether keyboard shortcuts have been defined, contact the author of the widget application. For information on adding keyboard accelerators to your own widget applications, see “Enhancing Widget Application Usability” (Chapter 4, Widget Application Programming).

Enabling Alt Key Accelerators on Macintosh

If you are using IDL on a Macintosh and wish to use keyboard accelerators that use the Alt key, you will need to perform the following steps to make the Apple (Command) key to function as the Alt key:

1. Create a .Xmodmap file in your home folder and add the following three lines to it:
   
clear mod1
   clear mod2
   add mod1 = Meta_L

   When Apple’s X11 program starts, this file will automatically be read, and the Apple key will be mapped to the left meta key ⌇, which for IDL’s purposes is the Alt key. (Windows Alt key accelerators are mapped to the Macintosh Apple key, not the Option (alt) key.)

2. Run Apple’s X11 program and change its preferences. Under Input in the X11 Preferences dialog, make sure that the following two items are unchecked:
   - Follow system keyboard layout
   - Enable key equivalents under X11
Chapter 2: Introducing IDL

Using Keyboard Accelerators

**Note**

You must relaunch Apple’s X11 program for these changes to take effect.

Once you have performed these steps, keyboard shortcuts will operate in the normal Macintosh fashion — namely, pressing the **Apple** key in conjunction with X, C, and V will perform cut, copy and paste. The IDLDE’s other shortcuts and any widget accelerators defined to use the **Alt** key will also work.
Getting Help with IDL

IDL’s online help system provides access to information on all aspects of IDL. The complete IDL documentation set is available online in HTML format. To use the IDL online help system, do one of the following:

- Enter the `?` command (optionally followed by a routine or object name) at the IDL command prompt
- Call the `ONLINE_HELP` procedure at the IDL command prompt or within an IDL program
- If you are running the IDL Development Environment (IDLDE), select the `Help` option from the menu bar
- Select `IDL Help` from the Microsoft Windows Start menu
- Double-click on the `IDLHelp` Macintosh icon

In addition to the online help format, IDL documentation is available in a set of Adobe Acrobat PDF files located on the IDL CD-ROM. See “Using the PDF Documentation Set” on page 43 for details.

Using the IDL Online Help Viewer

IDL’s online help system uses a cross-platform help viewer — `IDL Assistant` — based on the help viewer used by the Qt development toolkit from Trolltech. This section describes how to use the IDL Assistant application. For information on creating help content that uses the IDL Assistant for your own IDL applications, see Chapter 23, “Providing Online Help For Your Application” (Application Programming).

The Main Window

The IDL Assistant main window contains the text of the current topic. Within the main window you can:

- Follow hypertext links to other topics, or to sections within the current topic
- Navigate to the next or preceding topic using arrows at the top of the topic screen
- Display multiple topics simultaneously using the tabbed interface
- Create new tabs and close existing tabs using icons to the right and left of the tabs
Chapter 2: Introducing IDL

Getting Help with IDL

- Perform common tasks including display of the next/previous topic, tab management, text sizing, copying text to the clipboard, and finding text within the topic using the context menu

The Sidebar

The IDL Assistant sidebar provides four tabs that allow you to navigate through the IDL documentation set. All of the tabs provide a context menu that allows you to open the selected topic the current tab, a new tab, or a new window.

The Contents Tab

The Contents tab displays a hierarchical listing of the contents of the various books in the IDL documentation set.

The Index Tab

The Index tab provides a keyword index of the contents of the IDL documentation set. Enter a text string in the Look For: field to see keywords that match the string.

The Search Tab

The Search tab allows you to search the text of the IDL documentation set for words or phrases. Text matching your search string is highlighted when a topic is displayed in the main window.

Tip

Words or phrases entered in the Search tab are not case sensitive.

To search for words, enter one or more strings in the Searching for: field, separated by spaces and click Search. IDL Assistant displays a list of topics that contain all of the words you entered.

To search for a phrase, enclose the phrase in single or double quote marks.

Warning

The IDL documentation set is quite large. The results of a full-text search query may take several moments to appear in the Search tab.

The list of topics containing the search words or phrase is displayed as a list ranked roughly according to the number of occurrences of the words or phrases, with the topics containing the largest number of occurrences listed given higher rankings.

Each entry in the list of topics is followed by an abbreviation of the title of the manual in which the topic appears. See “Book Name Abbreviations” on page 38 for the list of abbreviations.

Getting Help with IDL
Chapter 2: Introducing IDL

Allowed Characters

The following characters are allowed in the Search tab:

- Letters (upper- and lower-case)
- Numbers (0–9)
- Quote marks (single (‘), double (“), backwards (’)
- Exclamation marks (!), colons (:), and periods (.)
- Spaces
- Hyphens (−)
- Underscores (_)
- Asterisk (*) as a wildcard matching one or more unspecified characters

Note: The * character cannot be used within quotes or at the beginning of a string.

All other characters are disallowed; you cannot enter them in the Searching for: field.

Warning: Searches that contain single-character strings (such as “a” or “8”) are not allowed and will return no results. This is true even when the single character is combined with a punctuation character such as a hyphen. For example, searching for the string “8-bit” will return no results.

Examples

- `convol` List all topics that contain the word “convol”
- `convol*` List all topics that contain a word beginning with “convol”
- `base widget` List all topics that contain the word “base” and the word “widget”
- `"base widget"` List all topics that contain the phrase “base widget”
Book Name Abbreviations

The following abbreviations of book titles are used in the list of topics returned by the search:

- bld Building IDL Applications
- cbr IDL Connectivity Bridges
- dm DataMiner Guide
- edg External Development Guide
- gs Getting Started with IDL
- img Image Processing Guide
- inst Installing and Licensing IDL
- ionj ION Java User’s Guide
- ions ION Script User’s Guide
- itd iTool Developer’s Guide
- itu iTool User’s Guide
- med Medical Imaging in IDL
- obj Object Programming
- obs Obsolete Features
- ref IDL Reference Guide
- sdf Scientific Data Formats
- use Using IDL
- wav Wavelet Toolkit User’s Guide
- wn What’s New in IDL

The Bookmarks Tab

The Bookmarks tab allows you to save links to specific topics in the IDL documentation set for easy reference.
The Menu Bar

The IDL Assistant menu bar runs across the top of the IDL Assistant window, and provides access to the features listed below. Keyboard shortcuts to invoke various menu items are listed in the menus themselves.

<table>
<thead>
<tr>
<th>Menu</th>
<th>Item</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>File</td>
<td>New Window</td>
<td>Open a new IDL Assistant window.</td>
</tr>
<tr>
<td></td>
<td>Add Tab</td>
<td>Open a new tab displaying the same topic as the currently selected tab.</td>
</tr>
<tr>
<td></td>
<td>Close Tab</td>
<td>Close the currently selected tab.</td>
</tr>
<tr>
<td></td>
<td>Print</td>
<td>Print the contents of the currently selected tab. See “Printing” on page 42 for details.</td>
</tr>
<tr>
<td></td>
<td>Close</td>
<td>Close the current IDL Assistant window.</td>
</tr>
<tr>
<td></td>
<td>Exit</td>
<td>Close all IDL Assistant windows.</td>
</tr>
<tr>
<td>Edit</td>
<td>Copy</td>
<td>Copy text selected in the main window to the system clipboard.</td>
</tr>
<tr>
<td></td>
<td>Find in Text...</td>
<td>Search for a text string in the currently displayed topic.</td>
</tr>
<tr>
<td></td>
<td>Find Next</td>
<td>Find the next instance of the text string in the currently displayed topic.</td>
</tr>
<tr>
<td></td>
<td>Find Previous</td>
<td>Find the previous instance of the text string in the currently displayed topic.</td>
</tr>
<tr>
<td></td>
<td>Settings...</td>
<td>Display the Settings dialog. See “Settings” on page 42 for details.</td>
</tr>
</tbody>
</table>

Table 2-5: IDL Assistant Menus
### Chapter 2: Introducing IDL

#### Getting Help with IDL

<table>
<thead>
<tr>
<th>Menu</th>
<th>Item</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>View</strong></td>
<td><strong>Zoom in</strong></td>
<td>Increase the text size in the main window. See “Text Zoom” on page 41 for important notes.</td>
</tr>
<tr>
<td></td>
<td><strong>Zoom out</strong></td>
<td>Decrease the text size in the main window. See “Text Zoom” on page 41 for important notes.</td>
</tr>
<tr>
<td></td>
<td><strong>Views...</strong></td>
<td>Control display of the Sidebar and Standard toolbar.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note</strong> - The Line Up feature realigns the toolbar if it has been moved.</td>
</tr>
<tr>
<td><strong>Go</strong></td>
<td><strong>Previous</strong></td>
<td>Display the current tab’s previous topic.</td>
</tr>
<tr>
<td></td>
<td><strong>Next</strong></td>
<td>Display the current tab’s next topic.</td>
</tr>
<tr>
<td></td>
<td><strong>Home</strong></td>
<td>Display the IDL online help Home page.</td>
</tr>
<tr>
<td></td>
<td><strong>Next Tab</strong></td>
<td>Select the tab to the right of the current tab, if any.</td>
</tr>
<tr>
<td></td>
<td><strong>Previous Tab</strong></td>
<td>Select the tab to the left of the current tab, if any.</td>
</tr>
<tr>
<td><strong>Bookmark</strong></td>
<td><strong>Add Bookmark</strong></td>
<td>Create a bookmark for the currently selected topic.</td>
</tr>
<tr>
<td></td>
<td><strong>Bookmark list</strong></td>
<td>Existing bookmarks are displayed at the bottom of this menu.</td>
</tr>
<tr>
<td><strong>Help</strong></td>
<td><strong>IDL Assistant Manual</strong></td>
<td>Display this help topic.</td>
</tr>
<tr>
<td></td>
<td><strong>About IDL Assistant</strong></td>
<td>Display information about IDL Assistant.</td>
</tr>
<tr>
<td></td>
<td><strong>What’s This?</strong></td>
<td>Display context-sensitive pop-up help about some portions of the IDL Assistant interface.</td>
</tr>
</tbody>
</table>

*Table 2-5: IDL Assistant Menus*
Chapter 2: Introducing IDL

The Tool Bar

The IDL Assistant tool bar provides quick access to a subset of the features available via the menubar.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔄</td>
<td>Previous</td>
<td>Display the current tab’s previous topic.</td>
</tr>
<tr>
<td>🔄</td>
<td>Next</td>
<td>Display the current tab’s next topic.</td>
</tr>
<tr>
<td>🏛️</td>
<td>Home</td>
<td>Display the IDL online help Home page.</td>
</tr>
<tr>
<td>📜</td>
<td>Copy</td>
<td>Copy text selected in the main window to the system clipboard.</td>
</tr>
<tr>
<td>🕵️</td>
<td>Find in Text</td>
<td>Search for a text string in the currently displayed topic.</td>
</tr>
<tr>
<td>🗒️</td>
<td>Print</td>
<td>Print the contents of the currently selected tab. See “Printing” on page 42 for details.</td>
</tr>
<tr>
<td>📡</td>
<td>Zoom in</td>
<td>Increase the text size in the main window. See “Text Zoom” on page 41 for important notes.</td>
</tr>
<tr>
<td>🔎</td>
<td>Zoom out</td>
<td>Decrease the text size in the main window. See “Text Zoom” on page 41 for important notes.</td>
</tr>
<tr>
<td>🤔</td>
<td>What’s this?</td>
<td>Display context-sensitive pop-up help about some portions of the IDL Assistant interface.</td>
</tr>
</tbody>
</table>

Table 2-6: IDL Assistant Toolbar

Text Zoom

Select **Zoom in** or **Zoom out** from the View menu to change the size of the text in the IDL Assistant main window.

The smoothness of the text zoom operation depends on the ability of the operating system to provide fonts of the appropriate size for the zoomed text. On platforms that provide robust font-management mechanisms, the **Zoom** operations will work smoothly. On platforms that provide more limited font support, a single **Zoom** operation may, depending on the current text size and font support, change the text size for only some text elements in the main window, or none at all. In these cases, repeated applications of the **Zoom** operations may change the text size.
If you find that the text zooming feature does not work adequately with the default fonts, try changing the fonts used by IDL Assistant (see “Settings” on page 42 for details.) On platforms that use a set of fixed-size fonts, choosing a font with a larger number of available sizes will allow smoother text zooming.

Printing

Select Print from the File menu or toolbar to display a platform-native Print dialog that allows you to select a printer on which to print.

Note
Currently, the only text range option available is All. Printing all will print the entire contents of the topic currently displayed in the main window.

Tip
The quality of the printed output from IDL Assistant depends on the platform and printer in use. For high-quality printed output, consider printing from the PDF version of the document you are viewing. See “Using the PDF Documentation Set” on page 43 for details.

Settings

Select Settings from the Edit menu to display a tabbed dialog that allows you to control several IDL Assistant settings.

General Tab

The General tab allows you to select fonts for text display in the main window. By default, the Font is set to Helvetica, and the Fixed Font is set to Courier.

Tip
Depending on the configuration of your system, you may be able to select alternate fonts that provide better appearance or smoother zooming behavior than the defaults. This is especially true on UNIX systems that have a limited set of fonts available. Trying different font settings may improve both the legibility of the text and the ability to zoom in the IDL Assistant viewer.

The General tab also allows you to select a color for hyperlinks and specify whether the links should be underlined. Depending on your platform, changing these values may not produce the effect you expect.
Chapter 2: Introducing IDL

Web Tab

The Web tab allows you to define the web browser that should be invoked when you click on a hyperlink that refers to a web site rather than to a file in the IDL documentation set.

The Web tab also allows you to specify an HTML file that should be displayed when you select Home from the Go menu or click the Home toolbar icon. By default, the home page is defined as

<IDL_DIR>/help/online_help/home.html

where <IDL_DIR> is the full path to your IDL installation.

PDF Tab

The PDF tab allows you to define a Portable Document Format (Adobe Acrobat) file browser that should be invoked when you click on a hyperlink that refers to a PDF file.

Note

On the Macintosh, if you choose to define your PDF file browser as Adobe Acrobat, you must use version 7 or later. Under Windows, you should use version 6 or later. See the readme.txt file in the info/docs directory of your IDL CD-ROM for additional information.

Using the PDF Documentation Set

The complete IDL documentation set is available in a set of Adobe Portable Document Format (PDF) files. The PDF documentation set is hyperlinked, provides navigational bookmarks in the bookmarks pane, and provides a compiled full-text search index.

Adobe Systems Inc. created the Portable Document Format in the early 1990s, basing it on their PostScript language. PDF is intended to allow documents to be displayed in exactly the same manner on a wide variety of computing platforms.

The IDL PDF files are electronic representations of the individual books in the documentation set, and can be either viewed on screen or printed (in full or in part) on a local printer. When viewed on-screen, the PDF books provide hyperlinked cross-references, tables of contents, and indices, allowing for speedy navigation through the set. In addition, some versions of the Adobe Acrobat software provide a fast full-text search capability, using a pre-compiled full-text index of the entire document set.
Viewing PDF Files

Viewing PDF files requires a separate application, not included in the IDL installation. Various PDF viewing applications are in wide use, and one or more may already be installed on your system.

The PDF version of the IDL documentation set is designed to be viewed using Adobe Acrobat or Adobe Reader. Other third-party PDF viewers (notably GhostScript and Apple’s Preview) are available, but these viewers may not support all of the features available when viewing the IDL PDF files in Adobe Acrobat. Inter-document hyperlinks may or may not work correctly when using other viewers, and the compiled full-text search index (the Acrobat “Search” feature) will almost surely not work correctly in other viewers.

The Adobe Reader software is available at no charge directly from Adobe:

www.adobe.com/reader

Locating the PDF Documentation Set

The PDF version of the documentation set is not installed with IDL. The PDF files are located in the info/docs subdirectory of the IDL installation CD-ROM. See the readme.txt file in that directory for information on installing the PDF files.
Chapter 2: Introducing IDL

Typographical Conventions

The following typographical conventions are used throughout the IDL documentation set:

- **UPPER CASE type**
  IDL functions and procedures, and their keywords are displayed in UPPER CASE type. For example, the calling sequence for an IDL procedure looks like this:
  
  \[ \text{CONTOUR, } Z \ [\, X, Y] \]

- **Mixed Case type**
  IDL object class and method names are displayed in Mixed Case type. For example, the calling sequence to create an object and call a method looks like this:
  
  \[ \text{object = OBJ_NEW('IDLgrPlot')} \]
  \[ \text{object -> GetProperty, ALL=properties} \]

- **Italic type**
  Arguments to IDL procedures and functions — data or variables you must provide — are displayed in italic type. In the above example, \( Z, X, \) and \( Y \) are all arguments.

- **Square brackets ([ ])**
  Square brackets used in calling sequences indicate that the enclosed arguments are optional. Do not type the brackets. In the above CONTOUR example, \( X \) and \( Y \) are optional arguments. Square brackets are also used to specify array elements.

- **Courier type**
  In examples or program listings, things that you must enter at the command line or in a file are displayed in courier type. Results or data that IDL displays on your computer screen are also shown in courier type. An example might direct you to enter the following at the IDL command prompt:
  
  \[ \text{array = INDGEN(5)} \]
  \[ \text{PRINT, array} \]

  In this case, the results are shown like this:

  \[ 0 \ 1 \ 2 \ 3 \ 4 \]
Quitting IDL

To quit IDL, do one of the following:

- Enter the \texttt{EXIT} command at the IDL command prompt.
- If you are running the IDL Development Environment (IDLDE), select the \texttt{Exit} option from the \texttt{File} menu.
- Under Microsoft Windows, press \texttt{Alt+F4}.
- Under UNIX or MacOS X, if you use IDL’s command-line mode, press \texttt{Ctrl+D} as the first character in command-line mode causes IDL to exit back to the operating system. The \texttt{EXIT} procedure has the same function. If \texttt{Ctrl+D} is not the first character, it simply ends the input line as if a return had been entered.

\textbf{Note} \\
When using IDL’s command-line mode under UNIX or MacOS X, you can normally press \texttt{Ctrl+Z} to suspend IDL and return you to the shell process without exiting IDL. After completing any shell commands, type \texttt{fg} to return IDL to the foreground. Although the UNIX suspend character can be changed by the user outside of IDL, this is rarely done. For the purposes of this manual, we assume the default convention.
Chapter 2: Introducing IDL

Reporting Problems

We strive to make IDL as reliable and bug free as possible. However, no program with the size and complexity of IDL is perfect, and problems do surface. When you encounter a problem with IDL, the manner in which you report it has a large bearing on how well and quickly we can fix it.

The relnotes.txt file accompanying each release includes information about new features in that release, bug fixes, and known problems which may be of help.

This section is intended to help you report problems in a way which helps us to address the problem rapidly.

Background Information

Sometimes, a problem only occurs when running on a certain machine, operating system, or graphics device. For these reasons, we need to know the following facts when you report a problem:

- Your IDL installation number.
- The version of IDL you are running.
- The type of machine on which it is running.
- The operating system version it is running under.
- The type and version of your windowing system if you are on UNIX.
- The graphics device, if the problem involves graphics and you know what graphics device is on your system.

The installation number is assigned by us when you purchase IDL and is included in the license information that we sent you. The IDL version, site number, and type of machine are printed when IDL is started.

For example, the following startup announcement appears indicating you are running IDL version 6.4 under Sun Solaris using installation number xxxxx-x, under a floating license located on a particular license manager.

IDL Version 6.4, Solaris (sunos sparc m64).
(c) 2004, ITT Visual Information Solutions
Installation number: xxxxx-x.
Licensed for use by: ITT Visual Information Solutions IDL floating licenses
Under UNIX, the version of the operating system can usually be found in the file 
/etc/motd. It is also printed when the machine boots. In any event, your system 
administrator should know this information.

Under Windows, select **About** from the **Help** menu in the Windows Explorer.

**Double Check**

Before reporting a problem double check with the manual or a local expert if one is 
available. Sometimes, it is a simple matter of misinterpreting what is supposed to 
happen.

If you cannot determine what should happen in a given situation by consulting the 
reference manual, the manual needs to be improved on that topic. Please let us know 
if you feel that the manual was vague or unclear on a subject.

Another question to ask is whether the problem lies within IDL, or with the system 
running IDL. Is your system properly configured with enough virtual memory and 
sufficient operating system quotas? Does the system seem stable and is everything 
else working normally?

**Describing The Problem**

When describing the problem, it is important to use precise language. Terms like 
crashes, blows up, and fails are vague and open to interpretation. Does it really crash 
IDL and leave you looking at an operating system prompt? This is how ITT Visual 
Information Solutions technical support personnel interpret a problem report of a 
crash. If the behavior being reported refers to an unexpected error message being 
issued before returning another prompt, then describing it as a crash becomes 
misleading. What is really meant by a term like “fails?”

It is also important to separate concrete facts from conjecture about underlying 
causes. For example, a statement such as "IDL dumps core when allocating dynamic 
memory" is not nearly as useful as this statement, "IDL dumps core when I execute 
the following statements... "

**Reproducibility**

Intermittent problems are by far the hardest kind to fix. In general, if we can't make it 
happen on our machine, we can't fix it. It is far more likely that we can help you if 
you can tell us a sequence of IDL statements that cause the problem to happen. 
Naturally, there are degrees of reproducibility. Situations where a certain sequence of 
statements causes the problem 1 time in 3 tries are fairly likely to be fixable. 
Situations where the problem happens once every few months and no one is sure 
what triggered it are nearly impossible to identify and correct.
Simplify the Problem

In accordance with ITT Visual Information Solutions Technical Support policy, when reporting a problem, it is important to give us the shortest possible series of IDL statements that cause it. Here are some suggestions for simplifying your problem:

Copy the procedure and function files that are involved to a scratch second copy. Never modify your only copy!

Eliminate everything not involved in demonstrating the problem. Don't do this all at once. Instead, do it in a series of slow careful steps. Between each step, stop and run IDL on the result to ensure that the problem still appears.

If a simplification causes the problem to disappear, then slowly restore the statements involved until you can identify the source of the problem. The end result of such simplification should be a small number of IDL statements that demonstrate the problem.

If the problem does not involve file Input/Output, strive to eliminate all file I/O statements. Use IDL routines to generate a dummy data set, rather than including your own data if at all possible. If your problem report does not involve your data, it will be much easier for us to reproduce.

On the other hand, if the problem involves file Input/Output, and the problem only happens with a certain data file or type of data, we will need to look at your data or a sample of your data.

If it is necessary to send us your data, use one of the following methods:

- If the data set is small, please send it as an attachment in your email to us: support@ittvis.com.
- If the data set is large, please place it on our ftp site at: ftp.ittvis.com/incoming.

Be sure to include the commands that reproduce your problem in your message to use. If you have placed your data on the ftp site, include the name of the data set and when it was uploaded.

Problems with Dynamic Loading

Under some operating systems, the CALL_EXTERNAL and LINKIMAGE system routines allow you to dynamically load routines written in other languages into IDL. This is a very powerful technique for extending IDL, but it is considerably more difficult than simply writing IDL statements. At this level, the programmer is outside the user level shell of IDL and is not protected from programming errors. These errors could give incorrect results or crash IDL. In such situations, the burden of
proving that a problem is within IDL and not the dynamically loaded code is entirely the programmer's.

Although it is certainly true that a problem in this situation can be within IDL, it is very important that you exhaust all other possibilities before reporting the problem. If you decide that you need to report the problem, the comments above on simplifying things are even more important than usual. If you send us a small example that exhibits the problem, we may be able to respond with a correction or advice.

**Contact Us**

To report a problem, contact us at the following addresses:

Electronic Mail
support@ittvis.com

Telephone
(303) 786-9900
(303) 786-9909 (Fax)
(303) 413-3920 (IDL technical support direct line)

Mail
ITT Visual Information Solutions
4990 Pearl East Circle
Boulder, CO 80301

Web Site
http://www.ittvis.com
Chapter 3
The IDL Development Environment

This chapter describes the IDL Development Environment.

Components of the IDLDE . . . . . . . . . . . . . 52
File Menu . . . . . . . . . . . . . . . . . . . . . . . . . . . 59
Edit Menu . . . . . . . . . . . . . . . . . . . . . . . . . . . 63
Search Menu . . . . . . . . . . . . . . . . . . . . . . . . . 65
Run Menu . . . . . . . . . . . . . . . . . . . . . . . . . . . 67
Project Menu . . . . . . . . . . . . . . . . . . . . . . . . . 72
Macros Menu . . . . . . . . . . . . . . . . . . . . . . . . . 73
Window Menu . . . . . . . . . . . . . . . . . . . . . . . . . 75
Help Menu . . . . . . . . . . . . . . . . . . . . . . . . . . . 78
Printing in IDL . . . . . . . . . . . . . . . . . . . . . . . . 79
IDL Printer Setup in UNIX or Mac OS X . . . . . . . . 80
Components of the IDLDE

The IDL Development Environment (IDLDE) is a convenient multiple-document graphical user interface that includes built-in editing and debugging tools. This section describes briefly the components of the IDLDE. The Windows version is shown on the left and the UNIX version is shown on the right within the following figure.

![Figure 3-1: The IDL Development Environment for Windows (left) and UNIX (right).](image)

**Note**

Individual components are similar across platforms.
Menu Bar

The menu bar, located at the top of the main IDLDE window, allows you to control various IDLDE features. When you select an option from a menu item in the IDLDE, the Status Bar displays a brief description.

You can display menu commands for each menu using the following methods:

- Clicking the menu on the Menu bar.
- Pressing the Alt key plus the underlined letter in the menu’s title. For example, to display the File menu, press Alt+F.

You can select or execute a menu command using the following methods:

- Clicking the item in the menu.
- Pressing the Alt key plus the underlined letter in the menu’s title, and then pressing the letter underlined in the menu item. For example, to select the menu item File → Open, press Alt+F+O.
- Using the cursor and the arrow keys to highlight a menu item, and then pressing the Enter key.

Note

Many items (on each platform) have keyboard shortcuts displayed to the right of the corresponding menu option.

The menu bar consists of the following menu items:

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description of Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Menu</td>
<td>The File Menu gives you options such as opening, closing and creating new Editor windows and Projects and other options such as printing, printer setup, preferences and exiting IDL.</td>
</tr>
<tr>
<td>Edit Menu</td>
<td>The Edit Menu provides edit-related options such as undo, redo, cut, copy, paste, delete, select all, clear all and clear log.</td>
</tr>
<tr>
<td>Search Menu</td>
<td>The Search Menu allows you to find text in currently active Editor windows as well as other options such as find again, find selection, enter selection, replace, replace &amp; find, go to line and go to definition.</td>
</tr>
</tbody>
</table>

Table 3-1: The IDLDE Menus
Chapter 3: The IDL Development Environment

Components of the IDLDE

IDL Interface

Toolbars

There are three toolbars in the IDLDE: **Standard, Run & Debug**, and **Macros**. In addition, when you open an IDL GUIBuilder window (Windows only), its associated toolbar is displayed. When you position the mouse pointer over a toolbar button, the Status Bar displays a brief description. If you click on a toolbar button which represents an IDL command, the IDL command issued is displayed in the Output Log. Display or hide toolbars by making selections among the **Windows → Toolbars** items.

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description of Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Run Menu</strong></td>
<td>Run Menu items are enabled when an IDL program is loaded into an IDL Editor window. The run menu allows you program related functionality such as compiling, resolving dependencies, resetting, and editing programs among other things. For more information on running programs in IDL, see Chapter 2, “Creating and Running Programs in IDL” (<em>Application Programming</em>).</td>
</tr>
<tr>
<td><strong>Project Menu</strong></td>
<td>The Project Menu provides project-related functionality such as adding/removing files, grouping and moving files, building, running and exporting projects and so on. For more information on working with IDL projects, see Chapter 22, “Creating IDL Projects” (<em>Application Programming</em>).</td>
</tr>
<tr>
<td><strong>Macros Menu</strong></td>
<td>The Macro Menu provides functionality for creating new macros and using existing macros in IDL. For more about working with macros in IDL, see Chapter 5, “Creating Development Environment Macros”.</td>
</tr>
<tr>
<td><strong>Window Menu</strong></td>
<td>The Window Menu gives functionality related to Multiple Document Panel windows.</td>
</tr>
<tr>
<td><strong>Help Menu</strong></td>
<td>The Help Menu allows you to call IDL Online Help. You can call the entire Online Help system in the IDL Online Help Viewer or find help by topic. For more information on the IDL Help System, see “Getting Help with IDL” on page 35.</td>
</tr>
</tbody>
</table>

*Table 3-1: The IDLDE Menus (Continued)*
Chapter 3: The IDL Development Environment

Project Window

The Project Window displays information about the current Project you have open in the IDLDE. IDL Projects allow you to easily develop applications in IDL. Through a Project, you can compile, run, and create distributions of your IDL application. The IDL Project Window allows you to access and manage all of the files required for your application. This makes it easier to create a distribution for other developers, colleagues, or users.

For further information on the IDL Projects, refer to Chapter 22, “Creating IDL Projects” (Application Programming).

Multiple Document Panel

The section of the main IDL window where IDL Editor windows and GUIBuilder windows are displayed is known as the multiple document panel. Any number of files may be open at a single time. You can access different files from the Windows menu by clicking on the appropriate file.

Editor Windows

IDL Editor windows allow you to write and edit IDL programs (and other text files) from within IDL. Any number of Editor windows can exist simultaneously. No Editor windows are open when IDL is first started. Editor windows can be created by selecting File → New or File → Open. See “Maximizing the Editor’s Capabilities” (Chapter 2, Application Programming) for more information on the IDL Editor.

To see the Multiple Document Panel at work, open the file examples.pro by typing .COMPILE examples.pro at the IDL command line. (See “Command Line” on page 56 for details.)

The following figure shows the IDL program file opened in the Windows IDLDE.

Figure 3-2: Editor Window showing example.pro
Components of the IDL Development Environment

Chapter 3: The IDL Development Environment

GUIBuilder Windows

Under Microsoft Windows, IDL GUIBuilder windows allow you to interactively create user interfaces. Then, you can generate the IDL code that defines the interface and the code to contain the event-handling routines. You can modify the code, compile, and run the application in the IDLDE. To open a GUIBuilder window, you can select File → New → GUI or you can select File → Open. See Chapter 3, “Using the IDL GUIBuilder” (Widget Application Programming), for more information on the GUIBuilder.

Graphics Windows

IDL Graphics windows are not displayed in the Multiple Document Panel, but do appear when you use IDL to plot or display data. You can copy the contents of a Graphics window—iTool, Object or Direct—directly to the operating system clipboard in a bitmap format using CTRL+C.

When an IDL Graphics window is minimized (iconized), the icon displays the name of the IDL window. This icon appears on the desktop, not in the Multiple Document Panel, as with an iconized Editor window.

Warning

If the backing store is not set when a window is iconized, it will not be refreshed upon return. For more information about setting the backing store for graphics windows, see “Graphics Preferences” on page 102.

Command Line

The Command Line is an IDL prompt where you can enter IDL commands. The text output by IDL commands is displayed in the Output Log window. IDL is an interpreted language and commands entered at the Command Line are executed immediately. To see the IDL Command Line in action, enter the following in the Command Line at the IDL prompt and press Enter:

```
pin, 'Hello World!'
```

Figure 3-3: IDLDE Command Line
If you click the right mouse button while positioned over the Command Input Line, a popup menu appears displaying the last 20 entries in the command history. Select an entry to reissue the command. See “Recalling Commands” (Chapter 2, Application Programming) for additional information about the command recall buffer.

**Output Log**

Output from IDL is displayed in the Output Log window, which appears by default when the IDLDE is first started. Notice the result of our print command in the Output Log in the following figure.

![Output Log](image)

*Figure 3-4: The IDL Output Log*

If you click the right mouse button while positioned over the Output Log, a context menu appears allowing you to move to a specified error or clear the contents of the Output Log. An additional Windows-only context menu option allows you to copy selected contents.

**Variable Watch Window**

The Variable Watch window appears by default when you start the IDLDE. It keeps track of variables as they appear and change during program execution (tabs exist for viewing variables by type; Locals, Params, Common and System). For more information about the Variable Watch window, see “The Variable Watch Window” (Chapter 8, Application Programming).

**Status Bar**

When you position the mouse pointer over a Control Panel or Toolbar button, or select an option from a menu in IDLDE, the Status Bar displays a brief description.

**Docking/Undocking**

In IDL for Windows, four sections of the IDLDE can be moved within and unanchored from the main IDLDE window: the Toolbars, Output Log, Variable
Watch Window, and Command Line. Click on the border and drag the left mouse button. You will notice the outline of the chosen section moving with your mouse. When a location is chosen, release the mouse button to dock the window. If you move this outline so that it overlaps an edge of the window space being used by the IDLDE, the section will be docked to the nearest available side of the main IDLDE window. The Toolbars, Output Log, Variable Watch window, and Command Line will remain between the Menu Bar and the Status Bar when docked. They can be docked in any order to an edge. If the outline doesn’t overlap an edge, the section will float on the desktop. If you hold down the [Ctrl] key, the sections will float instead of docking to the nearest available side of the IDLDE.

**Control Panel Buttons**

In IDL for UNIX, the Control Panel buttons issue IDL commands for the currently-selected Editor window when pressed. The IDL command issued is displayed in the Output Log. By default, there are three different toolbars and the buttons displayed as well as the commands they issue are completely configurable (see Chapter 4, “Setting IDL Preferences” for more on these toolbars). When you position the mouse pointer over a Control Panel Button, the Status Bar displays a brief description.
Chapter 3: The IDL Development Environment

File Menu

The following options are available in the File menu.

**Note**
See “Using Keyboard Accelerators” on page 33 for information about using IDL's keyboard shortcuts on a Macintosh.

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New</td>
<td>Select from the following sub-menu items:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Editor</strong> [Ctrl+N]: Opens a new IDL Editor window.</td>
</tr>
<tr>
<td></td>
<td>• <strong>GUI</strong> (Microsoft Windows Only): opens a new IDL GUIBuilder file. See Chapter 3, “Using the IDL GUIBuilder” (Widget Application Programming) for details.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Project...</strong>: opens the <strong>New Project</strong> dialog.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Visualization</strong>: Launches an iTool. See “Introducing the iTools” (Chapter 1, <em>iTool User’s Guide</em>).</td>
</tr>
</tbody>
</table>

New windows are Untitledn or UntitledPrcn (where n is the numerical index of the new file) until saved with another name.

*Table 3-2: IDLDE File Menu Items*
### Chapter 3: The IDL Development Environment

#### File Menu

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open...</td>
<td>Select this option to open single or multiple text files for editing. (On Microsoft Windows platforms, you can also select an IDL GUIBuilder *.prc portable resource file.) In the Open dialog, you can select a continuous range of files by holding down the Shift key after selecting the first file, or select multiple, separate files by selecting each file while holding down the Control key. A new IDL Editor window is created to contain each text file. <strong>Note</strong> - On Motif platforms, if the Multiple Windows option is selected, a new IDL Editor window is created outside the main window to contain each text file. See “Layout Preferences” on page 98 for details. <strong>Note</strong> - You can also open text files from the Command Line. Enter the following at the IDL prompt: <code>.EDIT file1 [file2 ... fileₙ]</code> where file is the name of the text file you want to open. If the file is not in a directory included in the <code>!PATH</code> system variable, you must enter the full path for file. See “.EDIT” (IDL Reference Guide) for more information.</td>
</tr>
<tr>
<td>Close</td>
<td>Select this option to close the currently-selected IDL Editor window. If you have made changes in an IDL Editor window, you are asked if you want to save the changes before closing the window.</td>
</tr>
<tr>
<td>Open Project...</td>
<td>Select this option to open a new IDL Project. The Open dialog appears. Select the project you want to open and click Open.</td>
</tr>
<tr>
<td>Save Project</td>
<td>Select this option to save the current IDL Project. If the Project has not yet been saved, you are prompted for a filename with the Save As dialog.</td>
</tr>
<tr>
<td>Save Project As...</td>
<td>Select this option to save the current IDL Project to a specified filename. The Save As dialog appears.</td>
</tr>
<tr>
<td>Close Project</td>
<td>Select this option to close the current IDL Project. If you have made changes in to the project, you are asked if you want to save the changes before closing the window.</td>
</tr>
</tbody>
</table>

*Table 3-2: IDLDE File Menu Items (Continued)*
<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Save**       | Select this option to save the contents of an IDL Editor window. If the file has not yet been saved, you are prompted for a filename with the **Save As** dialog.  
**Note** - Changes made to a previously-compiled routine are not available to IDL until that routine is re-compiled. Executing the routine without first saving and re-compiling simply re-runs the previously-compiled version, without incorporating recent changes.  
Select the **Compile** option in the **Run** menu to return to the main program level and re-compile the routine. Select **Compile from Memory** in the **Run** menu to save and compile recent changes to a temporary file. |
| **Save As...** | Select this option to save the contents of an IDL Editor window to a specified filename. The **Save As** dialog appears. On Windows, when the **File → Save As...** option is selected, the default file name is the name of the last procedure or function in the file. On UNIX, the default file name is *.pro. For portability between platforms, the filename is lowercase letters. |
| **Ctrl+S**     |                                                                                                                                                                                                            |
| **Revert to Saved** | Select this option to reload the last saved version of the document.  
**Warning** - Unsaved changes are lost without warning.                                                                                     |
| **Generate .pro** | Microsoft Windows Only  
On a Microsoft Windows system, select this option to generate source code files from GUIBuilder interface definitions. When you generate code for the first time, all options open the **Save As** dialog so that you can select a location and specify a filename.  
The following are generated:  
- The widget definition code to a *.pro file.  
- The event-handler callback code to a *_eventcb.pro file.  
For information about the IDL GUIBuilder generated code, see “Generating Files” (Chapter 3, *Widget Application Programming*). |

Table 3-2: IDLDE File Menu Items (Continued)
### Chapter 3: The IDL Development Environment

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Print...</strong></td>
<td>On Microsoft Windows systems, select this option to immediately print the contents of the currently-selected window to the default printer.</td>
</tr>
<tr>
<td><strong>Ctrl+P</strong></td>
<td>On Motif systems, the <strong>Print</strong> dialog appears. Select <strong>Numbered Lines</strong> to include line numbers in the printout. Select <strong>Wrapped Lines</strong> to cause lines longer than the width of the printed page to wrap to a new line. Select <strong>Two Pages</strong> to print two pages per sheet of paper (each logical page is printed at half normal size). Select <strong>Header</strong> to include file information at the top of each page.</td>
</tr>
<tr>
<td><strong>Print Setup...</strong></td>
<td>Select this option to change the printer and printing options. The <strong>Print</strong> (Windows) or <strong>Printer Setup</strong> (Motif) dialog appears. For further information on setting up a printer, see “Printing in IDL” on page 79.</td>
</tr>
<tr>
<td><strong>Recent Files</strong></td>
<td>Select this option to open recently opened or created files. This menu item lists the last ten opened or created files. (On Microsoft Windows systems, it includes both text and GUIBuilder files.) To open a file on this list, select it. On Motif systems, to change the maximum number of files displayed from ten to another number, modify the <code>idlde.numRecentFiles</code> resource in your <code>.idlde</code> resource file. See Chapter 6, “Customizing IDL on Motif Systems”, for details.</td>
</tr>
<tr>
<td><strong>Recent Projects</strong></td>
<td>Select this option to open recently opened project files.</td>
</tr>
<tr>
<td><strong>Preferences...</strong></td>
<td>Select this option to display the tabbed Preferences dialog, which allows you to customize your interaction with the IDLDE. The options available via the Preferences dialog are described in detail in Chapter 4, “Setting IDL Preferences”.</td>
</tr>
<tr>
<td><strong>Exit</strong></td>
<td>Select this option to exit IDL.</td>
</tr>
</tbody>
</table>

*Table 3-2: IDLDE File Menu Items (Continued)*
Edit Menu

The following options are available in the **Edit** menu.

**Note**

See “Using Keyboard Accelerators” on page 33 for information about using IDL’s keyboard shortcuts on a Macintosh.

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Undo</strong></td>
<td>Select this option to undo previous editing actions. Multiple undo operations are supported; the first reverses the most recent operation, the next reverses the second most recent operation, etc. If the most recent action is irreversible, this option will not be accessible.</td>
</tr>
<tr>
<td>Ctrl+Z (Windows)</td>
<td></td>
</tr>
<tr>
<td>Alt+Z (Motif)</td>
<td></td>
</tr>
<tr>
<td><strong>Redo</strong></td>
<td>Select this option to redo previously undone editing actions. Successive redo operations are supported; the first redo reverses the most recent undo, etc.</td>
</tr>
<tr>
<td>Ctrl+Y (Windows)</td>
<td></td>
</tr>
<tr>
<td>Alt+Y (Motif)</td>
<td></td>
</tr>
<tr>
<td><strong>Cut</strong></td>
<td>Select this option to remove currently-selected text from an IDL Editor window or the Command Line to the Windows clipboard.</td>
</tr>
<tr>
<td>Ctrl+X (Windows)</td>
<td></td>
</tr>
<tr>
<td>Alt+X (Motif)</td>
<td></td>
</tr>
<tr>
<td><strong>Copy</strong></td>
<td>Select this option to copy the currently-selected text in an IDL Editor window, Output Log window, or Command Line to the clipboard. <strong>Copy</strong> also allows you to copy graphics from an IDL graphics window or draw widget to the clipboard.</td>
</tr>
<tr>
<td>Ctrl+C (Windows)</td>
<td></td>
</tr>
<tr>
<td>Alt+C (Motif)</td>
<td></td>
</tr>
<tr>
<td><strong>Paste</strong></td>
<td>Select this option to paste the contents of the Windows clipboard at the current insertion point. The insertion point can only be placed in an IDL Editor window.</td>
</tr>
<tr>
<td>Ctrl+V (Windows)</td>
<td></td>
</tr>
<tr>
<td>Alt+V (Motif)</td>
<td></td>
</tr>
<tr>
<td><strong>Comment</strong></td>
<td>Add the comment character (;) to a line or selected block of text in the Editor window.</td>
</tr>
<tr>
<td><strong>Uncomment</strong></td>
<td>Remove the comment character (;) from a line or selected block of text in the Editor window.</td>
</tr>
</tbody>
</table>

*Table 3-3: IDLDE Edit Menu Items*
### Menu Item | Description
--- | ---
**Delete** | Select this option to delete the currently-selected text. The deleted text is not placed on the clipboard.
**Del** |  
**Select All** | Use this option to highlight the entire contents of an IDL Editor window.
**Clear All** | Use this option to clear the entire contents of the current IDL Editor window.
**Ctrl-Del** (Windows) |  
**Clear Log** | Use this option to clear the entire contents of the Output Log.
**Ctrl+Y** (Motif) |  
**Properties** | Microsoft Windows Only
Select this option to open the GUIBuilder Properties dialog, which you can use to set the attribute and event properties for a widget. For information on the Properties dialog, see “Using the Properties Dialog” (Chapter 3, Widget Application Programming).
**Menu** | Microsoft Windows Only
Select this option to open the GUIBuilder Menu Editor, which you can use to define menus for top-level base widgets and button widgets. For information on the Menu Editor, see “Using the Menu Editor” (Chapter 3, Widget Application Programming).

*Table 3-3: IDLDE Edit Menu Items (Continued)*
Chapter 3: The IDL Development Environment

Search Menu

The following options are available in the **Search** menu.

**Note**

See “Using Keyboard Accelerators” on page 33 for information about using IDL’s keyboard shortcuts on a Macintosh.

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Find...</strong></td>
<td>Select this option to find text in an IDL Editor window or windows. The Search or Find/Replace dialog appears. Enter the text to find in the field marked <strong>Search for</strong> or <strong>Find</strong>: click <strong>Find next</strong> to highlight the search text in the currently active file.</td>
</tr>
</tbody>
</table>
| **Ctrl+F** (Windows) **Alt+F** (Motif) | **Platform Differences**  
- On Windows platforms, you can also choose an entry from the pulldown list of recent search terms rather than entering a new term in the **Search for** field.  
- On Windows platforms, you can specify replacement text by checking the **Replace with** checkbox and entering a replacement term. Click **Replace** to replace the selected text.  
Check the **Case sensitive** checkbox to match the case of the text you enter. Check **Whole words only** to match only entire words (the default is to match sub-strings). To replace all instances of the search text, check the **Replace all** checkbox and click **Replace**. Select **Forward from cursor** or **Backward from cursor** to specify the direction in which you would like to begin the search, or **Entire file** to search from the beginning of the file.  
By default, the search will take place in the currently-selected window. Choose a different file or **All Windows** from the pulldown list marked **Search in file** to search other windows. |

Table 3-4: IDLDE Search Menu Items
<table>
<thead>
<tr>
<th><strong>Menu Item</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Find Again</strong></td>
<td>Select this option to repeat the previous Find operation.</td>
</tr>
<tr>
<td>F3 (Windows) Alt+G (Motif)</td>
<td></td>
</tr>
<tr>
<td><strong>Find Selection</strong></td>
<td>Select this option to find the next occurrence of the selected text in an IDL Editor window.</td>
</tr>
<tr>
<td>Ctrl+E (Windows) Alt+I (Motif)</td>
<td></td>
</tr>
<tr>
<td><strong>Enter Selection</strong></td>
<td>Select this option to enter selected text in the Find field of the Find/Replace dialog.</td>
</tr>
<tr>
<td>Alt+T (Motif)</td>
<td>Motif Only</td>
</tr>
<tr>
<td><strong>Replace...</strong></td>
<td>Select this option to find text in an IDL Editor window and replace it with new text. The Replace dialog box appears. The Replace dialog has the same controls as the Search dialog, described above in the Find item. By default, the Replace with checkbox is checked.</td>
</tr>
<tr>
<td>Ctrl+H (Windows) Alt+R (Motif)</td>
<td></td>
</tr>
<tr>
<td><strong>Replace &amp; Find</strong></td>
<td>Select this option to repeat the most recent search-and-replace operation.</td>
</tr>
<tr>
<td>Alt+P</td>
<td>Motif Only</td>
</tr>
<tr>
<td><strong>Replace Again</strong></td>
<td>Select this option to repeat the previous Replace operation.</td>
</tr>
<tr>
<td>Shift+F3</td>
<td></td>
</tr>
<tr>
<td><strong>Go To Line...</strong></td>
<td>Select this option to jump directly to the specified line number in an IDL Editor window. The Go To Line dialog appears.</td>
</tr>
<tr>
<td>Ctrl+G</td>
<td></td>
</tr>
<tr>
<td><strong>Go To Definition</strong></td>
<td>Use this option to go to and mark with a current line indicator (blue arrow) the procedure or function call of the item next to which the cursor is positioned. The item must be either user-defined or a procedure or function written in IDL, and must have been compiled during the current IDLDE session.</td>
</tr>
<tr>
<td>Ctrl+D (Windows) Ctrl+T (Motif)</td>
<td></td>
</tr>
</tbody>
</table>

*Table 3-4: IDLDE Search Menu Items (Continued)*
Run Menu

**Run** menu items are enabled when an IDL program is loaded into an IDL Editor window and compiled. If you click the right mouse button while positioned over an editor window, a popup menu appears allowing you to quickly access several of the most convenient commands. The popup menu changes to display common debugging commands if IDL is running a program. See Chapter 8, “Debugging and Error-Handling” (Application Programming) for more information.

**Note**

See “Using Keyboard Accelerators” on page 33 for information about using IDL’s keyboard shortcuts on a Macintosh.

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compile filename.pro Ctrl+F5</td>
<td>Select this option to compile a .pro file. The currently-selected file is only recognized as an IDL procedure or function if suffixed with .pro. Selecting this option is the same as entering .COMPILE at the Command Line, with the appropriate Editor window selected in the Multiple Document Panel. You can also compile files from the Command Line. Enter the following at the IDL prompt: .COMPILE file1 [file2 ... filen] where file is the name of the file you want to open. IDL opens your files in editor windows and compiles the procedures and functions contained therein. If the path is not specified in the Path Preferences from the File menu, you must enter the full path for file. See “.COMPILE” (IDL Reference Guide) for a more detailed explanation.</td>
</tr>
</tbody>
</table>

*Table 3-5: IDLDE Run Menu Items*
### Menu Item

<table>
<thead>
<tr>
<th>Compile <code>filename.pro</code> from Memory</th>
<th>Ctrl+F6</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td></td>
</tr>
<tr>
<td>Select this option to save and compile changes to the current editor window without affecting the last-saved version of the file. The temporary file created allows you to experiment without committing changes to the permanent file. Selecting this option is the same as entering <code>.COMPILE -f</code> at the Command Line. See “<code>.COMPILE</code>” (<em>IDL Reference Guide</em>) for a more detailed explanation.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Compile All</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td></td>
</tr>
<tr>
<td>Select this option to compile all currently open <code>.pro</code> files.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Run <code>filename</code></th>
<th>F5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td></td>
</tr>
<tr>
<td>Select this option to execute the file called <code>filename</code> contained in the currently-active editor window. Selecting this option is the same as entering the <code>procedure name</code> at the Command Line or using the <code>.GO</code> executive command at the Command Line. If the file is interrupted while running, selecting this option resumes execution; it is the same as entering <code>.CONTINUE</code> at the Command Line. For more information, see <code>.CONTINUE</code> and <code>.GO</code> in the <em>IDL Reference Guide</em>.</td>
<td></td>
</tr>
</tbody>
</table>

**Warning** - In order for the **Run** option to reflect the correct procedure name in the **Run** menu, the `.pro` filename must be the same as the main procedure name. For example, the file named `squish.pro` must include: `pro squish` |

<table>
<thead>
<tr>
<th>Resolve Dependencies</th>
<th>Alt+F5 (Motif)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td></td>
</tr>
<tr>
<td>Select this option to iteratively compile all un-compiled IDL routines that are referenced in any open and compiled files. Selecting this option is the same as entering <code>RESOLVE_ALL, /QUIET</code> at the Command Line. The <code>QUIET</code> keyword suppresses informational messages. See “<code>RESOLVE_ALL</code>” (<em>IDL Reference Guide</em>) for a more detailed explanation.</td>
<td></td>
</tr>
</tbody>
</table>

---

*Table 3-5: IDLDE Run Menu Items (Continued)*
### Chapter 3: The IDL Development Environment

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Profile</strong></td>
<td>Select this option to access the Profile dialog. The IDL Code Profiler allows you to analyze the performance of your applications. You can identify which modules are used most frequently, and which modules take up the greatest amount of time. For more information about the IDL Code Profiler, see “The IDL Code Profiler” (Chapter 11, Application Programming).</td>
</tr>
</tbody>
</table>
| **Test GUI Ctrl+T** | Microsoft Windows Only  
Select this option to test the GUI interface in a GUIBuilder window. This option allows you to see how the interface you have designed will look when it is running.  
To exit test mode:  
Press the Esc key.  
or  
Click the X in the upper-right corner of the application window of the running test application.  
*Note* - This option is not available if a blocking widget is currently active. |
| **Break Ctrl+Break (Windows) Ctrl+C (Motif)** | Select this option to interrupt program execution. IDL inserts a marker to the left of the line at which program execution was interrupted. |
| **Stop Ctrl+R** | Select this option to stop program execution and return to the main program level. Selecting this item is the same as entering the following at the Command Line:  
```
RETAIL
WIDGET_CONTROL,/RESET
CLOSE,/ALL
HEAP_GC,/VERBOSE
```
See RETALL, WIDGET_CONTROL, CLOSE, or HEAP_GC in the IDL Reference Guide for details. |

*Table 3-5: IDLDE Run Menu Items (Continued)*
Chapter 3: The IDL Development Environment

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reset</strong></td>
<td>Select this option to completely reset the IDL environment. This option executes .RESET_SESSION. See “.RESET_SESSION” (IDL Reference Guide) for details.</td>
</tr>
<tr>
<td><strong>Step Into</strong>&lt;br&gt;F8</td>
<td>Select this option to execute a single statement in the current program. The current-line indicator advances one statement. If the statement being stepped into calls another IDL procedure or function, statements from that procedure or function are executed in order by successive Step commands. Selecting this item is the same as entering .STEP at the IDL Command Line. See “.STEP” (IDL Reference Guide) for a more detailed explanation.</td>
</tr>
<tr>
<td><strong>Step Over</strong>&lt;br&gt;F10</td>
<td>Select this option to execute a single statement in the current program. The current-line indicator advances one statement. If the statement which is stepped over calls another IDL procedure or function, statements from that procedure or function are executed to the end without interactive capability. Selecting this item is the same as entering .STEPOVER at the IDL Command Line. See “.STEPOVER” (IDL Reference Guide) for details.</td>
</tr>
<tr>
<td><strong>Step Out</strong>&lt;br&gt;Ctrl+F8</td>
<td>Select this option to continue processing until the current program returns. Selecting this item is the same as entering .OUT at the IDL Command Line. See “.OUT” (IDL Reference Guide) for a more detailed explanation.</td>
</tr>
<tr>
<td><strong>Trace...</strong></td>
<td>Select this option to access the Trace Execution dialog. You can modify the interval between successive .STEP or .STEPOVER commands, depending on whether <strong>Step into routines</strong> or <strong>Step over routines</strong> is checked. The current-line indicator points to program lines as they are executed. Selecting this item at full speed is the same as entering .TRACE at the IDL command prompt. See “.TRACE” (IDL Reference Guide) for a more detailed explanation.</td>
</tr>
</tbody>
</table>

Table 3-5: IDLDE Run Menu Items (Continued)
### Table 3-5: IDLDE Run Menu Items (Continued)

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run to Cursor F7</td>
<td>Select this option to execute statements in the current program up to the line where the cursor is positioned. Selecting this item is the same as setting a one-time breakpoint at a specific line. See “BREAKPOINT” <em>(IDL Reference Guide)</em> for details.</td>
</tr>
<tr>
<td>Run to Return Ctrl+F7</td>
<td>Select this option to execute statements in the current procedure or function up to the line where the return is positioned. Selecting this item is the same as setting a one-time breakpoint at a specific line. See “.RETURN” <em>(IDL Reference Guide)</em> for details.</td>
</tr>
<tr>
<td>Set Breakpoint Clear Breakpoint F9</td>
<td>Select this option to set or clear a breakpoint on the current line. See Chapter 8, “Debugging and Error-Handling” <em>(Application Programming)</em> for details.</td>
</tr>
<tr>
<td>Disable Breakpoint Ctrl+F12 (Motif)</td>
<td>Select this option to access disable a breakpoint in the current line. See Chapter 8, “Debugging and Error-Handling” <em>(Application Programming)</em> for details.</td>
</tr>
<tr>
<td>Edit Breakpoint...</td>
<td>Select this option to access the Edit Breakpoint dialog. See Chapter 8, “Debugging and Error-Handling” <em>(Application Programming)</em> for details.</td>
</tr>
<tr>
<td>Up Stack Ctrl+Up Arrow</td>
<td>Select this option to move up the call stack by one.</td>
</tr>
<tr>
<td>Down Stack Ctrl+Down Arrow</td>
<td>Select this option to move down the call stack by one.</td>
</tr>
<tr>
<td>List Call Stack</td>
<td>Select this option to display the current nesting of procedures and functions. Selecting this item is the same as entering HELP, /TRACEBACK at the IDL Command Line. See “HELP” <em>(IDL Reference Guide)</em> for details.</td>
</tr>
</tbody>
</table>
Chapter 3: The IDL Development Environment

Project Menu

For more information on the following Project menu items, see Chapter 22, “Creating IDL Projects” (Application Programming).

Note

See “Using Keyboard Accelerators” on page 33 for information about using IDL’s keyboard shortcuts on a Macintosh.

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add/Remove Files...</td>
<td>Select this option to add or remove files from the current project.</td>
</tr>
<tr>
<td>Remove Selected</td>
<td>Motif Only</td>
</tr>
<tr>
<td>Ctrl+H</td>
<td>Select this option to remove the currently selected file from your IDL Project.</td>
</tr>
<tr>
<td>Move To</td>
<td>Motif Only</td>
</tr>
<tr>
<td></td>
<td>Select this option to move the currently selected file to the indicated project directory.</td>
</tr>
<tr>
<td>Groups...</td>
<td>Selecting this option displays the Project Groups dialog from which you can create a new group or rename, remove, move up or down, or set to filter specific file types for the default groups within an IDL Project.</td>
</tr>
<tr>
<td>Options...</td>
<td>Select this option to change the options for a project. The Project Options dialog is displayed.</td>
</tr>
<tr>
<td>Compile</td>
<td>Select this option to compile files in a project. You can choose either All Files to compile all the source files in a project or Modified Files to compile only the files that have been modified since the last compile.</td>
</tr>
<tr>
<td>Build</td>
<td>Select this option to build your project.</td>
</tr>
<tr>
<td>Run</td>
<td>Select this option to run the project application.</td>
</tr>
<tr>
<td>Export</td>
<td>Select this option to export your project.</td>
</tr>
</tbody>
</table>

Table 3-6: IDLDE Project Menu Items
Macros Menu

The following options are available in the Macros menu.

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit...</td>
<td>Select this item to access the Edit Macros dialog. Macros which have already been defined are listed in the Macros field. To edit a macro, click on the macro to access its characteristics and click OK when your adjustments are complete. To add a macro, click Add..., which will access the Add Macro dialog. Enter the name of the new macro in the given field and click OK. Enter the IDL command to be executed by the new macro in the IDL Command field. Enter the menu item name, the full path to the toolbar bitmap file, the tooltip text, and the status bar text in the appropriate fields. Select the accelerator by specifying the key in the Key field and then optionally clicking on any combination of Ctrl, Alt and Shift. Note - Bitmap files for toolbar buttons must be 16 pixels by 16 pixels, and must contain 256 colors or fewer. To remove a macro, click Remove. To change the position of a macro in the Macro menu and on the Macro Toolbar, click on the macro to highlight it and click on either Move Up or Move Down.</td>
</tr>
<tr>
<td>Import...</td>
<td>Microsoft Windows Only Use this menu selection to display the Import Macros dialog box. Use this dialog to select the previous IDL installation from which you want macros to be imported.</td>
</tr>
</tbody>
</table>

Table 3-7: IDLDE Macros Menu Items
### Table 3-7: IDLDE Macros Menu Items (Continued)

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Print Var</strong> (Windows)</td>
<td>Select this option to print the selected variable. Selecting this item is the same as entering <code>PRINT, x</code> at the IDL Command Line, where <code>x</code> is the selected variable.</td>
</tr>
<tr>
<td><strong>Print Variable</strong> (Motif)</td>
<td></td>
</tr>
<tr>
<td><strong>Help On Var</strong> (Windows)</td>
<td>Select this option to list attributes of the selected variable. Selecting this item is the same as entering <code>HELP, x, /STRUCTURE</code> at the IDL Command Line, where <code>x</code> is the selected variable.</td>
</tr>
<tr>
<td><strong>Help On Variable</strong> (Motif)</td>
<td></td>
</tr>
<tr>
<td><strong>Import Image</strong></td>
<td>Select this option to import an image file into IDL. For more information, see “Using Macros to Import Image Files” on page 167.</td>
</tr>
<tr>
<td><strong>Import ASCII</strong></td>
<td>Select this option to import an ASCII file into IDL. For more information, see “Using Macros to Import ASCII Files” (Chapter 6, Using IDL).</td>
</tr>
<tr>
<td><strong>Import Binary</strong></td>
<td>Select this option to import a binary file into IDL. For more information, see “Using Macros to Import Binary Files” (Chapter 6, Using IDL).</td>
</tr>
<tr>
<td><strong>Import HDF</strong></td>
<td>Select this option to import an HDF file into IDL. For more information, see “Using Macros to Import HDF Files” (Chapter 6, Using IDL).</td>
</tr>
<tr>
<td><strong>Demo</strong></td>
<td>Select this option to access IDL’s Demo application.</td>
</tr>
<tr>
<td><strong>Export Bridge Assistant</strong></td>
<td>Select this option to build a Java or COM object from an IDL object. For more information, see “Using the Export Bridge Assistant” (Chapter 7, IDL Connectivity Bridges).</td>
</tr>
</tbody>
</table>
Chapter 3: The IDL Development Environment

Window Menu

The following options are available in the Window menu.

Note
See “Using Keyboard Accelerators” on page 33 for information about using IDL's keyboard shortcuts on a Macintosh.

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read Only</td>
<td>Motif Only</td>
</tr>
<tr>
<td></td>
<td>Select this option to enable or disable editing of the currently selected window. A filled square next to the item indicates Read-Only status.</td>
</tr>
<tr>
<td>Next</td>
<td>F6 (Windows) F11 (Motif)</td>
</tr>
<tr>
<td></td>
<td>Select this option to shift IDL's focus to the next numbered editor window.</td>
</tr>
<tr>
<td>Previous</td>
<td>Shift+F6 (Windows) Alt+F11 (Motif)</td>
</tr>
<tr>
<td></td>
<td>Select this option to shift IDL’s focus to the previous numbered editor window.</td>
</tr>
<tr>
<td>Cascade</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Select this option to cascade all the IDL Editor windows within the main window.</td>
</tr>
<tr>
<td>Tile Horizontally</td>
<td>Microsoft Windows Only</td>
</tr>
<tr>
<td></td>
<td>Select this option to tile all the IDL Editor windows on top of one another within the main window.</td>
</tr>
<tr>
<td>Tile Vertically</td>
<td>Microsoft Windows Only</td>
</tr>
<tr>
<td></td>
<td>Select this option to tile all the IDL Editor windows side-by-side within the main window.</td>
</tr>
<tr>
<td>Tile</td>
<td>Motif Only</td>
</tr>
<tr>
<td></td>
<td>Select this option to arrange all open windows in a non-overlapping fashion.</td>
</tr>
</tbody>
</table>

Table 3-8: IDLDE Window Menu Items
## Chapter 3: The IDL Development Environment

### Window Menu

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Arrange Icons</strong></td>
<td>Select this option to arrange all minimized Editor or Graphics windows.</td>
</tr>
<tr>
<td><strong>Close All</strong></td>
<td>Select this option to close all IDL Editor windows. If the text within an IDL Editor window has changed, you are asked if you want to save the file before closing.</td>
</tr>
</tbody>
</table>
| **Configure**     | Motif Only  
Select this option to access a pulldown menu which alters the appearance of the IDLDE. Select each toggle option to hide or show each component. For more information about each component, see “Components of the IDLDE” on page 52.  
- Hide Control  
- Hide View  
- Hide Log  
- Hide Variable Watch  
- Hide Command  
- Hide Status  
- Hide Project |
| **Command Input** | Microsoft Windows Only  
If this menu item has a check mark by it, the IDL Command Line is visible in the main IDL window. If this item does not have a check mark next to it, the IDL command line is not visible. Use this menu item to toggle between the two states. |
| **Output Log**    | Microsoft Windows Only  
If this menu item has a check mark by it, the Output Log is visible in the main IDL window. If this item does not have a check mark next to it, the Multiple Document Panel is maximized in the main IDL window. Use this menu item to toggle between the two states. |

Table 3-8: IDLDE Window Menu Items (Continued)
### Chapter 3: The IDL Development Environment

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
</table>
| Variable Watch Ctrl+A | Microsoft Windows Only  
If this menu item has a check mark by it, the **Variable Watch Window** is visible in the main IDL window. If this item does not have a check mark net to it, the **Variable Watch Window** is not visible. Use this menu item to toggle between the two states. |
| Project            | Microsoft Windows Only  
If this menu item has a check mark by it, the **Project Window** is visible in the main IDL window. If this item does not have a check mark net to it, the **Project Window** is not visible. Use this menu item to toggle between the two states. |
| Toolbars           | Select this option to access a pulldown menu with the three Windows toolbars: **Standard**, **Run & Debug**, and **Macros**. If a toolbar has a check mark by it, it is visible below the menu bar items. |
| Status Bar         | Microsoft Windows Only  
If this menu item has a check mark by it, the **Status** bar is visible at the very bottom of the Main IDL window. |
| Numbered Windows   | The numbered menu items at the bottom of the **Window** menu display open files. Select any of these menu items to make that window the current window. |

*Table 3-8: IDLDE Window Menu Items (Continued)*
Help Menu

The following options are available in the Help menu.

Note
See “Using Keyboard Accelerators” on page 33 for information about using IDL’s keyboard shortcuts on a Macintosh.

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contents</td>
<td>Select this menu item to display the IDL Online Help Viewer.</td>
</tr>
<tr>
<td>Ctrl+F1</td>
<td></td>
</tr>
<tr>
<td>Find Topic...</td>
<td>Select this menu item to display the Search dialog for IDL Online Help.</td>
</tr>
<tr>
<td>F1</td>
<td></td>
</tr>
<tr>
<td>About IDL...</td>
<td>Select this option to display information on the IDL version in use.</td>
</tr>
</tbody>
</table>

Table 3-9: IDLDE Help Menu Items
Printing in IDL

IDL allows you two ways to print:

- Printing graphics from the IDL language
- Printing IDL source code from the **File** menu of the IDLDE.

While these sources are fundamentally different, the methods used to specify and configure a print device according to your operating system are the same. This topic is covered in the following sections. See “Printing Graphics” (Chapter 8, *Using IDL*) for information on how to print from an IDL program.

Printer setup in Windows is relatively straightforward, and is described in the following section. UNIX printer setup is slightly more involved and is covered in “IDL Printer Setup in UNIX or Mac OS X” on page 80.

**IDL Printer Setup in Windows**

Setting up a printer in IDL for Windows uses the common Windows Printer Setup dialog. For more information on setting up a Printer on Windows, see your Windows operating system documentation or support.

![Common Printer Setup Dialog in Windows](image_url)

*Figure 3-5: Common Printer Setup Dialog in Windows*
Chapter 3: The IDL Development Environment

IDL Printer Setup in UNIX or Mac OS X

IDL for UNIX uses the Xprinter print technology from Bristol Technology to create and output information to a wide variety of printers. This section describes the Xprinter setup dialogs.

The Xprinter Setup Dialog

The Xprinter Setup dialog allows you to select model-specific printer options such as paper trays, paper size, page orientation, and the UNIX print spooler command. Printer options are saved in the $HOME/.XprinterDefaults file. Once configured, the desired information is saved to the file system and used in future IDL sessions.

Figure 3-6: The Printer Setup Dialog
Chapter 3: The IDL Development Environment

Printer Setup Dialog Buttons

The action area of the Printer Setup dialog contains six buttons:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OK</td>
<td>Writes current configuration information to your default printer information file $HOME/.XprinterDefaults. This button also dismisses the dialog.</td>
</tr>
<tr>
<td>Save</td>
<td>Writes current configuration information to your default printer information file $HOME/.XprinterDefaults.</td>
</tr>
<tr>
<td>Reset</td>
<td>Reloads default configuration from $HOME/.XprinterDefaults.</td>
</tr>
<tr>
<td>Cancel</td>
<td>Closes dialog and cancels all configuration changes.</td>
</tr>
<tr>
<td>Options</td>
<td>Displays the options dialog box that lets you select an alternate printer setup. This button is disabled if output is configured to be sent to a file instead of a printer.</td>
</tr>
<tr>
<td>Install</td>
<td>Displays the installation dialog box that allows you to add or remove printer devices and printer ports from the $HOME/.XprinterDefaults file.</td>
</tr>
</tbody>
</table>

Table 3-10: Printer Setup Dialog Buttons
## Configuring Printer Setup Options

Specify the following options on the initial Printer Setup dialog:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Output Format:</strong></td>
<td>Specify whether to send output to a file or a printer. If you choose Printer Specific, you can send output to any printer type/port combination configured in your $HOME/.XprinterDefaults file. If the port is FILE:, Xprinter creates an output file for the specified printer type. If you choose Generic (File Only), print output is sent to an Encapsulated PostScript or generic PCL file.</td>
</tr>
<tr>
<td><strong>Printer:</strong></td>
<td>This field appears only if you select Output Format: Printer Specific. It specifies the name of the default printer type/port to which to send print output. Click the Options button to specify a different printer type/port combination.</td>
</tr>
<tr>
<td><strong>File Name:</strong></td>
<td>This field appears only if you choose Output Format: Generic (File Only). Type the name of the print file you wish to create. To pipe print output to a command, enter a ! character as the first character and then specify the command to which to send output. For example, to send output to the lp command, enter the following: !lp</td>
</tr>
<tr>
<td><strong>Orientation</strong></td>
<td>Specify portrait or landscape.</td>
</tr>
<tr>
<td><strong>Scale</strong></td>
<td>To increase the size of the output, specify a value greater than 1.00. To reduce the size, specify a value less than 1.00. For example, a value of 2.00 would double the size of the output; a value of 0.50 would reduce it by half.</td>
</tr>
<tr>
<td><strong>Copies</strong></td>
<td>Specify the number of copies to print.</td>
</tr>
</tbody>
</table>

*Table 3-11: Specifying Printer Setup Options*
To set additional options, such as selecting a different printer or changing the page size, click the **Options** button. The Options dialog appears.

The Options dialog is only available when sending output to a printer.

![Options Dialog](image)

**Figure 3-7: The Options Dialog**

Use this dialog to set the Printer Setup options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Printer Name</strong></td>
<td>Use this field to select the current printer. Click the down arrow to display a list of configured printers.</td>
</tr>
<tr>
<td><strong>Resolution</strong></td>
<td>Specify printer resolution with this field. Values vary depending on printer.</td>
</tr>
<tr>
<td><strong>Page Size</strong></td>
<td>Specify paper size with this field. Values vary depending on printer.</td>
</tr>
<tr>
<td><strong>Paper tray</strong></td>
<td>Specify paper tray with this field. Values vary depending on printer.</td>
</tr>
<tr>
<td><strong>Duplex</strong></td>
<td>Specify duplex options (if the selected printer supports duplex printing). Valid values include None (no duplex printing), Duplex Tumble (flips over the short edge), and Duplex No Tumble (flips over the long edge). If the selected printer does not support duplexing, this field is disabled.</td>
</tr>
</tbody>
</table>

**Table 3-12: The Printer Setup Options**

**Adding a New Printer to the List of Printer Choices**

To add a new printer to your list of available printers:
Chapter 3: The IDL Development Environment

- Define a port, which is an alias for the print command.
- Associate the port with the printer’s PPD file.

**Defining a New Port**

To define a new port using the Printer Setup dialog:

1. Display the Ports dialog. From the Printer Setup dialog, select Install, Add Printer, and Define New Port.

2. Type the port definition in the **Edit Port** edit box. Port definitions have the following format:

```
port=print_command
```

The `print_command` is the command for sending output to the printer port. If you were to have two printers named ORION and SIRIUS for example, the definitions would appear as follows:

```
ORION=rsh bandit "lp -d ps"
SIRIUS=rsh bandit "lp -d ps -T pcl5"
```

Both printers here are connected to the system bandit, so the print command is a remote shell command executed on bandit. ORION is a PostScript printer, so the command `lp -d ps` is executed on bandit to print to ORION. SIRIUS though is a PCL5 printer, so the print command executed on bandit to print to SIRIUS is `lp -d ps -T pcl5`.

**Figure 3-8: Defining a New Port**

![Image of the Printer Setup dialog with ports ORION and SIRIUS defined.](image)
3. Click Add/Replace and the new port is now included in the list of current port definitions.

4. Repeat the above step for each printer to which you wish to send output.

**Note**

To create a printer port for each available queue on hp700 systems, click the Spooler button on the Ports dialog. This command creates a default printer port for each available printer queue returned by the `lpstat -a` command.

### Modifying an Existing Port

In order to modify an existing port using the Printer Setup dialog:

1. Display the Ports dialog. From the Printer Setup dialog, click **Install**, **Add Printer**, and **Define New Port**.

2. Select the port you wish to modify and edit the port information in the **Edit Port** edit box.

3. Click Add/Replace. The modified port is now included in the list of current port definitions.

### Matching a Printer Device to a Port

In order to match a printer device to a port using the Printer Setup dialog:

1. Display the Add Printer dialog. From the Printer Setup dialog, click **Install** and **Add Printer**.

2. In the **Printer Devices** field, select the description that matches the printer you are to install. If no description matches this printer, contact your printer vendor for a printer description (PPD) file.

3. Select the desired port in the Current Port Definitions list box and click Add Selected. The new printer is now included in the list of currently installed printers.
Chapter 3: The IDL Development Environment

Removing an Installed Printer

In order to remove a printer device/port combination using the Printer Setup dialog:

1. Display the Printer Installation dialog. From the Printer Setup dialog, click Install.

2. In the Currently Installed Printers list box, select the printer you wish to remove and click on Remove Selected.

Manually Modifying Default Printer Setup Values

Xprinter retrieves default printer setup information from the file .XprinterDefaults in your home directory. If this file does not exist, Xprinter reads the information from the file $XPHOME/xprinter/XprinterDefaults or $XPPATH/XprinterDefaults.

Note

For IDL, $XPATH is set to $IDL_DIR/resource/xprinter.

The Xprinter Printer Setup dialog writes modifications to the default information in $HOME/.XprinterDefaults. However, it never modifies the default information in the file $XPHOME/XprinterDefaults or $XPPATH/XprinterDefaults. If the file $HOME/.XprinterDefaults does not already exist, the Xprinter Printer Setup dialog creates it.

Although the most common way to modify the default Printer Setup is using the Printer Setup dialog, which updates $HOME/.XprinterDefaults automatically, you may also edit this file with any text editor and make changes directly.
You may also set up the `$HOME/.XprinterDefaults` file to do the following:

- Define printer ports.
- Match printer types to defined ports.
- Specify the default printer.
- Specify printer-specific options.

**Defining a Port**

A printer port is an alias for the print command. It is defined in the `[ports]` section of `$HOME/.XprinterDefaults` and appears as part of the Printer Name in the Printer Setup dialog. For instance, the following is the first Printer Name in the Printer Setup dialog before you make any changes to `$HOME/.XprinterDefaults`:

```
AppleLaserWriter v23.0 PostScript on FILE:
```

For this Printer Name, `FILE:` is the port name. To send output to a printer instead of a file, you first must define a port for each printer to which you wish to direct output. Port entries in the `[ports]` section have this format:

```
port=print_command
```

The `print_command` is the command for sending output to the printer port. For instance, if you have two printers (ORION and SIRIUS), your `[ports]` section may appear as follows:

```
[ports]
ORION=rsh bandit "lp -d ps"
SIRIUS=rsh bandit "lp -d ps -T pcl5"
```

In the above, both printers are connected to the system bandit, so the print command is a remote shell command executed on bandit. ORION is a PostScript printer, so the command `lp -d ps` is executed on bandit to print to ORION. SIRIUS, though, is a PCL5 printer, and thus the print command executed on bandit to print to SIRIUS is `lp -d ps -T pcl5`.

If a printer is connected to your local system, you will need to add an entry for that printer as well. For the local printer, your entry should be like the following:

```
[ports]
ORION=rsh bandit "lp -d ps"
SIRIUS=rsh bandit "lp -d ps -T pcl5"
LOCAL=lp -d ps
```

Your printer port can be any name you choose except `FILE:`, which is the only reserved port name. It causes Xprinter to create a print file formatted specifically for the specified printer type.
Chapter 3: The IDL Development Environment

An entry must be created in the [ports] section for every printer to which you wish to be able to print.

Matching a Printer Type to a Defined Port

After you have defined a port for each printer, you must tell Xprinter what type of printer is associated with each port. List device types in the [devices] section of the .XprinterDefaults file. Each entry in the [devices] section has the following format:

\[
\text{alias=PPD\_file driver,\text{port}}
\]

**Note**

There must be a space between the PPD_file and driver and a comma between the driver and the port. The following table describes each part of this entry.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alias</td>
<td>The alias is a descriptive name used to identify the printer. It can be anything you choose. The alias is the name which appears in the Printer Setup dialog (such as HP LaserJet III SI PostScript).</td>
</tr>
<tr>
<td>PPD_file</td>
<td>The PPD_file is the name of the printer description (PPD) file used by the printer, without a .PPD extension. Search in the directory $XPHOME/xprinter/ppds/ to find the PPD file for your printer.</td>
</tr>
<tr>
<td>driver</td>
<td>The driver is the type of driver your printer uses. Value values are PostScript, PCL4, and PCL5.</td>
</tr>
<tr>
<td>port</td>
<td>The port is the printer port as listed in the [ports] section of the .XprinterDefaults file (ORION, SIRIUS, and LOCAL in the example [ports] section).</td>
</tr>
</tbody>
</table>

*Table 3-13: Associating a Printer with a Port*
Here's an example configuring three printers:

<table>
<thead>
<tr>
<th>Port</th>
<th>Printer Type</th>
<th>Output Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORION</td>
<td>HP LaserJet IIISi PostScript v52.3</td>
<td>PostScript</td>
</tr>
<tr>
<td>SIRIUS</td>
<td>HP LaserJet 4M PCL Cartridge</td>
<td>PCL</td>
</tr>
<tr>
<td>LOCAL</td>
<td>QMS-PS 2200 v52.3</td>
<td>PostScript</td>
</tr>
</tbody>
</table>

**Table 3-14: Example Configuration**

First, be sure to choose an alias for each printer. In order to make it simpler to identify the printer from the Printer Setup dialog you wish to use, you may use the following aliases:

- HP LaserJet PS
- HP LaserJet PCL
- QMS PS

It is important to note that if you utilize the Printer Setup dialog to associate ports and PPD files, you cannot specify a printer alias. You must instead choose an alias from the predefined listing that appears in the Printer Devices list box in the Add Printer dialog. The corresponding PPD file is already associated with the printer aliases in this list box.

Now, identify the PPD file associated with each of these printers.

Thus the [devices] section of the .XprinterDefaults file would be as follows:

```plaintext
[devices]
HP LaserJet PS=HP3S1523 PostScript,ORION
HP LaserJet PCL=HP4M PCL,SIRIUS
QMS PS=Q2200523 PostScript,LOCAL
```

After these entries have been added to your .XprinterDefaults file, the following printer choices are available from the Printer Setup dialog:

- HP LaserJet PS on ORION
- HP LaserJet PCL on SIRIUS
- QMS PS on LOCAL

**Specifying a Default Printer**

After you have configured all available printers, you may select one of them as the default printer. To make a specific printer the default printer on the Printer Setup dialog, add an entry (in the following format) to the [windows] section of the .XprinterDefaults file:
Chapter 3: The IDL Development Environment

IDL Printer Setup in UNIX or Mac OS X

[windows]
device=PPD_file,driver,port

Simply provide the same information that you used in the [devices] section. Only the format of the entry is different; there is a comma between the PPD_file and the driver instead of a space.

For example, suppose you wish the default printer to be the printer at port ORION. The [windows] section would appear as follows:

[windows]
device=HP3SI523,PostScript,ORION
[ports]
ORION=rsh bandit "lp -d ps"
SIRIUS=rsh bandit "lp -d ps -T pcl5"
LOCAL=lp -d ps
[devices]
HP LaserJet PS=HP3SI523 PostScript,ORION
HP LaserJet PCL=HP4M PCL,SIRIUS
QMS PS=Q2200523 PostScript,LOCAL

In your default .XprinterDefaults file, the [windows] entry appears:

[windows]
device=NULL,PostScript,FILE:

Since no PPD file is listed (NULL), the default on the Printer Setup dialog is to print generic PostScript to a file. You may specify the filename and change the type of output to PCL on the Printer Setup dialog.

Specifying Printer-Specific Options

You may include a section that lists the default printer-specific options for each printer defined in the devices section. The options available vary between differing printers, but typical options include number of copies, page size, paper tray, and orientation. An example follows of a printer-specific section for a default printer in the example .XprinterDefaults file:

[HP3SI523,PostScript]
Scale=0.80
Copies=1
PaperTray=Lower
PageSize=Letter
Orientation=Portrait
DPI=300
Chapter 4
Setting IDL Preferences

The IDL Development Environment can be customized by setting preferences. This chapter describes the sections of the Preferences dialog:

- About IDL Preferences ................. 92
- Customizing IDL ......................... 93
- General Preferences ...................... 95
- Layout Preferences ...................... 98
- Graphics Preferences .................... 102
- Editor Preferences ....................... 105
- Startup Preferences ...................... 108
- Font Preferences ......................... 110
- Path Preferences ......................... 113
Chapter 4: Setting IDL Preferences

About IDL Preferences

Preferences are internal values that control various aspects of the environment that IDL presents to its users. Preferences supply initial values for many system variables and control the layout of the IDL development environment (IDLDE) and a variety of other aspects of IDL’s behavior. Preferences can be specified from a variety of sources. They persist between IDL sessions, meaning that once you get them set in a way that satisfies your needs, you can forget them, and IDL will behave in the way you have specified every time you run it.

You can specify values for many of the IDL preferences through the IDLDE’s Preferences dialog. For more information, see “Customizing IDL” on page 93.

Some preferences are not visible in the Preferences dialog. To customize them, use the IDL PREF_* routines, environment variables, or user preference files to specify preference/value pairs. You can also use these mechanisms to modify preference values visible in the Preferences dialog. For more information, see Appendix E, “IDL Preferences” (IDL Reference Guide).

Unavailable Preferences

The value of a preference can come from a variety of sources. There is a hierarchy to these sources, and IDL will use the value from the source with the highest priority.

Preferences specified at the command line when launching IDL have the highest priority, followed by preferences specified in environment variables. If a preference takes its value from either of these sources, you will not be able to change the preference’s value during the course of the IDL session, and the value will be desensitized in the Preferences dialog.

See “Understanding Preference Sources” (Appendix E, IDL Reference Guide) for additional information about the hierarchy of preference sources.
Various settings for the IDL Development Environment can be customized using the Preferences dialog. To open the Preferences dialog, select Preferences from the IDL Development Environment File menu.

**Note**

On UNIX platforms, including Macintosh OS X, some settings can also be customized by editing IDL’s resource files. For further information about editing resource files on UNIX and Macintosh OS X, see Chapter 6, “Customizing IDL on Motif Systems”.

The Preferences dialog contains tabbed sections that allow you to customize your interaction with the IDLDE. The tabs and their uses are described below.

**Note**

The terminology used on the Preferences dialogs differs between Microsoft Windows and Motif systems. In this documentation, if the wording is significantly different between the two platforms, the wording used in the Windows dialogs is listed first, followed by the wording used in the Motif dialogs.

<table>
<thead>
<tr>
<th>Tab</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Preferences</td>
<td>This tab allows you to specify how the IDLDE session begins and ends, to control the number of lines in the recall buffer and the Output Log, and to designate how the files should be opened and read.</td>
</tr>
<tr>
<td>Layout Preferences</td>
<td>This tab allows you to specify the location and size of the main IDLDE window on the screen. You can also designate which components of the IDLDE will be visible.</td>
</tr>
<tr>
<td>Graphics Preferences</td>
<td>This tab allows you to set the layout of windows that contain IDL graphics, and to specify the backing store, the size of the TrueType font cache, and the object graphic rendering preference.</td>
</tr>
<tr>
<td>Editor Preferences</td>
<td>This tab allows you to customize the IDL’s built-in editor and also offers several compiling options.</td>
</tr>
</tbody>
</table>

*Table 4-1: Preference Dialog Tabs*
Chapter 4: Setting IDL Preferences

Customizing IDL

Platform Differences

Microsoft Windows and UNIX platforms (including Macintosh OS X) implement the Preferences dialog using different dialog application buttons. The following table lists the buttons, the platforms on which they are found in the Preferences dialog, and the action performed when the button is used.

<table>
<thead>
<tr>
<th>Platform</th>
<th>Button</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows, UNIX</td>
<td>OK</td>
<td>Changes are saved and applied to the current session, and the Preferences dialog is dismissed.</td>
</tr>
<tr>
<td></td>
<td>Cancel</td>
<td>Any changes that were not applied are ignored, and the Preferences dialog is dismissed.</td>
</tr>
<tr>
<td></td>
<td>Apply</td>
<td>Changes are applied to the current session, but not saved. (On UNIX, changes to items marked on the dialog with an asterisk take effect in the next session. To make the changes for the current session, use OK.) The Preferences dialog remains visible.</td>
</tr>
<tr>
<td></td>
<td>Help</td>
<td>Displays IDL online help.</td>
</tr>
<tr>
<td>Windows only</td>
<td>Reset</td>
<td>Restores the preferences on the dialog to the preference values from the start of the current IDL session.</td>
</tr>
</tbody>
</table>

Table 4-2: Preferences Dialog Button Descriptions
Chapter 4: Setting IDL Preferences

General Preferences

The General tab of the Preferences dialog has three sections: Program, Log and Command Window, and Files.

![Figure 4-1: General Preferences Dialog]

**Note**

Some preference settings may be desensitized. See “Unavailable Preferences” on page 92 for details.

Program Section

You can specify how IDL handles starting up and exiting. Click on the following check boxes to apply or disable the options:

- **Show Splash Screen** — Select this option to show the IDL splash screen on startup. This selection takes effect the next time an IDL session is started. This control sets the value of the IDL_WDE_SPLASHSCREEN preference (Windows) and the IDL_MDE_SPLASHSCREEN preference (UNIX). For more information, see Appendix E, “IDL Preferences” (IDL Reference Guide).

- **Confirm Exit** — Select this option to display a warning dialog when you exit IDL.
Chapter 4: Setting IDL Preferences

General Preferences

This control sets the value of the `IDL_WDE_EXIT_CONFIRM` preference (Windows) and the `IDL_MDE_EXIT_CONFIRM` preference (UNIX). For more information, see Appendix E, “IDL Preferences” (*IDL Reference Guide*).

**Log and Command Window Section**

**Note**

On Microsoft Windows systems, these preferences are divided between the Log Window and Command Recall Buffer sections of the dialog.

The number of lines saved in the recall buffer for the Command Line has an impact on the performance of IDL. The amount of memory required for greater numbers of saved lines in the buffer affects the speed at which IDL runs. Click in the field next to each description and enter your adjusted value to change the settings.

- **Number of lines to display in the log / Lines to Save** — This field controls the maximum number of lines retained by the Output Log window. The default is 1000 lines.

  This control sets the value of the `IDL_WDE_LOG_LINES` preference (Windows) and the `IDL_MDE_LOG_LINES` preference (UNIX). For more information, see Appendix E, “IDL Preferences” (*IDL Reference Guide*).

- **Number of log lines to delete at limit / Delete on Limit** — This field controls the number of lines that will be deleted from the Output Log window when the maximum number of lines is reached. The earliest lines in the log are deleted. The default is 100 for Microsoft Windows systems and 250 for UNIX systems.

  This control sets the value of the `IDL_WDE_LOG_TRIM` preference (Windows) and the `IDL_MDE_LOG_TRIM` preference (UNIX). For more information, see Appendix E, “IDL Preferences” (*IDL Reference Guide*).

- **Number of lines saved in the recall buffer** — This field controls the maximum number of lines saved in the recall buffer. (See “Recalling Commands” (Chapter 2, *Application Programming*) for information on using the recall buffer.) The default is 20 lines.

  This control sets the value of the `IDL_RBUF_SIZE` preference. For more information, see Appendix E, “IDL Preferences” (*IDL Reference Guide*).

- **Save Recall Buffer Between Sessions** — Select this option to have the recall buffer persist between IDL sessions.

  This control sets the value of the `IDL_RBUF_PERSIST` preference. For more information, see Appendix E, “IDL Preferences” (*IDL Reference Guide*).
Files Section

You can change the way in which IDL handles opening files. Select or clear the following check boxes to apply or disable the options:

- **Change Directory on Open** — Select this option to cause IDL to change the current working directory when you open a file. The new current working directory will be the directory that contains the opened file.

  This control sets the value of the `IDL_WDE_EDIT_CWD` preference (Windows) and the `IDL_MDE_EDIT_CWD` preference (UNIX). For more information, see Appendix E, “IDL Preferences” (IDL Reference Guide).

- **Open Files Read Only** — Select this option to open files so that they can be viewed, but not changed.

  This control sets the value of the `IDL_WDE_EDIT_READONLY` preference (Windows) and the `IDL_MDE_EDIT_READONLY` preference (UNIX). For more information, see Appendix E, “IDL Preferences” (IDL Reference Guide).
Chapter 4: Setting IDL Preferences

Layout Preferences

This tab allows you to control the appearance and placement of the IDLDE.

Figure 4-2: Layout Preferences Dialog

Note

Some preference settings may be desensitized. See “Unavailable Preferences” on page 92 for details.

Main Window Section

Use the fields in this section to specify the default size and placement of the IDL Development Environment’s main window. (See “Components of the IDLDE” in Chapter 3 for descriptions of the components of the IDLDE.)

- Select the Default Layout radio button to use the IDLDE’s default layout, which depends on the size and resolution of your computer screen. If you select this radio button, all of the IDLDE’s windows and toolbars will be displayed in their standard locations.

- Select the Specify Layout radio button to manually specify the layout of the IDLDE:
  - Enter the number of pixels from the left-hand edge of the screen the IDLDE window should be displayed in the Left field.
Chapter 4: Setting IDL Preferences

- Enter the number of pixels from the top edge of the screen the IDLDE window should be displayed in the **Top** field
- Enter the width of the IDLDE window in pixels in the **Width** field
- Enter the height of the IDLDE window in pixels in the **Height** field

Note that if you select the **Default Layout** radio button after specifying values in these fields, your values will be replaced with “-1” to indicate that the default values will be used the next time IDL starts.

- Select the windows and toolbars to be displayed from the **Show Window** section (Windows) or **Windows** and **Control Panel** sections (Motif)

Click **Apply** to apply your changes to the current IDLDE window without saving the values. (This allows you to use the **Layout** tab to control the appearance of the IDLDE for the current session without making your changes permanent.) Click **OK** to apply your changes and save the values; they will be used the next time IDL starts.

- Select the **Remember Layout** radio button and click **OK** to save the current layout of the IDLDE windows for use the next time IDL starts. This option is useful if you have configured the windows manually and wish to save your changes.

**Undocking IDLDE windows**

Some of the elements of the IDLDE can be “undocked” from the interface and appear as separate, free-floating windows. On Microsoft Windows systems, use the mouse to select an element and drag it away from the main IDLDE window to undock the element. On UNIX systems, you can use the checkboxes in the **Windows** section to undock elements. For more information, see “Docking/Undocking” in Chapter 3.

The following elements can be undocked:

- Command Line
- Toolbars
- Output Log
- Variable Watch Window
- Project Window
Chapter 4: Setting IDL Preferences

Show Window Section (Windows Only)

By default, all the listed options are checked, signifying that they are all visible in the IDLDE main window. Click on the check boxes to show or hide the following windows:

- Command line
- Output Log window
- Status Bar
- Variable Watch window
- Standard Toolbar
- Run & Debug Toolbar
- Macros Toolbar
- Project window

Click **Apply** to apply your changes to the current IDL session. (This is the same as selecting the corresponding options in the **Window** menu.)

Windows Section (Motif Only)

Use the options in this section to control the appearance of the window elements of the IDLDE.

- **Editor Layout** — Click **Multiple** to display open Editor and Project windows separately from the main IDLDE window. Note that if the **Multiple Windows** option is enabled, the choice to hide or view the Editor windows is not available.

- **Hide** — Select the check box for elements of the IDLDE you wish to hide from view. By default, none of the sections are hidden.
  - **Control** hides the toolbars;
  - **View** hides the Project window and the Editor window;
  - **Log** hides the Output Log window;
  - **Watch** hides the Variable Watch window;
  - **Command** hides the Input Command Line;
  - **Status** hides the fly over status line at the base of the Main IDL window;
  - **Project** hides the Project window and extends the Editor window to the full width of the IDLDE.
Chapter 4: Setting IDL Preferences

- **Separate** — Select the check box for the constituent window you want to separate from the IDLDE Main Window. When the **Separate** action is applied, the element is “undocked” from the interface and appears as separate, free-floating window.

Click **Apply** to apply your changes to the current IDL session. (This is the same as selecting the corresponding options in the Window menu.)

**Control Panel Section (Motif Only)**

You can specify how you would like to display the various toolbars on the Control Panel.

- **Hide Tools** — Select the check box for any of the available toolbars (Standard, Run & Debug, and User) to hide that toolbar.

- **Number of Rows** — Enter the number of rows to use in displaying any visible toolbars. You can select from 1 to 3 rows.

- **Vertical** — Select this check box to cause the toolbars to be stacked vertically one on top of the other rather than horizontally next to each other.
Chapter 4: Setting IDL Preferences

Graphics Preferences

This tab allows you to control the layout and default size of IDL graphics windows. You can also control IDL's default use of backing store and the size of the TrueType font cache. Note that the values set here are defaults; the values can be overridden when a graphics window is created.

![Graphics Preferences Dialog](image)

**Figure 4-3: Graphics Preferences Dialog**

**Note**

Some preference settings may be desensitized. See "Unavailable Preferences" on page 92 for details.

**Window Layout / Windows Size Section**

Specify the default width and height of IDL graphics windows in the **Width** and **Height** fields. These controls set the values of the `IDL_GR_WIN_HEIGHT` and `IDL_GR_WIN_WIDTH` preferences (Windows) and the `IDL_GR_X_HEIGHT` and `IDL_GR_X_WIDTH` preferences (UNIX). For more information, see Appendix E, “IDL Preferences” (IDL Reference Guide).

Alternatively, you can specify that graphics windows have a default width and height of half the screen width and height by checking the **1/4 Screen Size** checkbox. This control sets the value of the `IDL_GR_WIN_QSCREEN` preference (Windows) and...
Chapter 4: Setting IDL Preferences

the \texttt{IDL\_GR\_X\_QSCREEN} preference (UNIX). For more information, see Appendix E, “IDL Preferences” \textit{(IDL Reference Guide)}.

**Platform Differences**

On Windows systems, you can specify that graphics windows should be created side-by-side, with no overlap by selecting the \textit{Tile} radio button, or that they should be created overlapping by selecting the \textit{Cascade} radio button. This control sets the value of the \texttt{IDL\_GR\_WIN\_LAYOUT} preference. For more information, see Appendix E, “IDL Preferences” \textit{(IDL Reference Guide)}.

Select the \textit{Always On Top} checkbox to ensure that graphics windows float above all other IDL windows. This control sets the value of the \texttt{IDL\_GR\_WIN\_ONTOP} preference. For more information, see Appendix E, “IDL Preferences” \textit{(IDL Reference Guide)}.

**Backing Store Section**

When backing store is enabled, a copy of each Graphics window is kept in memory; the copy is used to refresh the window when it has been covered and uncovered. IDL’s performance may increase when no backing store is used, since the amount of memory required to save copies can affect the speed at which IDL will run. Settings in this section correspond to settings of the \texttt{RETAIN} keyword to the \textit{DEVICE} procedure; see “Backing Store” \textit{(Appendix A, IDL Reference Guide)} for more information.

- \textit{None} (\texttt{RETAIN = 0}): Select this option to refrain from keeping a copy of the window. In some situations, disabling backing store may lead to an increase in IDL’s performance.
- \textit{System} (\texttt{RETAIN = 1}): Select this option to request backing store from the windowing system. This is the default.
- \textit{Bitmap / Pixmap} (\texttt{RETAIN = 2}): Select this option to specify that IDL should maintain the backing store using its own memory.

**Note**

Backing Store preference changes do not take effect until the next IDL session.

This control sets the value of the \texttt{IDL\_GR\_WIN\_RETAIN} preference (Windows) and the \texttt{IDL\_GR\_X\_RETAIN} preference (UNIX). For more information, see Appendix E, “IDL Preferences” \textit{(IDL Reference Guide)}.
True Type Fonts Section

Note

On UNIX systems, this preference is included in the Graphics Attributes section of the dialog, described below.

IDL saves TrueType fonts as a set of glyphs; each glyph represents the triangulation data for drawing one character. The Size of TrueType Font Cache (in glyphs) field allows you to set the number of glyphs to keep in cache memory; keeping glyphs in memory speeds drawing of fonts in IDL graphics windows. The default number of glyphs in cache memory is 256, roughly two TrueType font sets.

Enter the number of TrueType characters for which to save triangulation information. Saving the triangulation information for TrueType characters means that IDL will not have to calculate the polygons to draw the next time a character of the same font and size is rendered. Larger values will use more memory but can increase drawing speed if multiple fonts are used. The default is 256.

This control sets the value of the IDL_GR_TTCACHESIZE preference. For more information, see Appendix E, “IDL Preferences” (IDL Reference Guide).

Default Object Graphics Renderer / Graphics Attributes Section

IDL supports two methods of rendering object graphics: via a hardware graphics accelerator or via a software rendering package. Select Hardware rendering if your system has OpenGL graphics accelerator hardware. Select Software rendering otherwise. This control sets the value of the IDL_GR_WIN_RENDERER preference (Windows) and the IDL_GR_X_RENDERER preference (UNIX). For more information, see Appendix E, “IDL Preferences” (IDL Reference Guide).

See “Hardware vs. Software Rendering” (Chapter 2, Object Programming) for information about the differences between the two rendering systems.
Chapter 4: Setting IDL Preferences

Editor Preferences

This tab allows you to specify settings for the built-in IDL Editor and control the way IDL compiles files loaded in editor windows. On Microsoft Windows systems, this tab also allows you to specify syntax-highlighting and other editor features.

Note

Some preference settings may be desensitized. See “Unavailable Preferences” on page 92 for details.

Backup on Save

Select the **Backup on Save / Make backup copy of source file** check box to cause IDL to create a backup of the original file when saving a file in an IDL editor window.

This control sets the value of the IDL_WDE_EDIT_BACKUP preference (Windows) and the IDL_MDE_EDIT_BACKUP preference (UNIX). For more information, see Appendix E, “IDL Preferences” (*IDL Reference Guide*).
Syntax Highlighting (Windows Only)

On Microsoft Windows systems, you can choose to use *syntax highlighting* in IDL editor windows. If syntax highlighting is turned on, IDL statements are displayed in different colors. Select the **Enable colored syntax** checkbox to enable syntax highlighting. This control sets the value of the `IDL_WDE_EDIT_CHROMACODE` preference. For more information, see Appendix E, “IDL Preferences” (*IDL Reference Guide*).

Open on Debug (Windows Only)

If you want IDL to open the source file for a program that generates an error in an IDL editor window, select the **Enable Open on debug** checkbox. This control sets the value of the `IDL_WDE_EDIT_OPEN_ON_DEBUG` preference. For more information, see Appendix E, “IDL Preferences” (*IDL Reference Guide*).

Compiling Section

Select the **Ask to save changes before compiling** radio button if you would like to save changes when you compile a program in an IDL editor window. This is the default.

Select the **Automatically save changes before compiling** radio button if you do not want to be prompted each time you compile, but do want to save the changes.

Select the **Compile from memory (don’t save before compile)** radio button if you do not want to save files before compiling them.

**Note**

You can override your default selection by selecting the appropriate menu item from the Run menu.

This control sets the value of the `IDL_WDE_EDIT_COMPILE_OPTION` preference (Windows) and the `IDL_MDE_EDIT_COMPILE_OPTION` preference (UNIX). For more information, see Appendix E, “IDL Preferences” (*IDL Reference Guide*).

Tabs Section (Windows Only)

You can specify the width of the white space to be used when you press the **Tab** key in an IDL editor window. Enter a number in the **Number of spaces to indent for each tab** field to specify the width of the indent to be used. This control sets the value of the `IDL_WDE_EDIT_TAB_WIDTH` preference. For more information, see Appendix E, “IDL Preferences” (*IDL Reference Guide*).
Chapter 4: Setting IDL Preferences

If you want the IDL editor to insert a tab character (ASCII 9) when you press the Tab key, select the Use tabs radio button. If you want IDL to insert the specified number of space characters (ASCII 32) when you press the Tab key, select the Use spaces radio button. This control sets the value of the IDL_WDE_EDIT_TAB_ENABLE preference. For more information, see Appendix E, “IDL Preferences” (IDL Reference Guide).

If you have selected the Use spaces radio button, you have the option to convert tab characters to spaces when the file is saved by selecting the Convert tabs to spaces on save checkbox. This control sets the value of the IDL_WDE_EDIT_TAB_SP_ON_SAVE preference. For more information, see Appendix E, “IDL Preferences” (IDL Reference Guide).

Colors Section (Windows Only)

Use this section to select the colors that will be used in the IDL editor when syntax highlighting is enabled. To set colors, select a type of IDL statement from the scrolling listbox at left, then select the foreground and background colors for that type of statement.

These controls set the values of the IDL_WDE_EDIT_BCOLOR_* and IDL_WDE_EDIT_FCOLOR_* preferences. For more information, see “IDL_WDE_EDIT_[B|F]COLOR_*” (Appendix E, IDL Reference Guide).
Startup Preferences

This tab allows you to specify the locations of the default working directory and any startup file to be run.

![Startup Preferences Dialog](image)

Figure 4-5: Startup Preferences Dialog

Note

Some preference settings may be desensitized. See “Unavailable Preferences” on page 92 for details.

Working Directory

This field allows you to set the initial working directory for future IDL sessions. The General Preferences tab contains a “Change Directory on Open” option, which also affects the working directory.

This control sets the value of the IDL_WDE_START_DIR preference (Windows) and the IDL_MDE_START_DIR preference (UNIX). For more information, see Appendix E, “IDL Preferences” (IDL Reference Guide).
Chapter 4: Setting IDL Preferences

Startup File

Use this field to specify the name of an IDL batch file to be executed automatically each time IDL is run. See “Startup Files” on page 29 for additional details.

This control sets the value of the IDL_STARTUP preference. For more information, see Appendix E, “IDL Preferences” (IDL Reference Guide).
Font Preferences

This tab allows you to specify fonts to be used in various sections of the IDLDE interface.

Note

Some preference settings may be desensitized. See “Unavailable Preferences” on page 92 for details.

Microsoft Windows

Under Microsoft Windows, IDL uses a standard Windows font-selection dialog. You can select different fonts for IDL Editor windows, the Command Line, and the Output Log. Click on one of these areas in the Window list, then select the font, style, and size using the appropriate lists. Click Use Default Fonts to change to the IDL default font selections for all three areas.

These controls set the values of the IDL_WDE_EDIT_FONT, IDL_WDE_INPUT_FONT, and IDL_WDE_LOG_FONT preferences. For more information, see Appendix E, “IDL Preferences” (IDL Reference Guide).
UNIX

This tab allows you to control which fonts are to be used for the main IDL window. Click on any of the following buttons to specify the relevant font:

- **Default** — dialog boxes
- **Menubar** — menu items
- **Control** — the Control Panel
- **Edit** — editor windows
- **Log** — the Output Log
- **Command** — the Command Line

**Selecting a Font**

Clicking any of the buttons on the **Fonts** tab of the Preferences dialog brings up the **Select Font** dialog. This dialog allows you to select fonts from the X Windows Server font database, based on the attributes **Foundry**, **Family**, **Weight**, **Slant**, **SetWidth**, and **Size**. Using this dialog is similar to using the **xfontsel** X Window...
utility. See your X Window system font documentation for additional details. Once you have selected a font, click **OK** to accept your selection or **Cancel** to abandon it.

![Motif Select Font Dialog](image)

**Figure 4-7: Motif Select Font Dialog.**

**Note**
Path Preferences

This tab allows you to control where IDL looks for procedures and functions. The path elements specified in the Search Path / IDL Files Search Path are used to set the IDL_PATH preference and the !PATH system variable.

**Note**
Some preference settings may be desensitized. See “Unavailable Preferences” on page 92 for details.

**Search Path / IDL Files Search Path**

The IDLDE Path Preferences dialog uses the same mechanism to expand the elements of the Search Path field as is used by the EXPAND_PATH function. By default, this field is populated with the current value of the IDL_PATH preference. For more information, see Appendix E, “IDL Preferences” (IDL Reference Guide).

**Note**
If you have not changed the value of the IDL_PATH preference, it contains a single entry (<IDL_DEFAULT>) indicating that the default IDL path will be used. See “The Path Definition String” under “EXPAND_PATH” (IDL Reference Guide) for complete details on how this token is expanded.
Chapter 4: Setting IDL Preferences

Path Preferences

If the box to the left of a path element is checked, all directories below the listed directory that contain at least one .pro or .sav file will be included in !PATH. (This mechanism is analogous to the use of a "+" symbol in an EXPAND_PATH path definition string.)

Note

If the <IDL_DEFAULT> entry is present, the box to its left is both checked and greyed out (Windows) or completely blacked out (Motif), indicating that the token will always be expanded.

You can modify the value of the !PATH system variable in the following ways using this dialog:

- **Change the order of the path elements** — using the up- and down-arrows, you can reorder the path elements. When searching the directories in the !PATH system variable for files, IDL will use the first matching file it finds. If you have multiple files with the same name in different directories within !PATH, you may need to adjust the order in which the directories are scanned.

- **Insert...** — To add a path to the Search Path list, click Insert... to display the Select Directory dialog. The new path is inserted before the first selected path. If none of the paths are selected, the new path is appended to the end of the list.

- **Insert Standard Libraries** — Click Insert Standard Libraries to insert the <IDL_DEFAULT> path element into the list.

- **Remove** — Click on Remove to delete the selected path.

- **Expand** — Click on Expand to include the individual subdirectories of the selected path element in the Search Path list. When you click Expand, the checkmark is removed from the original path element, since the subdirectories are now explicitly included in the path search list.

See “Automatic Compilation” (Chapter 2, Application Programming) for more information on how !PATH is used by IDL when compiling and running programs.

Enable Path Cache

Select Enable Path Cache to enable IDL’s path caching mechanism. Path caching is enabled by default, and in almost all cases should be left enabled. See “PATH_CACHE” (IDL Reference Guide) for more information about IDL’s path cache.

This control sets the value of the IDL_PATH_CACHE_DISABLE preference. For more information, see Appendix E, “IDL Preferences” (IDL Reference Guide).
Chapter 5
Creating Development Environment Macros

This chapter discusses the following topics:

What Are Macros? .................... 116
Creating UNIX Macros .............. 117
Creating Windows Macros .......... 120
Command Stream Substitutions .... 122
Building IDL Example Macros .... 123
What Are Macros?

A macro allows you to execute commonly-used IDL tasks with the press of a mouse button or through a single keystroke (“hot key”) combination. In IDL you can create your very own macros using the following items:

- routines
- procedures
- statements
- command stream substitutions

For example, you may customize and extend the functionality of the IDL Development Environment (such as writing a procedural macro to change IDL’s working directory, which we will see later in this section).

Predefined IDL Macros

IDL offers several existing macro options on its Macro Toolbar. These macros allow you quick access to commonly used IDL functionality such as printing a variable, importing various file types, and running the IDL Demos.

See “Using IDL Macros” (Chapter 6, *Using IDL*) for more information.
Chapter 5: Creating Development Environment Macros

Creating UNIX Macros

You can modify the contents of the Macros menu and macros toolbar, either using the Edit Macros dialog (displayed by selecting Edit... from the Macros menu) or by manually editing the user resource (.idlde) file.

Using the Edit Macros Dialog

The Edit Macros dialog allows you to add, remove, or modify macros that appear either in the Macros menu or the Macros toolbar.

To add a new macro, do the following:

1. Enter a name for your macro in the Name field. The Name appears only in the Edit Macros dialog.
2. Enter a label for your macro in the Label field. The label will be used in the Macros menu (if selected).
3. Enter the name of the bitmap (.xbm or .xpm) file associated with the macro in the Bitmap field. The bitmap will be used on the Macros toolbar (if selected). See “Bitmaps for Control Panel Buttons” on page 118 for details.
Creating UNIX Macros

1. Enter text to be displayed on the IDLDE status bar in the **Status bar text** field.
2. Enter text to be displayed as a tooltip when the mouse cursor is positioned over the toolbar button in the **Tip text** field.
3. Enter the IDL command to be executed in the **IDL Command** field. See “Command Stream Substitutions” on page 122 for information on the types of dynamic information that can be included in the command.
   
   In addition to IDL-language commands, you can attach IDL Motif Action Routines to a macro. See “Action Routines” on page 137 for details.
4. Select the **Menu** checkbox if you want the macro to appear on the **Macros** menu.
5. Select the **Toolbar** checkbox if you want the macro to appear on the Macros toolbar.
6. Click **Add** to add the new macro, then click **OK**. To remove an existing macro, select it from the list and click **Remove**. To rearrange macros in the list, use the up- and down-arrow buttons.

**Bitmaps for Control Panel Buttons**

It is recommended that bitmaps for control panel buttons:

1. Be in either XBM (X11 bitmap file) or XPM (X11 system pixmap file) format, with the file extension .xbm or .xpm.
2. Supply the full path name to the bitmap file. Alternatively, if the bitmap is located in one of the following directories, you can supply only the base file name:
   - $IDL_DIR/resource/X11/lib/app_defaults
   - $IDL_DIR/resource/X11/lib/app_defaults/bitmaps
   - $HOME
   - $HOME/bitmaps

**Note**

The above directories show the default search path for a bitmap file if nothing other than the root file name is specified in the .idlde file.
Chapter 5: Creating Development Environment Macros

Manually Editing the Resource File

Although there is little advantage in doing so, you can also modify the Macros menu or toolbar by manually editing either your own local IDL resource file or the system-wide resource file. For details, see “Modifying the Control Panel” on page 134.
Creating Windows Macros

You can modify the contents of the Macros menu and macros toolbar using the Edit Macros dialog (displayed by selecting Edit... from the Macros menu). The Edit Macros dialog allows you to add, remove, or modify macros that appear either in the Macros menu or the Macros toolbar.

To add a new macro, do the following:

1. Click Add and enter a name for your new macro. The name you specify appears only in the Edit Macros dialog.

2. Enter the IDL command to be executed in the IDL Command field. See “Command Stream Substitutions” on page 122 for information on the types of dynamic information that can be included in the command.

3. If you want your macro to be included in the Macros menu, enter a label for your macro in the Menu Item Name field.

4. If you want your macro to be included in the Macros toolbar, enter the full path name of the bitmap button file in the Toolbar bitmap file field. Bitmaps used as macro buttons in IDL must be 16 by 16 pixel .bmp files. IDL’s default
Chapter 5: Creating Development Environment Macros

bitmaps are stored in the resources/bitmaps subdirectory of the IDL distribution.

5. Enter text to be displayed as a tooltip when the mouse cursor is positioned over the toolbar button in the **Tooltip text** field. This value is ignored if no bitmap file is specified.

6. Enter text to be displayed on the IDLDE status bar in the **Status bar text** field.

7. Optionally, in the **Accelerator** field, enter a keystroke shortcut combination for your new macro. Note that you can create a macro that is available only by pressing the keystroke combination if you supply neither a label for the **Macros** menu nor a bitmap for the Macros toolbar.

To Remove an existing macro, select it from the list and click **Remove**. To rearrange macros in the list, use the up- and down-arrow buttons.

Click **OK** to accept your changes or **Cancel** to abandon them.
Command Stream Substitutions

You can use command stream (%) substitutions as shortcuts to incorporate certain types of information into the IDL command for your macro.

<table>
<thead>
<tr>
<th>Command Stream Substitution</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>%F</td>
<td>The filename associated with the currently active editor window.</td>
</tr>
<tr>
<td>%P</td>
<td>The full path filename associated with the currently active editor window.</td>
</tr>
<tr>
<td>%N</td>
<td>The base name of the filename without its path or suffix.</td>
</tr>
<tr>
<td>%B</td>
<td>The base name of the filename without its path, but with its suffix.</td>
</tr>
<tr>
<td>%S</td>
<td>The currently selected text.</td>
</tr>
<tr>
<td>%L</td>
<td>The line number with the current insertion point.</td>
</tr>
<tr>
<td>%%</td>
<td>Inserts the “%” character.</td>
</tr>
</tbody>
</table>

Table 5-1: Listing of Useful Command Stream Substitutions

Note
When creating a new macro, you may store the macro in the folder (directory) which IDL has already provided for the existing IDLDE macros. This folder exists in the lib\macros directory of your installation directory. If you wish to create a unique folder for the storage of only macros which you have created you may do so.
Building IDL Example Macros

Below are two examples that illustrate how a macro is created in IDL. The first example below is a UNIX-only example; the second example will work on either Microsoft Windows or UNIX.

Creating a Macro to Call a Text Editor in IDL for UNIX

On UNIX platforms, you can create a macro to open a file that is currently open in the IDL Editor in another editor, such as emacs or vi. Use the following procedure to create the macro:

1. Select Macros → Edit menu to bring up the Edit Macros dialog box. You can use this dialog to create, edit, or remove macros.

2. Complete the fields in the Edit Macros dialog:
   - Name: The name that you wish to appear in the Macros list in the Edit Macros dialog. For example, enter Edit in emacs.
   - Label: The name that you wish to appear on the Macros menu. For example, enter emacs.
   - Bitmap: The bitmap to use as the toolbar button label. Use the file paths and file name extensions discussed in “Bitmaps for Control Panel Buttons” in Chapter 5.
   - Status bar text: The text that appears in the status bar when the mouse is help over the menu item or toolbar button.
   - Tip text: The text for the tool tip that appears when the mouse is held over the toolbar button.
   - IDL command: The IDL command to execute when the macro is selected. To create a macro for editing in Emacs, enter the following:

```
SPAWN, 'emacs +%L %P &'
```

   Select the Menu and/or Toolbar checkbox to specify whether the macro will appear in the Macros menu and/or the toolbar.

3. Create the new macro by pressing the Add button. If you entered emacs in the Label field, a new “emacs” macro is added to the Macros list.

4. To add a macro for editing in vi, repeat the above steps, but enter the following in the “IDL command” field:

```
SPAWN, 'xterm -e vi +%L %P &'
```
Creating a Macro to Change the Working Directory

The following macro will select and change your current working directory. The steps below describe the fields of the Macros dialog on a Microsoft Windows system, but the macro will work equally well on a UNIX system.

First we will create a .pro file in IDL which will display a platform-specific directory-selection dialog.

1. From the IDLDE, open a new IDL Editor window by selecting **File → New → Editor**.

2. Type (or copy) the following lines of code into the new Editor window to form a program:

   ```idl
   PRO cd_test
   dir = DIALOG_PICKFILE(/DIRECTORY)
   IF (dir) THEN BEGIN
     PRINT, 'Changing to: ', dir
     CD, dir
   ENDIF
   END
   ```

3. Save the file as **cd_test.pro** in a directory included in IDL’s path. (The file must be in IDL’s path so that IDL will find it automatically when the command cd_test is executed by the macro we will create.)

4. Select **Macros → Edit** menu to bring up the Edit Macros dialog box.

5. Click **Add** to create a new macro. Enter “Change Directories” as the macro name.
6. Complete the following fields in the Edit Macros dialog:
   • Enter “cd_test” in the **IDL command** field.
   • Enter “Change Directories” in the **Menu item name** field.
   • Leave the **Toolbar bitmap file** field blank. This macro will appear only in the **Macros** menu.
   • Leave the **Tooltip text** field blank. This value is used only when a toolbar button is present.
   • Leave the **Status bar text** field blank. This value is used only when a toolbar button is present.

   To use the new macro, select “Change Directories” from the **Macros** menu.
Chapter 6
Customizing IDL on Motif Systems

This chapter describes techniques for customizing versions of IDL running under the X Window System (Motif) graphical user interface.

Using X Resources to Customize IDL . . . 128
X Resources at the Command Line . . . 132
Modifying the Control Panel . . . . . 134
Action Routines . . . . . . . . . . . . . . . . . . . . 137
Chapter 6: Customizing IDL on Motif Systems

Using X Resources to Customize IDL

IDL on UNIX platforms respects the values of a number of X Window System (Motif) resources.

X Resources and IDL Preferences

Beginning with IDL 6.2, many values used to customize the appearance and behavior of IDL on UNIX platforms are stored in IDL preferences rather than in X resources. See Appendix E, “IDL Preferences” (IDL Reference Guide) for a detailed description of IDL’s preferences system.

To provide backwards compatibility with older versions, current versions of IDL are still able to check the values of X resources set in the user’s .idlde or .Xdefaults files and transfer them, if found, to the corresponding IDL preference setting. The mechanism used is described in detail in “Support for Obsolete Preference Mechanisms” (Appendix E, IDL Reference Guide).

Not all X resources have corresponding preference values. Generally speaking, the X resource values that have not been implemented as preferences either control aspects of the appearance of the IDL Development Environment or define user macros. These values may become IDL preferences in a future version of IDL.

X Resources in Brief

The component widgets of an X Window System application each have two names, a class name that identifies its type (e.g., XmText for the Motif text widget) and an instance name (e.g., command, the name of the IDLDE command input text widget). The class name can be used to set resources for an entire class of widgets (e.g., to make all text widgets have a black background) while the instance name is used for control of individual widgets (e.g., set the IDLDE command input window font without affecting other widgets).

Applications consist of a tree of widgets, each having a class name and an instance name. To specify a resource for a given widget, list the names of the widgets lying between the top widget and the target widget from left to right, separated by periods. In a moderately complicated widget hierarchy, only some of the widgets are of interest; there are intervening widgets that serve uninteresting purposes (such as a base that holds other widgets). A star (*) character can be used as a wildcard to skip such widgets. Another fact to keep in mind is that a given resource specification is interpreted as broadly as possible to apply to any widget matching that description.
This allows a very small set of resource specifications to affect a large number of widgets.

Resource Files

There are two resource files used to customize the IDL Development Environment. An installation-wide resource file called `idl` is located in

$IDL_DIR/resource/X11/lib/app-defaults

and a user resource file called `.idlde` is located in your home directory.

Modifying the global `idl` resource file effects an installation-wide customization. Changes to the `idl` file are not migrated when a new version of IDL is installed.

The user resource file, `.idlde`, customizes individual versions of IDLDE and is divided into two sections. The first section contains user-defined customization resources. You can place comments starting with “!” or “!!” in the first section of `.idlde`. When newer versions of `.idlde` are written, system comments are prefixed with “!!!”. The second section of `.idlde` is used to store IDLDE preferences; it is modified when IDLDE preferences are modified via the Preferences tab of the Motif IDLDE, and should not be modified manually.

**Note**

IDLDE preferences saved in the `.idlde` file should not be confused with the IDL preference system included in IDL versions 6.2 and later. In some cases, values of the IDLDE preferences from the `.idlde` file are migrated to the newer preferences system; see “Support for Obsolete Preference Mechanisms” (Appendix E, *IDL Reference Guide*) for details.

If you use IDL in command-line mode rather than via the IDL Development Environment, you can include resources in the `.Xdefaults` file located in your home directory.

Format of IDL Resources

IDL resource strings begin with the characters “`Idl`”. Most of these resources have been superseded by preferences in the IDL preference system.

Resource strings that apply only to the IDL Development Environment begin with the characters “`idlde`” or “`idlde`”. For example, the resource

```
idlde*hideCommand
```

controls whether the IDLDE Command Line is visible when IDL starts up.
Resources that include the string “idlde” must be included either in the system-wide \texttt{Idl} resource file, or in a \texttt{.idlde} file in your home directory. Resources that apply to IDL whether it is running in command-line mode or via the IDLDE can be included in either the system-wide \texttt{Idl} resource file or in a \texttt{.Xdefaults} file in your home directory.

To specify a value for an \texttt{X} resource, append a colon character and the value after the resource string. Whitespace is ignored. For example:

\begin{verbatim}
idlde*hideCommand:False
\end{verbatim}

is the same as

\begin{verbatim}
idlde*hideCommand: False
\end{verbatim}

### X Resources Used by IDL

IDL uses a large number of resources to control the behavior and appearance of the IDL Development Environment and any graphical application written in IDL. To learn more about the specific resources used, or to modify individual values, inspect the installation-wide resource file \texttt{Idl}, located in

\begin{verbatim}
$IDL_DIR/resource/X11/lib/app-defaults
\end{verbatim}

\textbf{Note}

In order to maintain backward compatibility with previous versions of IDL, the \texttt{Idl} resource file contains values for some resources that correspond to IDL preferences. \textit{X resources that have been superseded by preferences are ignored by IDL.} See “Support for Obsolete Preference Mechanisms” (Appendix E, \textit{IDL Reference Guide}) for details.

\textbf{Tip}

We suggest that you use preferences rather than \texttt{X} resources when possible. If you must make changes to \texttt{X} resources, make the changes in a user-specific \texttt{.idlde} file or \texttt{.Xdefaults} file.

### Reserving Colors

If you use a PseudoColor display device, when IDL starts, it attempts to secure entries in the shared system color map for use when drawing graphics. If the entry \texttt{Idl.colors} exists in one of the \texttt{X} resource files inspected by IDL at startup, IDL will first migrate the specified value to the value of the \texttt{IDL\_GR\_X\_COLORS} preference, and then attempt to allocate the number of colors specified from the shared colormap. If for some reason it cannot allocate the requested number of colors
from the shared colormap, IDL will create a private colormap. Using a private colormap ensures that IDL has the number of colormap entries necessary, but can lead to colormap flashing when the cursor or window focus moves between IDL and other applications.

One way to avoid creating a private colormap for IDL is to set the \texttt{IDL\_GR\_X\_COLORS} preference equal to a negative number. This causes IDL to try to use the shared colormap, allocating all but the specified number of colors. For example, setting the preference value to $-10$ instructs IDL to allocate all but 10 of the currently available colors for its use. Thus, if there are a total of 220 colors not yet reserved by other applications (such as the windowing system), IDL will allocate 210 colors from the shared colormap.

The IDLDE application itself uses between 10-15 colors. On startup, the IDLDE will attempt to use colors in the shared colormap, but will reserve colors for itself if appropriate matching colors in the shared colormap are not found. As a result, running IDL with the IDLDE may use more colors than running IDL with the tty (plain command line) interface.

\textbf{Note}

If you use a TrueColor display device, IDL does not rely on the system’s shared color map when drawing graphics. There is no need to either reserve colors from the shared color map or create a private color map.
Chapter 6: Customizing IDL on Motif Systems

X Resources at the Command Line

The appearance of the UNIX IDLDE can also be customized from the command line using the command line flags described below. Command line flags are given precedence over global resource files (Idl) and user resource files (.idlde). For more information about resources, see “Using X Resources to Customize IDL” on page 128.

X Resource Command Line Switches

The following command line switches can be used to control the values of X resources when invoking IDL on UNIX platforms. Unless otherwise noted, switches can be combined, and can be specified in any order.

-nocommand

Hides the Output Log window and Command Line at startup. The related resource is Idl*idlde*hideCommand: True.

-command

Displays Log window and Command Input window at startup. The related resource is Idl*idlde*hideCommand: False.

-nocontrol

Hides the Control panel buttons at startup. The related resource is Idl*idlde*hideControl: True.

-control

Displays the Control Panel buttons at startup. The related resource is Idl*idlde*hideControl: False.

-nolog

Hides the Output Log at startup. The related resource is Idl*idlde*hideLog: True.

-log

Displays the Output Log at startup. The related resource is Idl*idlde*hideLog: False.
**-nostatus**

Hides the Status Bar at startup. The related resource is 
`Idl*idlde*hideStatus: True`.

**-status**

Displays the Status Bar at startup. The related resource is 
`Idl*idlde*hideStatus: False`.

**-single**

Displays files in a single window, which is a child of the main IDLDE window. The related resource is 
`Idl*idlde*multiWindowEdit: False`.

**-multi**

Displays files in multiple windows, each one in a separate main level window. The related resource is 
`Idl*idlde*multiWindowEdit: True`.

**-view**

Displays the Multiple Document Panel in single window mode at startup. The related resource is 
`Idl*idlde*hideView: False`.

**-noview**

Hides the Multiple Document Panel at startup. The related resource is 
`Idl*idlde*hideView: True`.

**-title "Title"**

Use *Title* as the title of the main IDLDE window. The related resource is 
`idlde.title`.
Modifying the Control Panel

The Control Panel, with the resource name control, is located below the IDL Development Environment Menu bar. The Control Panel bar is a RowColumn widget containing buttons which serve as shortcuts for common commands.

You can modify the existing Control Panel settings by editing the idlde*control values in the system-wide Idl resource file or overriding those settings in your local .idlde file. In addition, you can add buttons to the Macros toolbar or menu by adding resources to your .idlde file.

**Note**

If you wish to add, modify, or remove the buttons on the Macros toolbar or menu, you can do so via the IDLDE interface using the Edit Macros dialog. See “Creating UNIX Macros” on page 117 for details. Whether you modify your macros using the dialog or by editing a resource file manually, the results are the same. There is little advantage to adding macros to the .idlde file manually.

Adding Macros Toolbar Buttons

The idlButtonsUser resource defines the resource name for each button on the Macros toolbar in the Control Panel. The resource name details button attributes, such as its label or pixmap, its associated IDL command, and its status bar message.

To add a button to the Macros toolbar, make the following modifications to the .idlde file:

- Add a new name to the idlde*controlidlButtonsUser list. The buttons are created in the order specified.
- Add idlde*control*<new button>*labelString or labelPixmap resources (or both). These resources define the button text or image. If you choose to use a pixmap label, be sure the file you specify abides by the restrictions described in “Bitmaps for Control Panel Buttons” on page 118.
- Add an idlde*control*<new button>*idlCommand resource. This is the text of the IDL command to execute. You can also include command stream substitutions; see “Command Stream Substitutions” on page 122 for details.

Alternatively, you can add an idlAction resource. See “Action Routines” on page 137 for details.
Chapter 6: Customizing IDL on Motif Systems

- Add an `idlde*control*<new button>*hint` resource. This is the text that appears in the Status Bar when the cursor is positioned over the new button.

- Add an `idlde*control*<new button>*tip` resource. This is the text that appears as a “tooltip” when the cursor is positioned over the new button.

If you want your changes to be available to all users on the system, you can also modify the system-wide Idl resource file, located in the following directory:

$IDL_DIR/resource/X11/lib/app-defaults

Adding Macros Menu Entries

To add entries into the Macros menu, follow the same steps outlined above, modifying the `idlde*menubar*macrosMenu*macrosListUser` resource and substituting `idlde*menubar*macrosMenu*<new menu item>` for `idlde*control*<new button>` in the above steps.

Examples

To add a button called Reset All to the Control Panel with a color pixmap stored in the file `resetall.xpm` located in your home directory, add the following resources to the `.idlde` file in your `$HOME` directory:

```
idlde*control*idlButtonsUser: <exiting buttons> resetall
idlde*control*resetall*labelPixmap: resetall.xpm
idlde*control*resetall*labelString: Reset All
idlde*control*resetall*idlCommand:\n  RETALL & WIDGET_CONTROL,/RESET
idlde*control*resetall*statusString:\n  Stop execution of the current code and return to\n  the main programming level
```

Note that in this example the new button is added at the end of the list of existing buttons. You can locate the new button anywhere in the list.

To specify a pixmap located in particular directory, specify the full file path of the pixmap file, for example:

```
idlde*control*resetall*labelPixmap:\
/home/user/bitmaps/resetall.xpm
```

To create two rows of the Control Panel from the default of one row, set the `numColumns` resource to 2:

```
idlde*control*numColumns: 2
```
To use label (text) buttons in the Control Panel set `labelType` to `XmSTRING`. To use icon (graphics) buttons set `labelType` to `XmPIXMAP`.

```
idlde*control*labelType: XmSTRING
or
idlde*control*labelType: XmPIXMAP
```
Chapter 6: Customizing IDL on Motif Systems

Action Routines

Most Motif widgets supply action routines which can be bound to events (such as keypress events). Action routines provided by IDL can be used to define commands for Control Panel buttons or menu items by using the idlAction resource.

The following action routines can be used in the same manner as the IDL commands specified in an idlCommand resource. The syntax to add an action routine to a control panel button is:

```
Idl*idlde*control*buttonName*idlAction: Action
```

or

```
Idl*idlde*control*buttonName*idlAction: Action(Arguments)
```

where `buttonName` is the name of the button and `Action` is the name of the action routine. Arguments to the action routine, if required, are enclosed in parentheses.

**IdlBreakpoint**

Use `IdlBreakpoint` to control the placement of breakpoints. If no parameter is specified, the breakpoint is set on the current line. At least one of the arguments from the following table must be set:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET</td>
<td>Set a breakpoint on the current line.</td>
</tr>
<tr>
<td>CLEAR</td>
<td>Clear the breakpoint on the current line.</td>
</tr>
<tr>
<td>TOGGLE</td>
<td>Toggle (SET or CLEAR) the state of the breakpoint on the current line.</td>
</tr>
<tr>
<td>COMPLEX</td>
<td>Display breakpoint dialog to set a complex breakpoint.</td>
</tr>
<tr>
<td>LIST</td>
<td>List all currently set breakpoints</td>
</tr>
</tbody>
</table>

*Table 6-1: Breakpoint Arguments*

For example, to use this action routine to clear a breakpoint, the `Action` specified would be:

```
IdlBreakpoint (CLEAR)
```
Chapter 6: Customizing IDL on Motif Systems

**IdlClearLog**

Use `IdlClearLog` to erase the contents of the Output Log.

**IdlClearView**

Use `IdlClearView` to clear the contents of the currently-active file in the Multiple Document Panel.

**IdlCommandHide**

Use `IdlCommandHide` to hide or expose the Command Area, which includes the Command Line and the Output Log. One of the following arguments must be set: `Show`, `Hide`, or `Toggle`.

**IdlCompile**

Use `IdlCompile` to compile the file in the currently-active editor window. One of the arguments from the following table must be set:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILE</td>
<td>Compiles the currently-active file.</td>
</tr>
<tr>
<td>TEMPORARY</td>
<td>Compiles the currently-active file into a temporary file</td>
</tr>
<tr>
<td>RESOLVE</td>
<td>Resolves all referenced and uncompiled IDL routines</td>
</tr>
</tbody>
</table>

*Table 6-2: Compiling Arguments*

**IdlControlHide**

Use `IdlControlHide` to hide or expose the Control Panel. One of the following arguments must be set: `Show`, `Hide`, or `Toggle`.
Chapter 6: Customizing IDL on Motif Systems

**IdlEdit**

Use `IdlEdit` to manipulate the contents of the currently-selected editor window. One of the arguments from the following table must be set:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNDO</td>
<td>Undo previous editing action.</td>
</tr>
<tr>
<td>REDO</td>
<td>Redo previously undone action.</td>
</tr>
<tr>
<td>CUT</td>
<td>Remove currently-selected text to UNIX clipboard.</td>
</tr>
<tr>
<td>COPY</td>
<td>Copy currently-selected text to UNIX clipboard.</td>
</tr>
<tr>
<td>PASTE</td>
<td>Paste contents of UNIX clipboard at current insertion point.</td>
</tr>
<tr>
<td>SELECTALL</td>
<td>Select all of the text in the currently-selected editor window.</td>
</tr>
<tr>
<td>GOTODEF</td>
<td>Display the definition of the currently-selected procedure or function.</td>
</tr>
<tr>
<td>GOTOLINE</td>
<td>Move directly to the specified line number.</td>
</tr>
</tbody>
</table>

*Table 6-3: Editor Window Editing Arguments*

**IdlEditMacros**

Use `IdlEditMacros` to display the Edit Macros dialog.

**IdlExit**

Use `IdlExit` to cause IDLDE to act as though the EXIT command has been entered. Note that this is usually tied to a menu accelerator (`Ctrl-Q` in this case), so this routine is rarely called directly.
Chapter 6: Customizing IDL on Motif Systems

**IdlFile**

Use **IdlFile** to manipulate the currently-selected editor window. One of the arguments in the following table must be set:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEW</td>
<td>Creates a new editor window.</td>
</tr>
<tr>
<td>OPEN</td>
<td>Opens an existing file.</td>
</tr>
<tr>
<td>SAVE</td>
<td>Saves the contents of the currently-selected editor window.</td>
</tr>
<tr>
<td>PRINT</td>
<td>Prints the contents of the currently-selected editor window.</td>
</tr>
</tbody>
</table>

*Table 6-4: Editor Window Arguments*

**IdlFileReadOnly**

Use **IdlFileReadOnly** to specify the read/write status of the currently-active editor window. One of the arguments from the following table must be set:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>READONLY</td>
<td>Disable editing of the currently-selected editor window.</td>
</tr>
<tr>
<td>READWRITE</td>
<td>Enables editing of the currently-selected window.</td>
</tr>
</tbody>
</table>

*Table 6-5: Read/Write Arguments*

**IdlFunctionKey**

Use **IdlFunctionKey** to allow entry of an IDL command into the input command stream. It is typically used to tie IDL commands to function keys. For example:

```
<Key>F5:IdlFunctionKey("print, 'F5 pressed'")
```

**Action Routines**

**IDL Interface**
**IdlInterrupt**

Use `IdlInterrupt` to cause IDLDE to receive an interrupt. Note that this is usually tied to Ctrl-C as a menu accelerator.

**IdlListStack**

Use `IdlListStack` to display the current nesting of procedures and functions (calling stack).

**IdlLogHide**

Use `IdlLogHide` to hide or expose the Output Log. One of the following arguments must be set: `Show`, `Hide`, or `Toggle`.

**IdlRecallCommand**

Use `IdlRecallCommand` to recalls previously entered commands into the command widget. Either the `BACK` or the `FORWARD` argument must be specified to indicate the direction of the recall. For example:

```plaintext
<Key>osfUp:IdlRecallCommand(BACK)
```

**IdlReset**

Use `IdlReset` to reset the IDL environment.

**IdlRun**

Use `IdlRun` to execute the currently-active file.

**IdlSearch**

Use `IdlSearch` to call the **Find** dialog for a search of the current **Multiple Document Panel**. One of the optional arguments from the following table may be used:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIND</td>
<td>Displays a search dialog (default).</td>
</tr>
<tr>
<td>FINDAGAIN</td>
<td>Finds the next occurrence of the specified string.</td>
</tr>
</tbody>
</table>

*Table 6-6: Find Dialog Arguments*
Chapter 6: Customizing IDL on Motif Systems

### IdlStatusHide

Use `IdlStatusHide` to hide or expose the Status Bar. One of the following arguments must be set: **Show**, **Hide**, or **Toggle**.

### IdlStep

Use `IdlStep` to control statement execution for debugging. At least one of the arguments from the following table must be set.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>FINDSELECTION</td>
<td>Finds next occurrence of the current selection.</td>
</tr>
<tr>
<td>ENTERSELECTION</td>
<td>Enters the current selection as the search string in the Find dialog.</td>
</tr>
<tr>
<td>REPLACE</td>
<td>Replaces the search string, with a specified replacement string.</td>
</tr>
<tr>
<td>REPLACEFIND</td>
<td>Finds the next occurrence of the search string, and replaces it with the specified replacement string.</td>
</tr>
</tbody>
</table>

**Table 6-6: Find Dialog Arguments**

**Table 6-7: Debugging Arguments**
Chapter 6: Customizing IDL on Motif Systems

IDL Interface

Table 6-8: Editor Window Display Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASCADE</td>
<td>Arrange open windows in a staggered, overlapping fashion.</td>
</tr>
<tr>
<td>TILE</td>
<td>Arrange all windows in a non-overlapping fashion.</td>
</tr>
<tr>
<td>MULTI</td>
<td>Open windows outside the IDLDE interface.</td>
</tr>
<tr>
<td>SINGLE</td>
<td>Display the most recent window on the Multiple Document Panel.</td>
</tr>
</tbody>
</table>

Table 6-7: Debugging Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>SKIP</td>
<td>Skips one statement and executes following statement.</td>
</tr>
<tr>
<td>CONTINUE</td>
<td>Continues execution of an interrupted program.</td>
</tr>
<tr>
<td>TOCURSOR</td>
<td>Executes file until encountering the cursor.</td>
</tr>
<tr>
<td>TORETURN</td>
<td>Executes file until encountering the return.</td>
</tr>
</tbody>
</table>

**IdlTrace**

Use **IdlTrace** to display a dialog box to control program tracing.

**IdlViewHide**

Use **IdlViewHide** to hide or expose the Multiple Document Panel. One of the following arguments must be set: **Show**, **Hide**, or **Toggle**.

**IdlWindows**

Use **IdlWindows** to manipulate the state of the Editor windows. One of the arguments from the following table must be set:
Index

A
action routines, 137

B
backing store
  bitmap buffered, 103
  graphics, 103
  system buffered, 103
batch files
  startup preference, 109
bitmap buffered backing store, 103
Bristol Technology
printer manager, 80

C
clipboard support
  graphics windows, 56
colormaps
  flashing, 130
  sharing (Motif), 130
colors
  reserving for IDL, 130
command line
  IDLDE, 56
command line options (Motif), 132
command line switches, 22
compiling
  from memory preference, 106
  preferences, 105, 106
Control Panel Buttons
  modifying in Motif, 134
Motif platform, 58
copyrights, 2
customizing IDL, preferences, 93

\textbf{D}
data formats
  supported, 10
deleting
    lines in Output Log, 96
Direct Graphics
  clipboard support, 56
DISPLAY environment variable, 18

\textbf{E}
editing
  resource files, 129
Editor window
  compiling and saving, 105
  defined, 55
  multiple, 100
  preferences, 105
editors, external (Motif), 123
environment variables
  CLASSPATH, 18
  DISPLAY, 18, 18
  HOME, 18
  IDL_BRIDGE_DEBUG, 19
  IDLJAVAB_CONFIG, 20
  IDLJAVAB_LIB_LOCATION, 20
  LD_LIBRARY_PATH, 20
  LM_LICENSE_FILE, 20
  PATH, 14, 20
  TERM, 20
exiting IDL
  confirm exit, 95
  options, 46
external
  editors (Motif), 123

\textbf{F}
file
  search path, 113
  supported formats, 10
file formats
  about supported, 10
  general data, 11
  image, 10
  scientific data, 11
file types, supported, 10
files
  specifying search path, 113
finding
text, IDLDE search features, 65
flashing colormaps, 130
fonts
  preferences, 110
  specifying
    Motif platform, 111
    Windows platform, 110

\textbf{G}
glyph. See TrueType fonts
graphics
  clipboard support, 56
  image file formats
    supported, 10
  windows
    backing store, 103
    layout preferences, 102
    OS clipboard support, 56
    sizing, 102

\textbf{H}
hardware rendering, setting preference, 104
help
  PDF files
    overview, 43
hiding
  toolbars, 101
HOME environment variable, 18

I
IDL GUIBuilder
  access, 56
  generating
    files, menu option, 61
IDL_BRIDGE_DEBUG, 19
IDLDE
  about, 52
  preferences, 91
image formats, 10
iTools
  clipboard support, 56
  introduction, 15

K
keyboard
  accelerators, Macintosh, 33
  shortcuts, 33

L
launching IDL, 13
layout, graphics window preferences, 102
legalities, 2
lines
  number displayed, output log, 96
  number saved, recall buffer, 96
  output log display number, 96
LM_LICENSE_FILE environment variable
  about, 20

M
Macintosh
  one-button mouse, 32
macros
  IDLDE
    creating in UNIX, 117
    creating in Windows, 120
    working with, 115
memory
  optimizing Windows performance, 96
Menu Editor
  opening, 64
menus
  IDLDE keyboard shortcuts, 33
Microsoft Windows
  mouse differences, 32
Motif widgets, 137
mouse
  emulating three-button, 32
Multiple Document Panel, 55

O
object graphics
  choosing a renderer, 104
objects
  object graphics
    clipboard support, 56
Output Log
  overview, 57
  preferences, 96

P
path
  IDLDE, 113
PDF environment variable, 14
PDF, 43
PDF files, 43
performance
improvement, 103
optimizing memory, 96
Portable Document Format, 43
preferences
  change directories, 97
  changing, 91
  read-only files, 97
  startup, 95
print manager, 80
printing
  from IDLDE, 62
  in UNIX, 80
  in Windows, 79
project
  interface, 55
Properties dialogs (GUIBuilder)
  opening, 64

R
recall buffer
  persistence, 96
  preferences, 96
recent
  files list, 62
  projects, 62
rendering
  hardware versus software, 104
replacing text, 66
reserving colors, 130
resource files, 129
resources for an X Window, 128

S
saving
  files, 61
search path
  specifying with preferences, 113
shared colormaps
about, 130
sizing
  graphics windows, 102
software rendering
  setting preference for, 104
splash screen preference, 95
standard
  data file formats, 11
  image file formats, 10
  scientific data formats, 11
starting
  IDL, 13
startup file
  batch file execution, 109
  overview, 29
startup preferences
  options, 108
  specifying, 95
startup working directory, 108
status bars
  IDLDE, 57
supported file formats, 10
switches, command line, 22
system buffered backing store, 103

T
TERM environment variable, 20
text
  replacing, 66
searching in IDLDE, 65
toolbars
  IDLDE, 54, 55, 55, 56
  Motif platform, 58, 134
  show/hide preference, 101
  specifying layout, 101
trademarks, 2
triangulation
drawing fonts, 104
TrueType fonts, 104
TrueType fonts
<table>
<thead>
<tr>
<th>Graphic preferences, 104</th>
</tr>
</thead>
<tbody>
<tr>
<td>Typographical conventions, 45</td>
</tr>
<tr>
<td>Separating the IDLDE, 101</td>
</tr>
<tr>
<td>Show/hide preference</td>
</tr>
<tr>
<td>Microsoft Windows platform, 100</td>
</tr>
<tr>
<td>Motif platform, 100</td>
</tr>
<tr>
<td>Working directory, changing on startup, 108</td>
</tr>
</tbody>
</table>

**V**

Variable Watch Window, 57

**W**

Windows
  - Arranging layout, 102
  - Clipboard support for graphics, 56

**X**

X resources
  - Using, 128
Xprinter
  - Defined, 80