cādence®

SimVision Introduction

Product Version 15.2
February 2016

Contents

3

Setting Preferences 43

4

Customizing the Layout of SimVision Windows

5

Customizing Toolbars

1

Introduction to SimVision

SimVision is a unified graphical debugging environment for Cadence simulators. You can use SimVision to debug digital, analog, or mixed-signal designs written in Verilog, SystemVerilog, VHDL, SystemC®, or a combination of those languages.

For More Information

■ SimVision Windows and Tools on page 8

■ Using SimVision's Built-in Window Layouts on page 10

■ Managing Windows on page 14

■ Managing Toolbars on page 17

■ Using the Sidebar on page 19

■ Managing Time in the SimVision Windows on page 20

■ Managing Simulation Objects on page 22

■ Setting Preferences on page 24

■ Using Keyboard Shortcuts and Hotkeys on page 27

■ Using the SimVision Command Language on page 27

■ Handling Errors in SimVision on page 28

■ Getting Help on page 36

Video

■ SimVision Quick Introduction to Major Windows

SimVision Windows and Tools

SimVision is made up of several windows. The following windows are available as toolbar buttons and as choices on menus:

The Properties window lets you manage the cursors, markers, expressions, and other debugging objects that you have created during the SimVision session.

The Design Browser window lets you monitor the signals and variables in the design. For more information, see Using the Design Browser.

The Waveform window plots simulation data along an X and a Y axis. Data is usually shown as signal values versus time, but it can be any recorded data. For more information, see Using the Waveform Window.

The Source Browser gives you access to the design source code. For more information, see Using the Source Browser.

The Schematic Tracer displays a design as a schematic diagram and lets you trace a signal through the design. For more information, see Using the Schematic Tracer.

The Memory Viewer lets you observe changes in the internal state of memory locations. During simulation, it also lets you set breakpoints, and force and deposit values to memory locations. For more information, see Using the Memory Viewer.

The Watch window lets you monitor selected signals and variables in the design in a more concise form than the Design Browser. For more information, see Using the Watch Window.

The Register window lets you use a free-form graphics editor to define any number of register pages, each containing a custom view of the simulation data. For more information, see Using the Register Window.

The Expression Calculator lets you define expressions, which combine signals to form buses, conditions, and virtual signals. For more information, see Using the Expression Calculator.

Other buttons might appear in the SimVision toolbar, depending on the technologies you have installed on your system.

The following windows are available only as choices on the *Windows* menu:

New – Measurement	The Measurement window lets you display analog measurements, such as the peak-to-peak ratio or root square mean value of selected signals. For more information, see <u>Using the Measturement Window</u>.
New – Assertion Browser	The Assertion Browser helps you monitor assertions during simulation. For more information, see <u>Viewing Assertions in the Assertion Browser</u>.
New – FSM Window	The FSM window displays a graphical representation of a state machine in your design. For more information, see <u>The FSM Window</u>.
New – Transaction Stripe Chart Viewer	The Transaction Stripe Chart Viewer helps analyze transaction attributes and their values, and the relationship between transactions. For more information, see <u>Analyzing Transactions in the Stripe Chart Viewer</u>.
New – UVM Register Window	The UVM Register Viewer displays a visual representation of UVM_REG registers, and helps you to debug these registers. For more information, see <u>Using the UVM Register Viewer</u>.
New – SystemVerilog Data Browser	The SystemVerilog Data Browser lets you view SystemVerilog dynamic objects, such as classes, queues, and dynamic arrays. For more information, see <u>Using the Data Browser</u>.
New – Embedded Software Trace	Embedded Software Trace helps you debug embedded software in a simulation or emulation environment. For more information, see <u>Using the Embedded Software Trace</u>.
Tools – Simulation Cycle Debug	The Simulation Cycle Debugger lets you step through a simulation cycle, stopping at each time point, delta cycle, simulation phase, or scheduled process. It is not available in Verilog-XL or AMS Designer. For more information, see <u>Debugging at the Delta Cycle Level</u>.
Tools – Console	The Console window lets you enter simulator commands, SimVision commands, or Tcl commands. For more information, see <u>Using the Console Window</u>.

Tools – SimCompare Manager	The SimCompare Manager window lets you perform complex comparisons of signal transitions in simulation databases. This window, plus the Comparisons sidebar, are the graphical user interface to the SimCompare Tool.
	For more information on SimCompare, see *Using SimCompare*.
Tools – Constraints Debugger	The Constraints Debugger window displays information about constraints in your SystemVerilog design. For more information, see Using the Constraints Debugger.
Tools – Design File Search	The Design File Search window lets you search for a text string in all of the source files that make up the design. For more information, see Searching All Design Source Files.

Other windows may appear in your menus, depending on the technologies you have installed on your system.

Using SimVision's Built-in Window Layouts

SimVision provides many debugging tools to be used with a variety of programming languages. As a result, not all toolbars and buttons are applicable to every user or every debugging task. SimVision's built-in layouts are designed for specific purposes. Starting with these layouts, you can further refine them, save your custom layouts, and re-use them or share them with others.

SimVision defines the following built-in layouts:

■ Default—When you are connected to the simulator, the default layout opens a Console window and a Design Browser window. When you are running in post-processing mode (analyzing a simulation database that was saved during a previous simulation run), the default layout opens a Waveform window.

■ RTL Design—This layout is targeted at RTL designers who are creating both design and verification code written in Verilog and VHDL. This layout opens a Design Browser, Console, Source Browser, and Waveform window in a pre-selected configuration. Toolbar buttons for advanced testbench debugging, such as the Stripe Chart Viewer, are removed.

■ SystemC/C/C++ Debugging—This layout is targeted at SystemC testbench verification engineers. See SimVision cdebug Layout in the *SystemC Simulation User Guide* for more information.

■ SystemVerilog Debugging—This layout is targeted at SystemVerilog testbench verification engineers, or RTL designers who want to create class-based SystemVerilog testbenches. This layout opens a Design Browser, Console, Source Browser, and Waveform window in a pre-selected configuration. See Using the SystemVerilog Debugging Layout in *Debugging SystemVerilog* for more information.

■ UVM SystemVerilog Debugging—This layout is targeted at engineers and designers creating SystemVerilog testbenches based on UVM. This layout opens a Design Browser, Console window, Source Browser, Waveform window, and UVM Register Viewer. See Using the SystemVerilog Debugging Layout in *Debugging UVM*.

■ *e* Debugging—This layout is targeted at *e* testbench verification engineers or RTL designers who want to verify RTL designs with *e* testbenches. This layout opens a Design Browser, Console, Source Browser, and Waveform window in a pre-selected configuration. Toolbar buttons for SystemVerilog testbench debugging are removed. See Built-in Layouts for Specman Users for more information.

■ *e* Development—This layout is targeted at e testbench verification engineers who want ot debug their *e* testbenches in the absence of the RTL design and simulator. This layout opens a Source Browser and Console window. Toolbar buttons for SystemVerilog testbench debugging are removed. See Built-in Layouts for Specman Users for more information.

■ VSP Embedded Software Debugging—This layout is targeted at engineers performing system-on-chip design and analysis with the Cadence Virtual Platform. See Embedded Software Debugging Layout in the *Virtual System Platform User Guide* for more information.

■ Embedded Software Tracing—This layout is targeted at verification engineers who are debugging embedded software in a simulation or emulation environment. For more information, see Using the Embedded Software Trace.

For example, the RTL Design layout opens a Design Browser, Source Browser, and Waveform window, arranged as shown in Figure 1-1 on page 12.

Figure 1-1 The RTL Design Window Layout

Design Browser   Source Browser

Waveform window
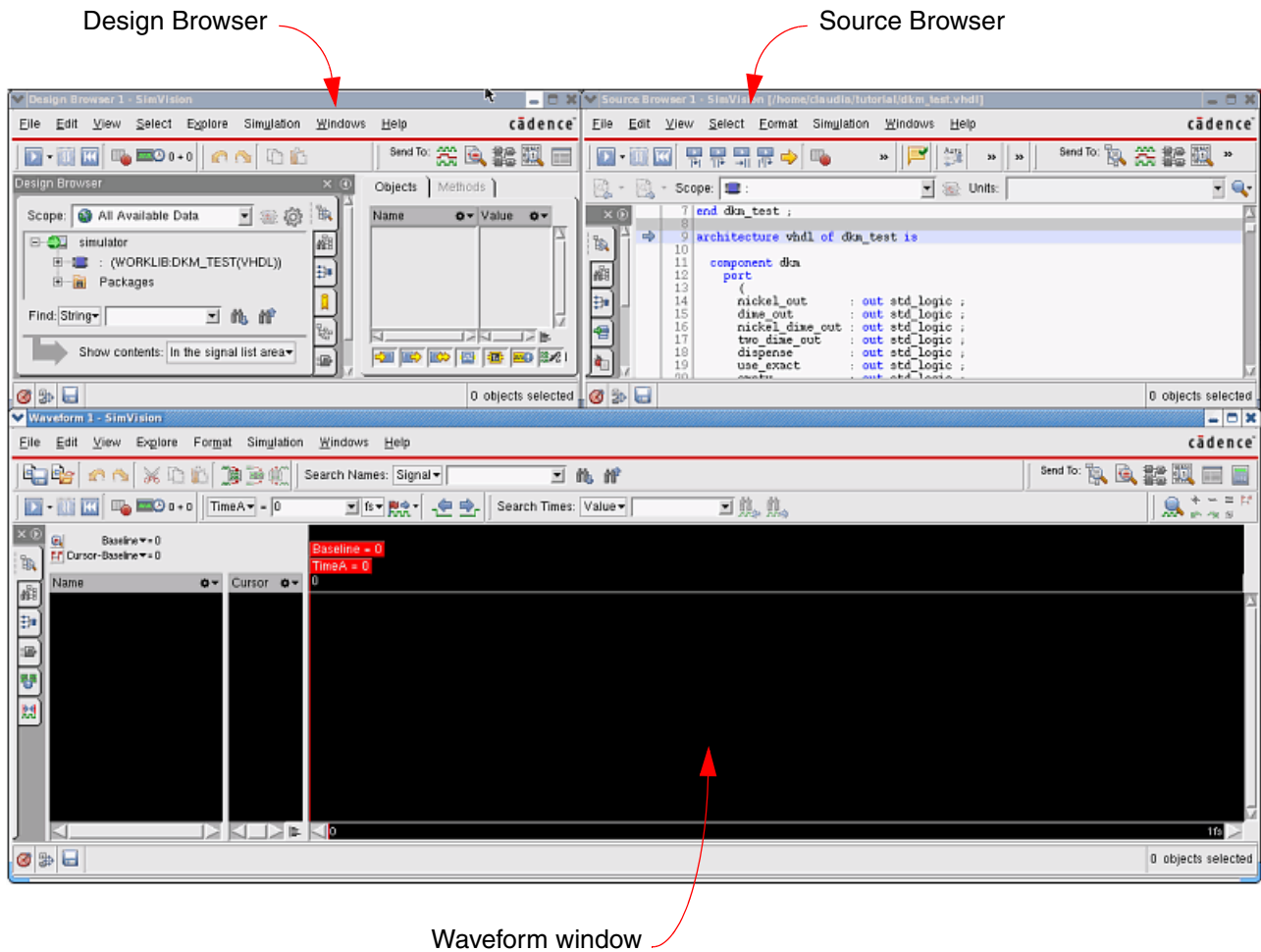
Figure 1-2 The Post-Process Debugging Window Layout
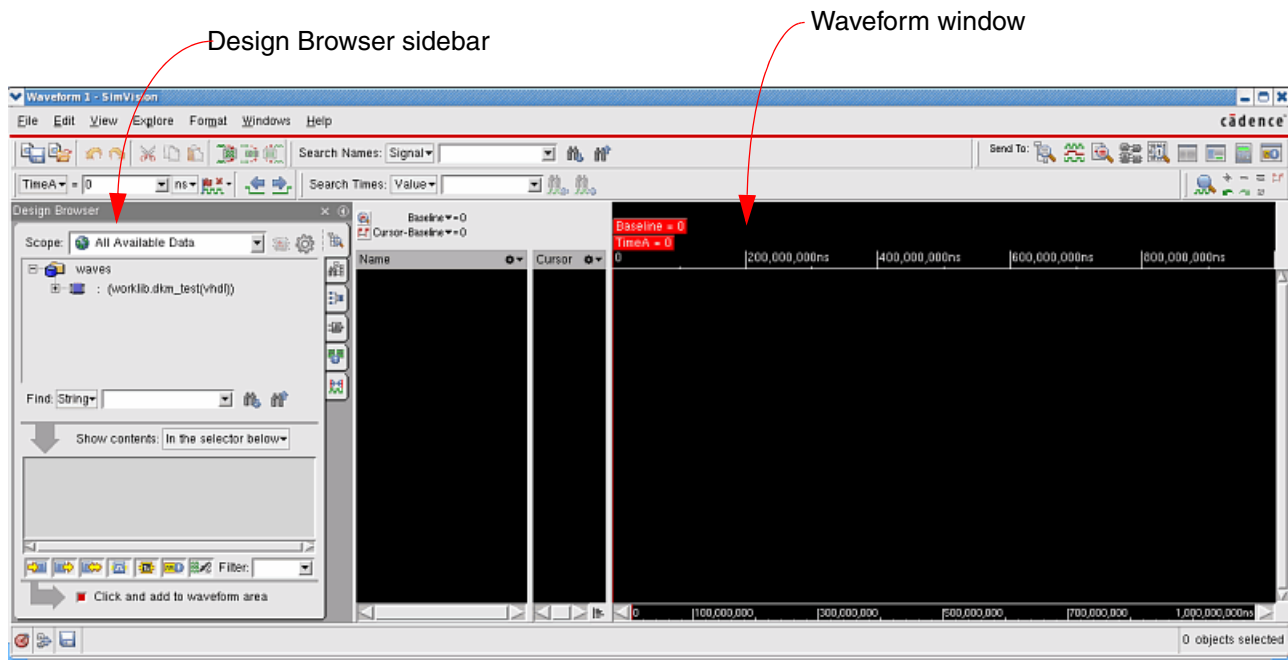
Note: The SimVision documentation describes each window in its entirety, as if no layout has been selected, so that you can see all possible toolbars and buttons that might appear on each window.

To select a built-in layout:

1. Choose *Edit – Preferences*.

2. Open the *Layouts* tab of the Preferences window

3. Choose the layout you want to use from the *Default layout you want to load when starting SimVision* menu.

When you select a layout, SimVision minimizes any currently open windows that are not part of the new layout. You can restore these windows if you want to use them along with the layout you have chosen.

You can also choose a layout when you invoke SimVision, by specifying the name of the layout as one of the following arguments to the `-layout` option:

■   `none`

■   `default`

■   `rtldesign`

■   `cdebug`

■   `svdesign`

■   `uvmdebug`

■   `edebug`

■   `edevelop`

■   `eswdebug`

■   `eswreverse`

■   `eswtrace`

The `-layout` option can be used directly on the simvision command line. When using `irun`, you can pass the argument to SimVision with the `-simvisargs` option. For example:

```
irun -f my_files.f -gui -simvisargs "-layout rtldesign"
```

The command-line option overrides any layout preference you have selected in a prior SimVision session.

If none of the built-in layouts serves your particular debugging style, you can create your own layout, as described in Customizing the Layout of SimVision Windows

Managing Windows

SimVision lets you create multiple instances of some windows, such as the Waveform window and Register window. You can create a new (empty) window, which you populate with signals and variables, or you can create a replica (clone) of an existing window, which contains the same signals and variables as the original window.

To create a window:

➤   Choose *Windows – New* or *File – New* from the window's menu bar, then choose the type of window that you want to create.

To create a clone of a window:

➤   Choose *File – New – Clone* from the menu bar.

When you have multiple instances of a window type, one window of that type is the target. The target window is the one to which any operation is applied. For example, if you select objects in one window, you can add those objects to the target window.

To make the window the target:

➤ In the bottom left corner of each window, enable the *Target* icon, 🔴.

For More Information

■ Opening and Closing Windows on page 15

■ Iconifying and Activating Windows on page 16

■ Tiling Windows on page 16

■ Renaming Windows on page 17

Opening and Closing Windows

You can access many SimVision windows with buttons on the Send To toolbar, shown in Figure 1-3 on page 15.

Figure 1-3 Send To Toolbar
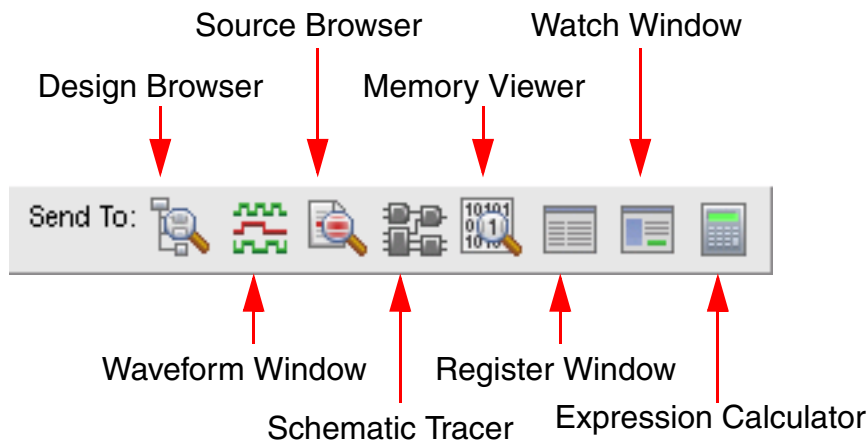
The window buttons have drop-down menus to help you manage multiple windows. You see these menus by hovering the cursor over the button for a few seconds. The drop-down menu contains the following choices:

■ *Send to target*—Adds any selected objects to the target window. If no window of that type exists, SimVision creates a new one. If no objects are selected, it either brings the current target window to the foreground or creates a target window of that type.

■ *Send to new*—Creates a window and adds the selected objects to that window. If no objects are selected, SimVision creates an empty window.

■   *Window name*—All of the windows that you create are listed in the menu, with the target window in bold letters. Choose a window from the menu, and SimVision adds any selected objects to that window.

From any window, you can open any other SimVision window using the following menu choices:

■   *Windows – New* and *File – New* create a window of the type you choose. If you select objects before creating the window, those objects are added to the new window.

■   *Windows – Tools* lets you access the specified tab of the Properties window. The Properties window lets you manage your debugging environment, including markers, cursors, mnemonic maps, and simulation properties, such as breakpoints and probes.

■   *Window Name*—The menu lists all windows that are currently open and lets you select a window from the list to bring the window to the foreground and give it focus.

■   *Windows – Windows* opens the Windows form. This form lists the windows that are currently open. You can select one or more windows from the list, then activate, iconify, or close the window. This form also lets you lay out the windows, as described in Tiling Windows.

To close a window:

➤   Choose *File – Close Window* from the window's menu bar.

Iconifying and Activating Windows

To iconify an individual window:

➤   Click the *Iconify* button in the window's title bar, or select the window in the Windows form and click *Iconify*.

When the Windows form is open, you can activate an individual window by selecting it in the list of windows and then clicking *Activate*.

To iconify all SimVision windows into a single SimVision icon:

➤   Choose *Windows – Iconify All* from any SimVision window.

Tiling Windows

By default, SimVision windows are different sizes, and they appear in different parts of your terminal display. You can move windows around in any way you like. However, if you create many windows, you may find it difficult to lay them out so that they are easy to find.

SimVision gives you the following layout options for your windows:

■ *Windows – Tile Horizontally* lays out the windows so that each spans the terminal display horizontally. The windows are of equal height and fill the display from top to bottom. This option makes each window the same height and width.

■ *Windows – Tile Vertically* lays out the windows so that each spans the terminal display vertically. The windows are of equal width and fill the display from left to right. This option makes each window the same height and width.

■ *Windows – Cascade* lays each window on top of another, offset from the upper left corner of each window. This option does not change the size of the windows.

Renaming Windows

SimVision gives every window a default name, such as *Waveform 1* or *Design Browser 1*.

To rename a window:

1. Choose *File – Rename Window* from the menu bar of the window you want to rename. SimVision opens the Rename Window form.

2. Enter the name in the *New Name* field and click OK.

Managing Toolbars

Many common menu functions are also available through buttons on toolbars. The toolbars that appear in a window differ from one window type to another. For example, the Design Browser window contains the following toolbars:

Send to	The Send To toolbar, which invokes the SimVision tools, such as the Waveform window, Souce Browser, and Schematic Tracer
Standard	The standard toolbar, which performs functions such as opening a database, and cutting and pasting objects
User Toolbar	User-defined toolbars and buttons

The Waveform window contains those toolbars, plus the following toolbars:

Cursor Control	The cursor control toolbar, which lets you choose a primary cursor and specify its location

Signal List Search	The signal list search toolbar, which lets you search for signals and groups
Search Times	The time search toolbar, which lets you search for values, rising edges, falling edges, expressions, markers, and assertion states
Zoom	The zoom toolbar buttons

Because toolbars take up space that you might want to use for displaying data, you can choose which toolbars you want to display in a window and remove those toolbars that you do not use.

To choose the toolbars that you want to display:

1. Choose *View – Toolbars* or press the right mouse button over any blank area of a toolbar to display a menu of toolbars that are available in the window.

2. Enable the toolbars you want to display; disable the toolbars you want to remove from the window.

You can move a toolbar to any position within the toolbar area, between the menu bar and the data area.

To move a toolbar:

➤ Place the mouse over the small grip on the left side of the toolbar, and drag the toolbar to its new location.

Changing the appearance of toolbars in one window does not change their appearance in any other windows that are already open. However, these settings are applied to any new windows that you open.

SimVision also lets you add and remove buttons from the SimVision toolbars, and it lets you create your own toolbars and buttons. For more information, see Customizing Toolbars

Using the Sidebar

Many windows contain a sidebar, which you use to access objects in your design. The sidebar can contain the following tabs, depending on the window type and language in which the design is written:

The Design Browser sidebar lets you access the scopes in the design hierarchy. For information, see Using the Design Browser Sidebar in *Using the Design Browser*.

The Design Search sidebar lets you search for objects in all open simulations and databases, without regard to the design hierarchy. For information, see Using the Design Search Sidebar in *Using the Design Browser*.

The Trace Signals sidebar lets you trace a signal value either forward or backward through the design hierarchy. For information on the Trace Signals sidebar, see Tracing Signals in the Trace Signals Sidebar in Simple Trace Mode and Tracing Signals in the Trace Signals Sidebar in Full Trace Mode in *Tracing Signals*.

The Call Stack sidebar displays a call stack for VHDL procedures, processes, and functions, API applications, and a thread manager for SystemC processes. This tab appears in the Source Browser sidebar. See Viewing the Call Stack in *Using the Source Browser* for more information.

The SystemC/C++/C Variables sidebar displays information about local variables and data members in SystemC and API applications. This tab appears in the Source Browser sidebar. For more information, see Observing SystemC/C++ Variables in the *SystemC Simulation User Guide.*

The Bookmarks sidebar lets you manage bookmarks, which save the current state of a window, so that you can return to it at any time. The bookmarks tab appears in the Design Browser, Source Browser, and Schematic Tracer windows.

The Class Hierarchy sidebar displays information about SystemVerilog classes. For more information, see Viewing Classes in the Class Browser in *Debugging SystemVerillog*.

The Power sidebar displays information about powered-up and power-down signals in the design. For more information, see Opening the Power Sidebar in *Debugging Low Power*.

The Comparison Results sidebar lets you access the mismatches found by SimCompare. For information on SimCompare, see the *Using SimCompare*.

Use the following buttons and menu choices to control the sidebar:

➤ To collapse the sidebar, click *Collapse Sidebar*, 🔳, in the upper right corner of the sidebar.

➤ To expand the sidebar, click *Expand Sidebar*, 🔳, or select the tab that you want to expand.

➤ To hide the sidebar, disable the *View – Sidebar* option or click *Hide Sidebar*, 🔳, in the upper right corner of the sidebar.

➤ To show the sidebar, enable the *View – Sidebar* option.

➤ To tear off a tab from the sidebar, click on the dotted line in the tab control button. The tab becomes a separate window, and the sidebar is collapsed.

➤ To re-attach a tab to the sidebar of its containing window, close the window.

🔅 *Tip*

Because of SimVision's cross-selection capability, any object you select in the sidebar is selected in all SimVision windows. Therefore, you can tear off a tab from any sidebar and close the sidebars in your other windows to maximize space in your SimVision windows.

Managing Time in the SimVision Windows

Most windows contain the time toolbar, shown in Figure 1-4 on page 20.

Figure 1-4 Time Toolbar
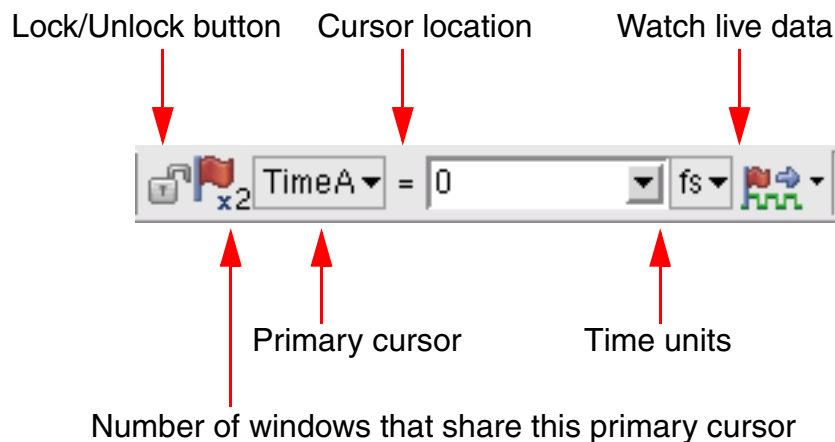
This toolbar lets you control the simulation time associated with the window, as follows:

■ Click the *Lock/Unlock* button to lock and unlock the primary cursor in the window at a particular time. When simulation time is locked, the values displayed in the window remain at the locked time, even as simulation progresses. When you unlock the window, the primary cursor tracks simulation time.

 Note: You must probe the signals of interest in order to lock their values.

 Note: Although the primary cusor and the values displayed in the window remain at the locked time, the simulation time in the simulation toolbar continues to update as simulation progresses.

■ The *Number of windows* icon shows you the number of windows that share the same primary cursor. When multiple windows share a primary cursor, the values displayed in those windows reflect the same simulation time.

■ The *Primary Cursor* field lets you select the primary cursor for the window. A drop-down list contains the names of all cursors currently defined in all windows. You can choose a cursor from the list or choose *New Cursor* to create a cursor.

■ The *Cursor Location* field lets you place the primary cursor at a specific simulation time. As you enter simulation times in the field, SimVision adds them to a drop-down list. You can quickly return to a time by selecting it from the list.

■ The *Time Units* field lets you select the time units used in the window.

■ The *Watch Live Data* button indicates whether the primary cursor is synchronized with the simulator or database. This can affect performance, but it lets you track signal value changes during simulation. The button has a drop-down menu from which you can choose the simulator or database that you want to track.

 When synchronization is on, the *Watch Live Data* button shows a green arrow. The cursor location updates in the window as simulation time progresses. Other windows that share the primary cursor do not get updated until simulation pauses or stops.

 If you move the primary cursor when *Watch Live Data* is enabled, synchronization is turned off.

When synchronization is off, the Time toolbar color changes to yellow, and a warning message pops up to tell you that the primary cursor is no longer synchronized with the simulation. The *Watch Live Data* button shows a red X and the cursor location does not change. Waveforms get updated periodically, but the view remains stationary.

Note: When you are in post-processing mode, the time toolbar does not turn yellow.

Managing Simulation Objects

The *Edit* menu lets you cut, copy, paste, and delete objects in the window. These functions are also available as buttons in the standard toolbar, shown in Figure 1-5 on page 22. Many operations that you perform in a window can be undone. If so, the *Undo* menu choice and button are activated. When you undo an operation, the *Redo* menu choice and button are activated.

Figure 1-5 Standard Toolbar
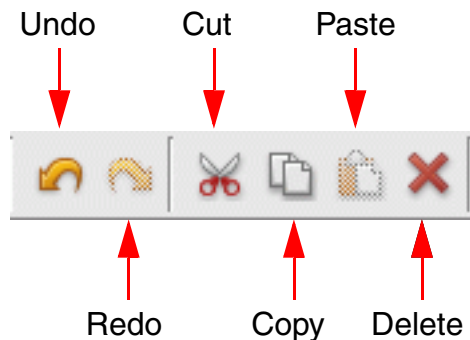
Note: The *Undo* and *Redo* functions are not available in the Console window.

For More Information

■ Selecting and Deselecting Objects on page 23

■ Using Pop-Up Menus on page 23

■ Adding Objects to a Window on page 23

Selecting and Deselecting Objects

Many operations begin by selecting one or more objects.

To select individual objects or multiple objects:

➤ Select a single object by clicking on it.

➤ Select several contiguous objects by `Shift`-clicking the first and last objects.

➤ Select non-contiguous objects by `Control`-clicking on the objects.

To deselect one or all selected objects:

➤ Deselect a single object by `Control`-clicking on the selected object.

➤ Deselect all selected objects by clicking on an empty space in the window.

Using Pop-Up Menus

When you have selected one or more objects, you can perform operations on them by using pop-up menus. Select the objects and right-click to display the pop-up menu. The choices in the pop-up menus differ, depending on the objects that you have selected.

Adding Objects to a Window

The *Add* button, ✚, is available in the standard toolbar of most SimVision windows. Click this button to add to the current window any objects that you have selected in another SimVision window.

You can quickly move objects from one window to another by performing a drag-and-drop operation.

To drag and drop objects:

1. Select the objects, then hold down the left mouse button and move the cursor to the new location.

   As you move the mouse, the cursor indicates whether you may drop the objects at the current cursor location:

   ⊘   The mouse is in an invalid location. You cannot drop the objects here.

 The mouse is in a valid location. You can drop the objects here.

2. Release the mouse button to drop the objects at the new location.

Drag-and-drop operations do not require you to first select an object. If you do not select an object, SimVision drags and drops the object closest to the mouse pointer. If that object is a scope in the design hierarchy, all objects in that scope are dragged and dropped to the destination. If that object is a signal, only that signal is dragged and dropped.

Setting Preferences

You can set preferences to customize SimVision windows. In most cases, preference settings take effect in all open window when you apply them. In a few cases, they take effect only in any new windows that you open after the setting is applied.

To open the Preferences window:

➤   Choose *Edit – Preferences* from any SimVision window.

The Preferences window organizes preferences in tabs, by window or by function. The Preferences window opens to the tab for the window in which you issued the menu command. That is, if you choose *Edit – Prefereces* from the Design Browser, the window opens the Design Browser tab, as shown in <u>Figure 1-6</u> on page 25.

Figure 1-6 Design Browser Tab in the Preferences Window
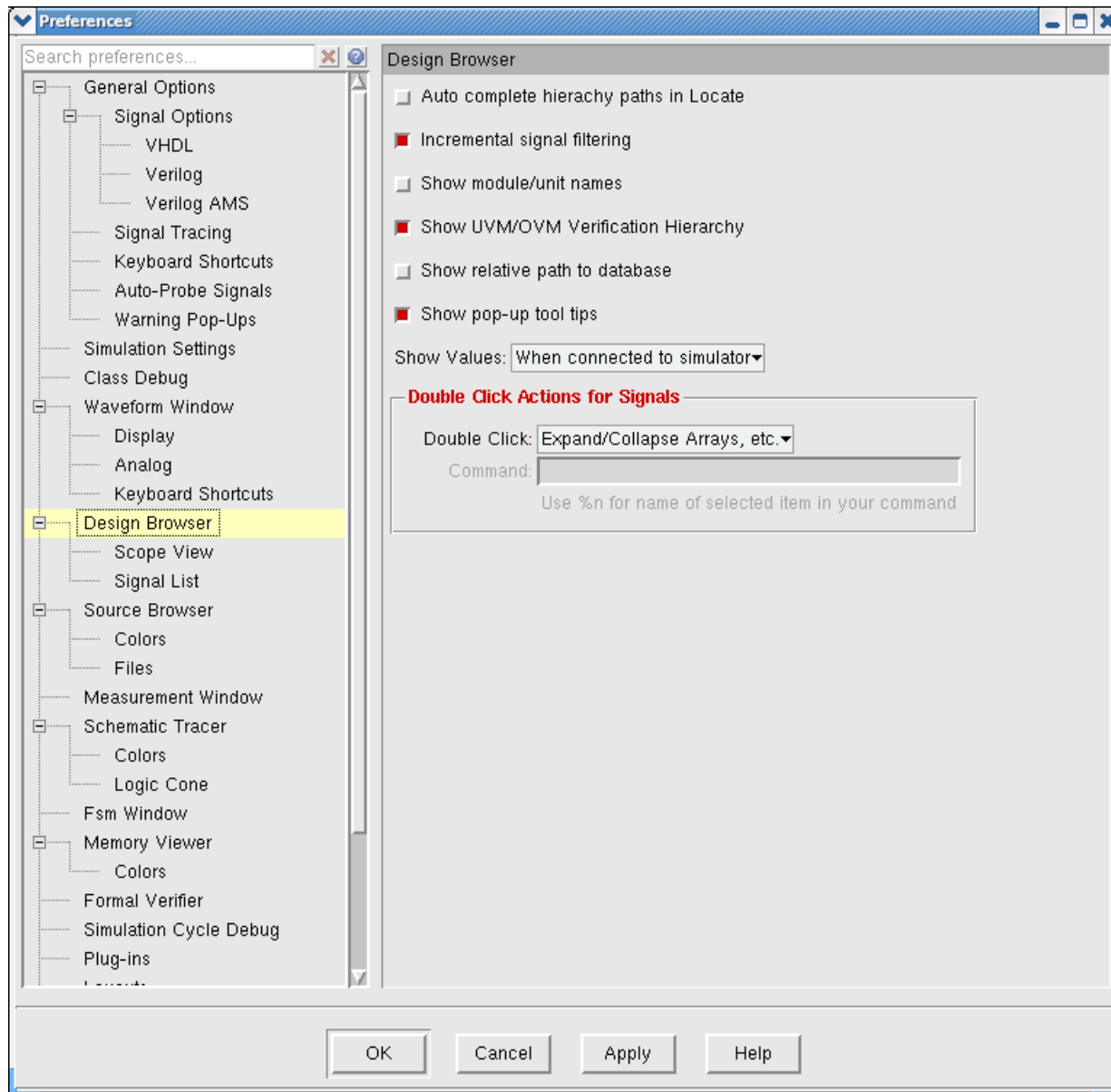
There are many preferences, and you might not always know which tab to open to set a particular preference.

For example, you might want to search for all preferences that let you change the colors used in SimVision windows.

A search field at the top of the list of tabs lets you search for preferences.

To search for preferences:

➤ Enter a glob-style search string in the search field at the top of the list of tabs.

SimVision removes from the list all tabs that do not have preferences that match the string. When you open a tab, it highlights the preferences in the tab that match the string, as shown in Figure 1-7 on page 26.

Figure 1-7 Preferences Containing the String "color"
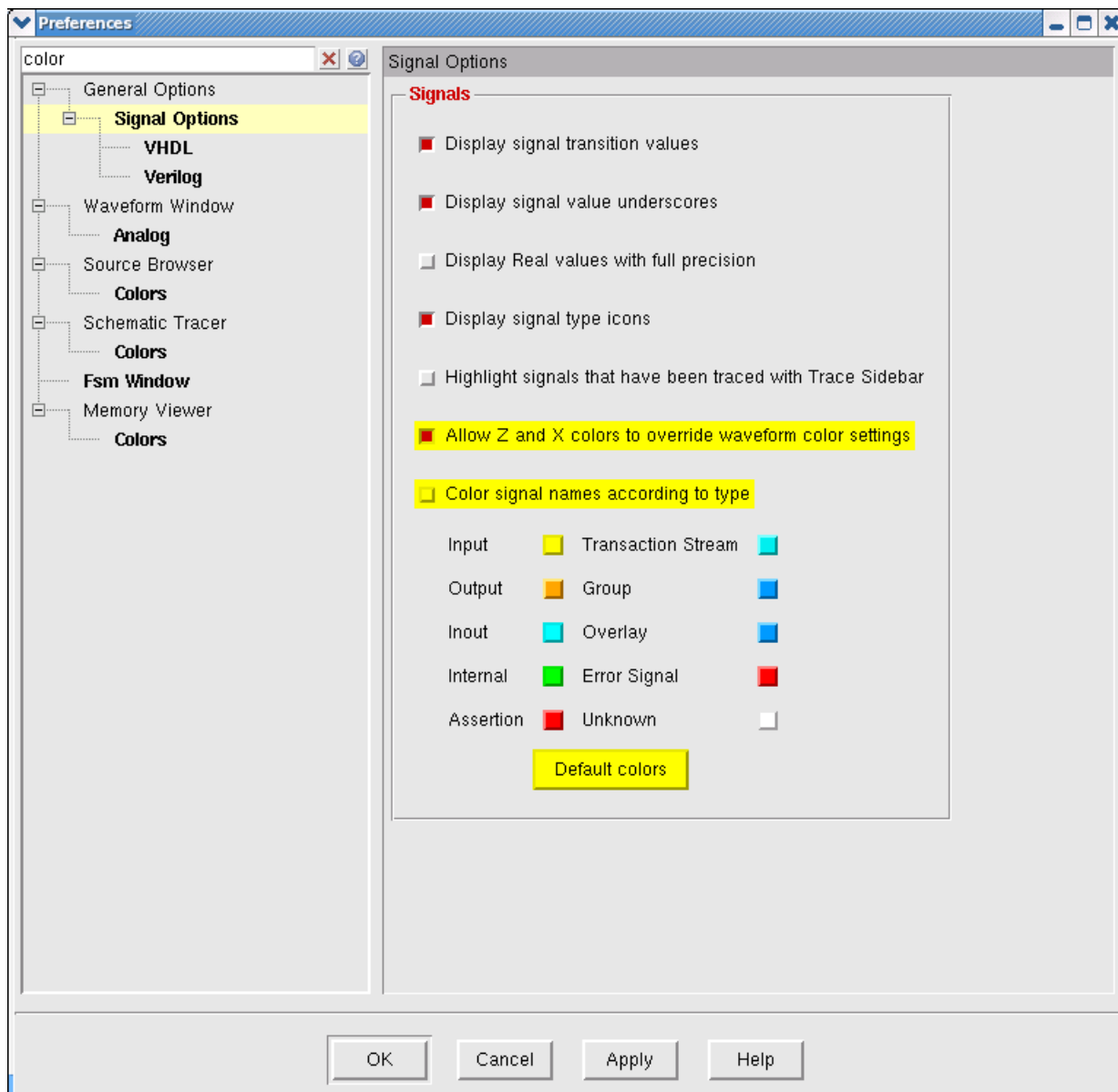
For a description of all preferences, see Chapter 3, "Setting Preferences."

Using Keyboard Shortcuts and Hotkeys

Every menu choice is bound to a shortcut key sequence. You may find these shortcuts faster to use than the pull-down menus. To use a shortcut, press the `Alt` key, then type the underlined letter in each menu choice. For example, to choose *File – Open Database*, type `Alt-FO`.

SimVision also defines a set of hotkeys, which are bound to certain commands. You can see a list of hotkey definitions by choosing *Help – Keyboard Shortcuts*. You can change these keyboard shortcuts or define your own.

To see your current shortcut settings and save them to a file:

1. Choose *Help – Keyboard Shortcuts*, and click *See all application shortcuts*. SimVision pops up a list of all keyboard shortcuts that are currently defined.

2. In the list of shortcuts, click *Save shortcuts*, 💾 , and specify the name of the file where you want to write the shortcuts. The default file name is `shortcuts.txt`. You can then print this file to use as a quick reference.

See Also

■   Keyboard Shortcuts tab of the Preferences window

■   Creating Custom Keyboard Shortcuts

Using the SimVision Command Language

Every operation that you can perform with the SimVision graphical user interface can also be performed with a SimVision Tcl command, as follows:

■   Enter commands in the Console window while running SimVision.

■   Create a command script that calls SimVision commands to open databases, create windows, cursors, and so on. You can then execute the command script at startup to initialize a SimVision session. SimVision can create this script for you, as described in Saving and Restoring Your Debugging Environment.

■   Create a command script that calls SimVision commands and Tcl commands to create a plug-in application. Plug-ins define new window types and add new features to SimVision.

See Also

■ *Using SimVision Tcl Commands*

■ *SimVision Tcl Commands*

Handling Errors in SimVision

When running SimVision, you might encounter problems:

■ You might want to stop an action that is taking a long time, such as loading a very large database.

■ SimVision might report a warning or error condition.

■ The simulator or SimVision might exit unexpectedly.

These kinds of problems should not occur often, if at all. However, if they do, SimVision has ways to handle them.

For More Information

■ Aborting a SimVision Action on page 28

■ Viewing and Saving SimVision Diagnostic Messages on page 29

■ Recovering a Debugging Session on page 30

■ Out-of-Memory Errors on page 33

Aborting a SimVision Action

There might be times when you want to stop an action before it has completed. For example, you might want to stop loading a very large database. When SimVision detects that an action will take a long time to complete, it adds an *Abort* button to the relevant window. You can click this button to stop the action and resume your SimVision session.

There might be times the *Abort* button is not available, for example, if you are trying to execute a command script that contains an infinite loop. If no *Abort* button is available, you can still stop an action and regain control of the SimVision windows by pressing the `Escape` key three times. The triple-Escape sequence will return control of the SimVision windows to you.

Viewing and Saving SimVision Diagnostic Messages

Occasionally, SimVision generates diagnostic messages. These messages usually indicate internal conditions that do not prevent SimVision from operating correctly. In rare instances, they might indicate a more serious problem.

SimVision writes diagnostic messages to a separate window, which you can open by choosing *Help – SimVision Diagnostics*. Under ordinary circumstances, this window displays messages such as those shown in Figure 1-8 on page 29. In this figure, the messages show that several windows opened successfully at startup. If an internal error had occurred, it would be flagged as a Warning or Error message.

Figure 1-8 SimVision Diagnostics Window
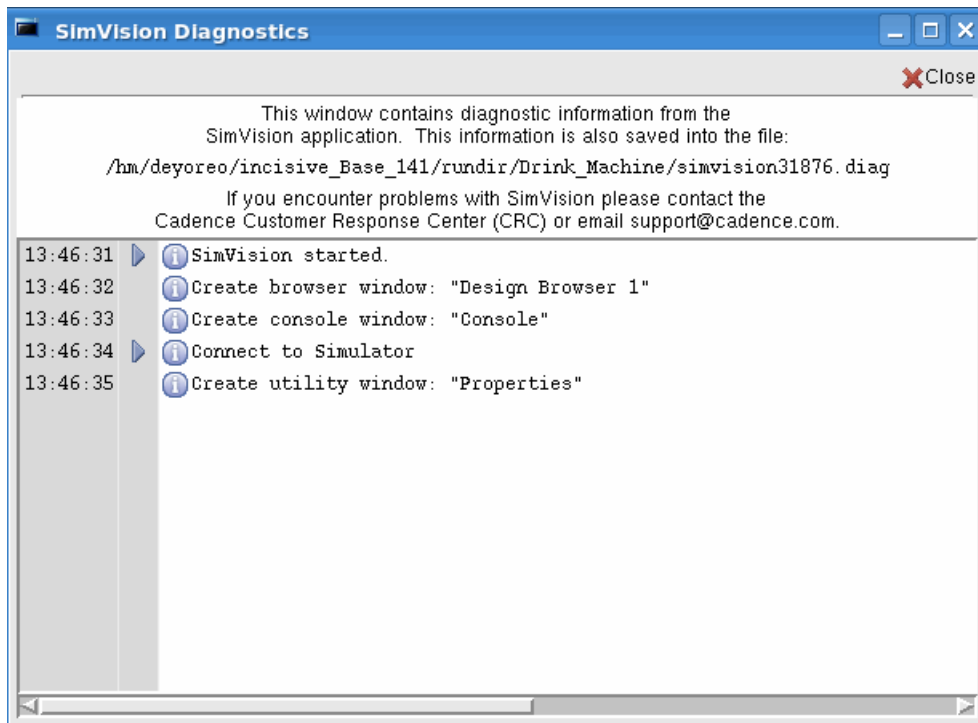
When a warning or error condition is detected, SimVision also writes the diagnostics messages to a file in your working directory. This file is named `simvision`*PID*`.diag`, where *PID* is the SimVision process ID. You can specify a different name for this file by using the `-diagfile` command-line option when you start SimVision. If you encounter problems while running SimVision, you can send this file to your Cadence Support Specialist.

SimVision sets a threshold on the number of diagnostic messages that can occur during a certain period of time. When this threshold is reached, SimVision displays a dialog box that

allows you to save your session, continue running SimVision, or quit SimVision, as shown in <u>Figure 1-9</u> on page 30.

Figure 1-9 SimVision Diagnostics Window when the Threshold has been Exceeded
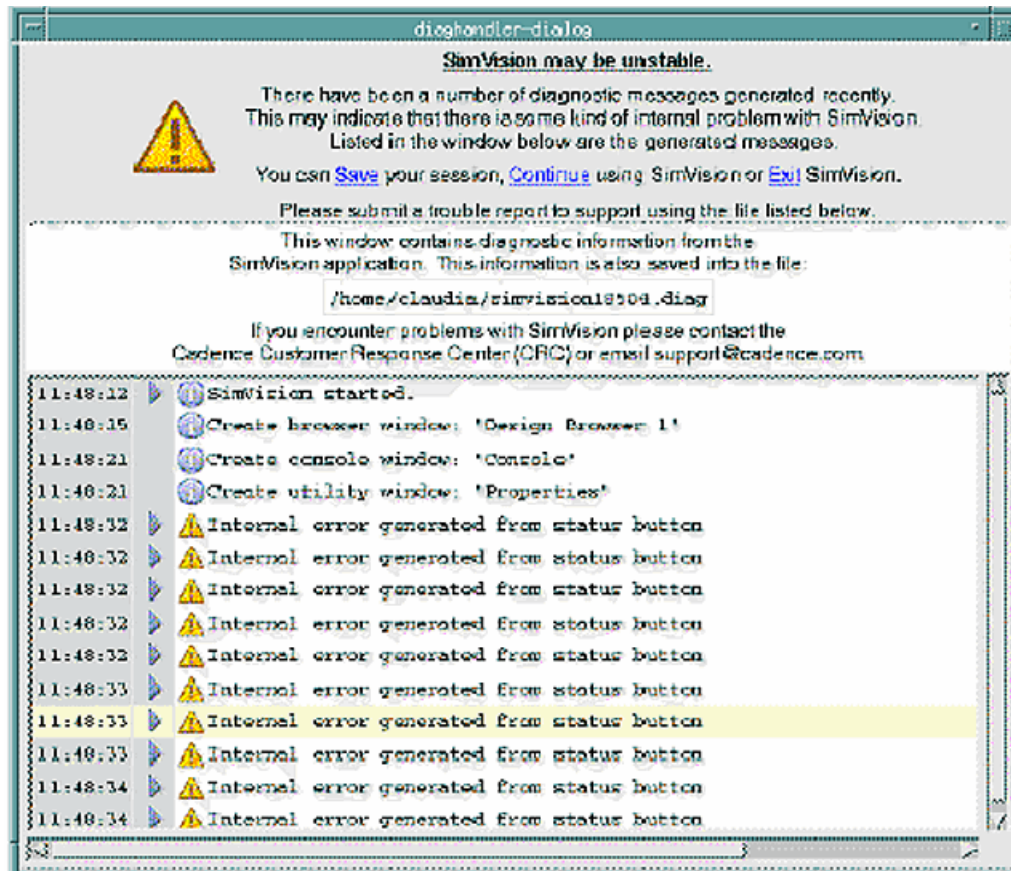
You can raise or lower the threshold by issuing the `preferences` command in the SimVision Console window. For example, to set the threshold to 30 messages in a one-minute time period, you would issue the following commands:

```
preferences set diagnostics-threshold-count 30
preferences set diagnostics-threshold-time 60000
```

Recovering a Debugging Session

If the simulator should encounter an error that causes the simulation to exit, SimVision pops up a message box containing information about the error, as shown in <u>Figure 1-10</u> on page 31.

Figure 1-10 Error Recovery Message Box
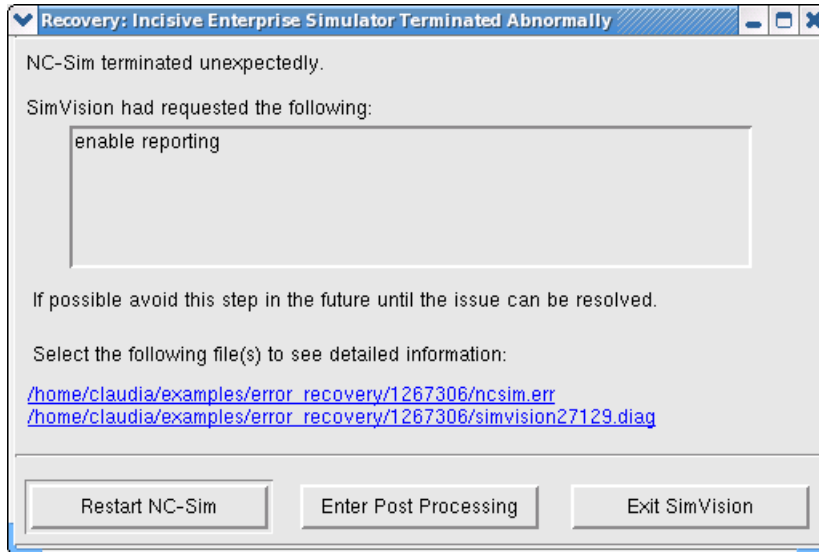
The message box describes the action that SimVision was attempting when the error occurred. It also provides links to any error log files that are available. In this example, the simulator has saved an error file, `ncsim.err`, and simviion has saved a diagnostics file, `simvision27129.diag`. You can click on these links to open the file in your chosen editor.

The message box also lets you choose how you want to handle the error:

■   Click *Restart NC-Sim* to return to time 0 and run the simulation again, perhaps stopping immediately before the error occurred.

■   Click *Enter Post Processing* if you have probed signals to a simulation dataase. You can debug the simulation in post-processing mode up to the point where the error occurred. If no simulation information was saved, this button is disabled.

■   Click *Exit SimVision* to end the SimVision session.

In addition to the simulator error file and SimVision diagnostic files, SimVision can also save a file ithat contains the following information:

■   All windows that you have opened, including their contents, size, and position

■   Any preferences that you have set

■   All cursors, markers, groups, or other objects that you have created during the debugging session

■   All probes, breakpoints, forces, and deposits that you have set during simulation

The autosave file is written periodically, and it is deleted when SimVision exits normally.

When you are running in simulation mode, the autosave file names have the following format:

```
PID_username_hostname_autosave.tcl
PID_username_hostname_autosave.tcl.svcf
```

When you are running in post-processing mode, the autosave file name has the following format:

```
PID_username_hostname_autosave.tcl.svcf
```

The file name includes the PID, user name, and host name in case several simulations are running in the same directory at the same time.

By default, the autosave file is updated every 8 minutes. You can change the interval by setting a preference. You can also set a preference to disable the autosave feature.

The status bar in all SimVision windows contains an icon that indicates whether autosave is enabled or disabled, as shown in Figure 1-12 on page 33. A check mark flashes in the icon whenever the autosave file is written.

Figure 1-11 Autosave Icon in the Status Bar

Autosave icon (enabled)
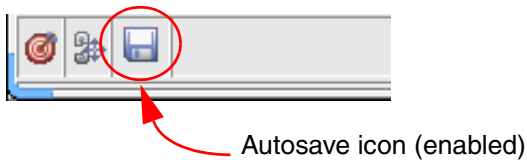
At startup, SimVision checks to see if an autosave file exists. If so, it pops up a dialog box and lets you choose whether you want to restore the debugging environment, as shown in Figure 1-12 on page 33.

Figure 1-12 Autosave Dialog Box
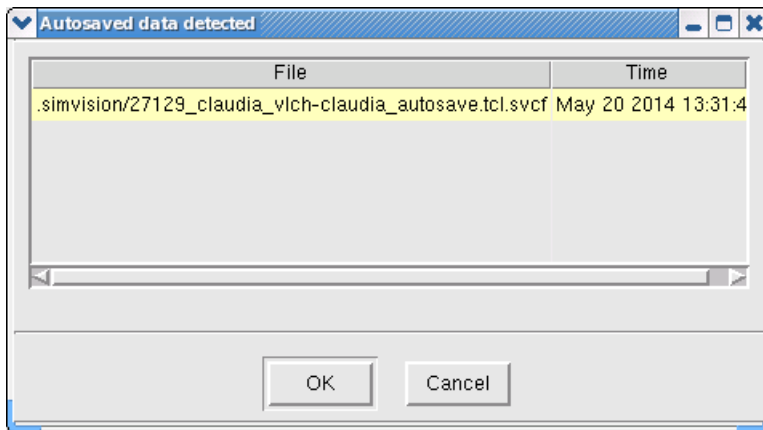
If more than one SimVision process was running in the same working directory when SimVision exited, this dialog box shows the name of each autosave file that it detects in that directory at startup.

To restore the session:

■   Select the autosave file, then click *OK*.

You can restore the autosave file in either simulation or post-processing mode. Furthermore, if an autosave file is generated in simulation mode, you can restore your environment in post-processing mode. If the autosave file is generated in post-processing mode, you can restore the environment in simulation mode. When mixing modes, however, some information might be lost or displayed differently, that is:

■   If you run SimVision in simulation mode and restore the environment in post-processing mode, you can access only those signals that were probed during simulation. In post-processing mode, you do not have access to the design snapshot, which contains information about all signals. To access all of the signals, you must reconnect to the simulator.

■   If you run SimVision in post-processing mode and restore the environment in simulation mode, the snapshot and the database are both available, but displayed separately in the Design Browser scope list.

Out-of-Memory Errors

SimVision can exit for an *out-of-memory* situation in the following ways:

■   While trying to allocate memory and there is none available. In this case, SimVision aborts the process in a *controlled exit*.

■ When, as part of normal operation, SimVision requires more memory (for example to increase the size of the stack) and none is available. This results in an *unexpected exit*, caused by a signal from the operating system. In this case the operating system aborts the SimVision process.

In both cases, SimVision provides information about the nature of the failure without the need for manual intervention regardless of type of debug operation that is in progress.

Controlled Exits

When SimVision performs a controlled exit, a pop-up message is displayed to inform you of the nature of the failure, and that the process will exit. For example, Figure 1-13 on page 34 indicates that the process could not allocate 253 bytes of memory.

Figure 1-13 Memory Allocation Failure Pop-Up Message
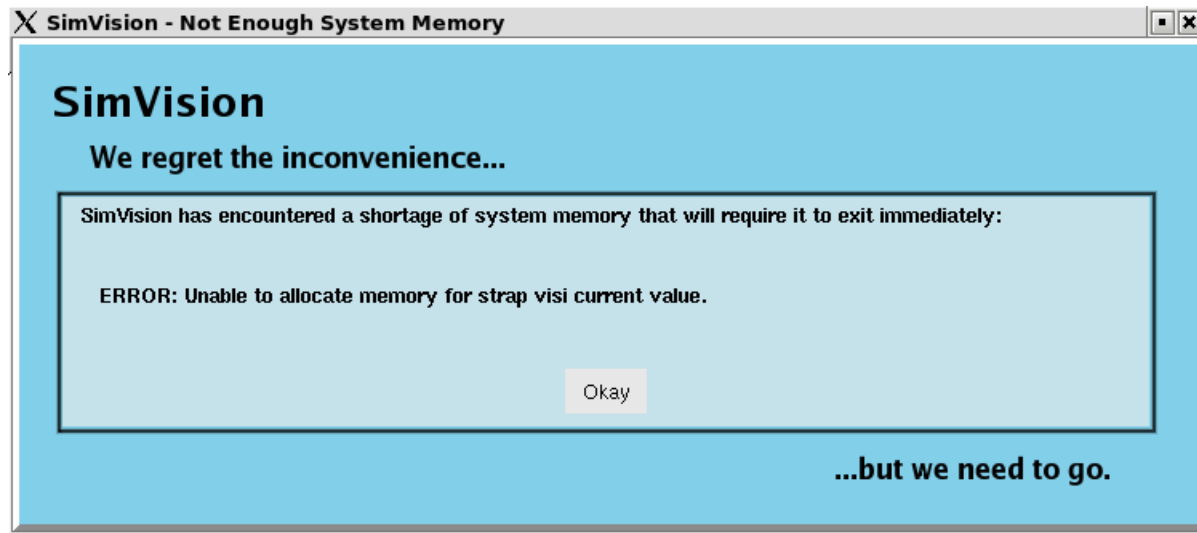
A text file containing the error information, similar to that shown in Figure 1-13 on page 34, is also created.

Figure 1-14 Memory Allocation Failure Text File Message

Unexpected Exits

Some of the reasons that would cause the operating system to unexpectedly exit the SimVision process include:

■   Out of memory situations such as heap and stack overruns.

■   Attempting to accessing memory outside the user space.

■   Other unprotected actions, such as writing to a file that no longer exists.

In the case of an unexpected exit, a message similar to that provided by the *ncsim* process is written to the shell where the process was invoked.

The message written to the shell includes the name of the signal received from the operating system, and any additional information made available by the process.

For example, Figure 1-15 on page 35 shows an unexpected exit message describing a SimVision Exception Handler exit that was initiated by the SIGSEGV operating system signal.

Figure 1-15 An Unexpected Exit Message displayed in the Shell
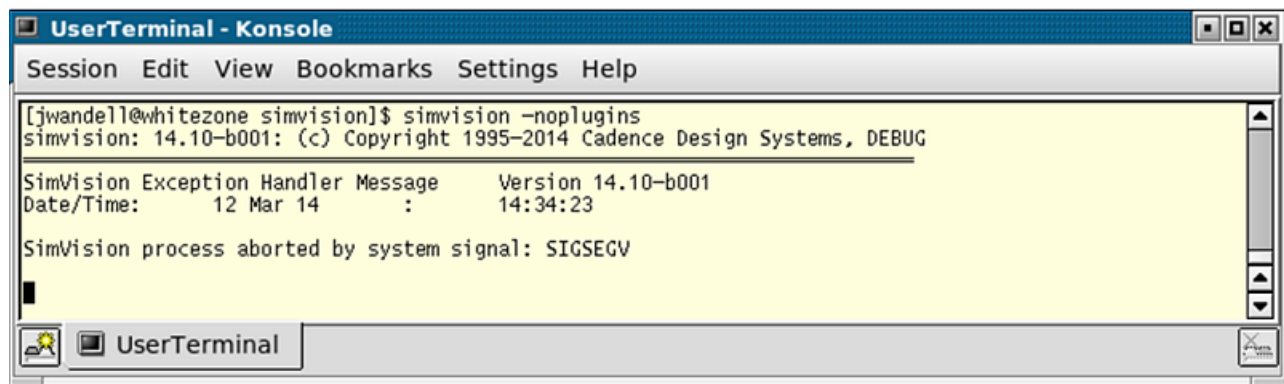
See Also

■   General Options on page 44

■   Saving and Restoring Your Debugging Environment

■   Connecting to a Simulation from SimVision

Getting Help

There are several ways to get help when running SimVision:

■ Tooltips appear whenever you point your cursor at a button for a few seconds. The same message appears in the status bar at the bottom of the window.

■ The *Help* button on the Preferences form displays interactive help. This help feature lets you click on a field in the form to see a description of the field.

■ *Help* buttons on other forms take you to the relevant parts of the documentation.

■ The *Help* menu contains the following choices:

❏ *Cadence Help Library* opens the Cadence Help Library window. This gives you access to all of the Cadence documentation installed on your system.

❏ *Introduction to SimVision* describes the basic features of the SimVision.

❏ *What's New* takes you to *What's New in IUS, which* describes the new features in this release.

❏ *Known Problems and Solutions* takes you to the *Known Problems and Solutions* documents for the products installed on your system. These documents describe any problems that are known to exist at the time of this release.

❏ *Keyboard Shortcuts* displays a list of hotkey definitions supplied with SimVision.

❏ *SimVision Diagnostics* opens a window showing any SimVision diagnostics messages that have been issued during the current session.

❏ *Customer Service* takes you to the Cadence web pages for Cadence Online Support and Web Collaboration.

❏ *What's New Screen* displays the What's New startup screen.

❏ *About SimVision* displays the SimVision version and copyright information.

Other menu entries may appear, depending on the tools and technologies you have installed on your system.

You can also watch a series of SimVision videos on YouTube at <u>SimVision Debug Video Series</u>. Note that the audio portion of these videos is not available if you run them in a VNC session.

2

Using the Console Window

The Console window gives you access to the command languages for SimVision and for any simulator you are running.

Each tool is represented by a tab in the Console window. For example, if you run SimVision with the simulator, the Console window contains a *SimVision* tab and a *simulator* tab, as shown in Figure 2-1 on page 37. If you are debugging a SystemC or API application, the Console window also contains a tab for the GDB debugger.

You can enter commands at the Console window prompt for any tool, and monitor the messages returned by the tool.

Figure 2-1 Console Window with the SimVision and Simulator Tabs
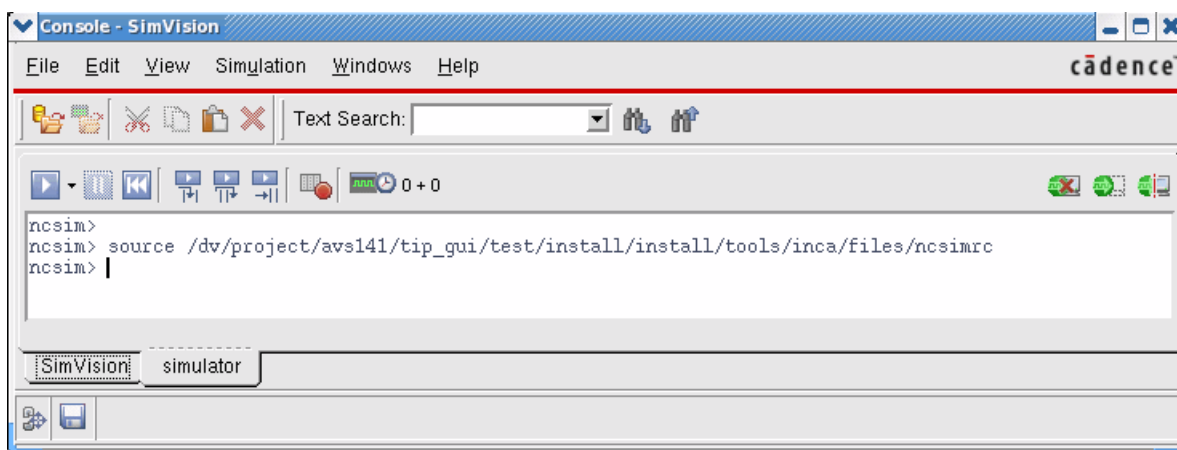
For More Information

■   Searching for a Text String in the Console Window on page 38

■   Launching the Debug Analyzer from the Console Window on page 40

■   Controlling the Simulaton from the Console Window on page 41

■   Changing the Colors of the Console Window on page 41

See Also

■   *SimVision Tcl Command Reference*

■   *Incisive Simulator Tcl Command Reference*

Searching for a Text String in the Console Window

To search for a text string by using the *Search* toolbar:

➤   Enter a search string in the *Text Search* field, and click *Search Up*, 🔍 , or *Search Down*, 🔍 .

The string can include any of the following special characters:

*   Matches any number of characters

?   Matches a single character

"   Matches a word boundary, where a word boundary is any character except a letter, number, or the underscore. Because the quotation mark is a special character, you must use a literal string or a regular expression to search for strings that contain quotation marks.

To specify a literal or a regular expression, begin the search string with one of the following patterns:

***=   Treat the remainder of the search string as a literal.

***:   Treat the remainder of the search string as a Tcl advanced regular expression. Tcl advanced regular expressions let you define complex search strings.

Because SimVision keeps a history of the search strings that you enter, you can enter a partial string and press `Tab`. If the history contains one matching string, SimVision seeds the *Search* field with that string. If the history contains more than one matching string, it displays a list of those strings, and you can choose one.

The following are some examples of searching for the string `out`:

`out`   Returns strings that contain `out`, such as `out`, `output`, and `inout`.

`<space>``out`

Returns strings that end with `out`, such as `out` and `inout`, but not `output`.

`"out`   Returns strings that begin with `out`, such as `out` and `output`.

`"out"`   Returns only the word `out`.

`***="out"` Returns all occurrences of the quoted string `"out"`.

`***:\yout\y`

Returns only the word `out`, where `\y` is a Tcl word boundary indicator.

`***:(\yout\Y|\Yout\y)`

Returns words that begin with `out`, and words that end with `out`; that is, it returns `inout` and `output`, but not `out`.

`***:((ino|outp)ut)`

Returns the words `inout` and `output`, but not `out`.

To search for a text string by using the Text Search form:

1. Choose *Edit – Text Search*, or click `Alt-S`. SimVision opens the Text Search form.

2. Enter a search string in the *Find what* field, including the special characters and patterns that you can use in the *Text Search* field.

   Note: If you select text before opening the Text Search form, the *Fine what* field is seeded with that text.

   The Text Search form also give you the following options:

   ❑ Disable *Regular Expression* to specify the exact string you want to find. When this button is disabled, special characters and patterns are treated like ordinary characters.

   ❑ Enable *Match Case* if you want the search to be case-sensitive.

   ❑ Enable *Up* or *Down* to control the direction of the search.

3. Click *Find Next* to find each occurrence.

4. Press *Close* to end the search.

If you want to select all of the text in the Console window (for example, to copy and paste the results of a debugging session into a text file), choose *Edit – Select All* from the menu bar, choose *Select All* from the pop-up menu, or click `Alt-A`.

Launching the Debug Analyzer from the Console Window

Note: This toolbar appears only when you have the Incisive Advanced Debug Option license.

The Incisive Debug Analyzer is a post-process, graphical debugger. You run the simulation once to gather data about both the DUT and the *e* testbench. After simulation is complete, you can load the data into the Debug Analyzer for analysis. Because the simulation data is stored in a database, you do not need to rerun the simulation in order to debug the test environment.

The Debug Analyzer lets you:

■ Explore your test environment for static and dynamic information

■ Step forward or backward through the simulation, or jump to a specific point in the simulation

■ Investigate possible reasons why the simulation has reached a particular state

■ Filter all messages coming from any platform—*e* and HDL code—and explore the cause of the messages

When SimVision detects the Advanced Debug Option at startup, it adds the Debug Analyzer toolbar to the Console window, as shown in

Figure 2-2 The Debug Analyzer Toolbar

Launch the Debug Analyzer

Set a recording probe   Stop recording

See Also

■ Introduction to Incisive Debug Analyzer in *Using Incisive Debug Analyzer*

Controlling the Simulaton from the Console Window

The Console window contains a simulation toolbar, which lets you control the simulation, as shown in Figure 2-3 on page 41. It is the only unconfigurable toolbar in the window. That is, you can add and remove other toolbars, or remove individual buttons from other toolbars, but not in the simulation toolbar.

Figure 2-3 The Simulation Toolbar
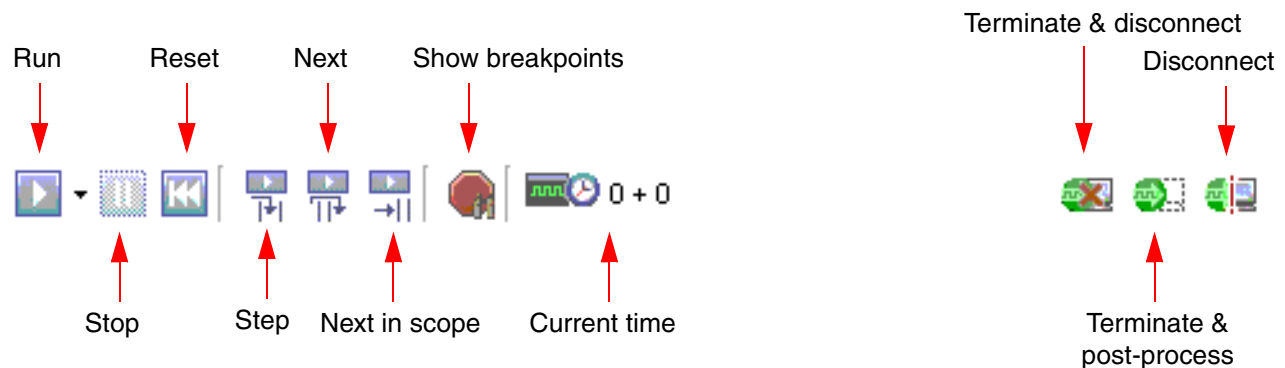
See Also

■ Controlling the Simulation in *Running SimVision*

Changing the Colors of the Console Window

There are four possible tabs that can appear on the Console window, referred to by the X Windows System with the following names:

■ `ConsoleSvTab`—The *SimVision* tab

■ `ConsoleSimTab`—The *simulator* tab

■ `ConsoleSpecmanTab`—The *Specman* tab

■ `ConsoleGdbTab`—The *Gdb* tab

The Console window can also have three background colors:

■ `normalBackground`—The background color during normal operation. The default is white. All tools use the `normalBackground` color setting.

■ `doneBackground`—The color of the background when the tool exits. The default is gray. The `doneBackground` color is used by all tools except SimVision.

■ `busyBackground`—The color of the background when the tool is busy. The default is light gray. The `busyBackground` color is used by the simulator and Specman.

To change these default colors:

➡ Create a file named `~/.simvision/Xdefaults` and specify the background colors that you want to use.

For example, the following `Xdefaults` file sets the normal background colors of the *SimVision* tab to pink, the *simulator* tab to cyan, the *Specman* tab to yellow, and the *Gdb* tab to orange:

```
*ConsoleSvTab.normalBackground: pink
*ConsoleSimTab.normalBackground: cyan
*ConsoleSpecmanTab.normalBackground: yellow
*ConsoleGdbTab.normalBackground: orange
```

Important

You cannot have any spaces at the end of the lines in the Xdefaults file. This can cause an error to occur at startup.

3

Setting Preferences

You can set the following preferences to customize SimVision windows. Preference settings take effect in all open windows when you apply them, except where noted.

For More Information

- General Options

- Signal Options

- VHDL Signal Options

- Verilog Signal Options

- Verilog AMS Signal Options

- Signal Tracing

- Keyboard Shortcuts

- Auto-Probe Signals

- Warning Pop-Ups

- Simulation Settings

- Class Debug

- Waveform Window

- Waveform Display

- Analog Waveform

- Design Browser

- Design Browser Scope View

- Design Browser Signal List

- Source Browser

- ■   Source Browser Colors

- ■   Source Browser Files

- ■   Measurement Window

- ■   Schematic Tracer

- ■   Schematic Tracer Colors

- ■   Schematic Tracer Logic Cone

- ■   FSM Window

- ■   Memory Viewer

- ■   Memory Viewer Colors

- ■   Formal Verifier

- ■   Simulation Cycle Debug

- ■   Plug-Ins

- ■   Layouts

- ■   Transaction Explorer

General Options

To customize the general appearance and behavior of SimVision:

1. Choose *Edit – Preferences* from the menu bar and select *General Options* in the left column of the Preferences form.

2. Select the toolbar style that you prefer. You can choose to display only the icons, only the icon labels, or both icons and labels. By default, SimVision displays only the icons.

3. Enable *Limit the number of lines kept in the console window*, if you want to set the size of the Console window buffer. The default size is 5000 lines.

4. Choose the behavior you want SimVision to have when it exits:

   ❑   By default, SimVision deletes all unused cursors and groups when it exits. When you start up SimVision again (without the `-norestore` option), it does not restore the unused objects. Disable *Delete Unused Groups* and *Delete Unused Cursors* if you do not want these things deleted.

❑ By default, SimVision prompts you on exit. You must click *Yes* to exit the tool. Disable *Prompt* in the Preferences form to exit without displaying the Exit form.

5. Choose startup defaults for the following options:

❑ Enable *Show Full Signal Names* if you want to display the full hierarchical names of signals, not including the database name.

❑ Enable *Show Database Names* if you want to include the database name in signal names.

6. Enable *Watch live data enabled with creating cursor*, if you want the SimVision windows to update as simulation progresses. This is the default when you are connected to a simulation.

Note: You might need to scroll the Preferences window to see the following options.

7. Enable *Create Cursor for each New Window*, if you want each window to have its own primary cursor. By default, windows share cursors, so that as you move a cursor in one window, all windows update to show the same cursor view. When you enable this option, each window maintains its own view.

When *Create Cursor for each New Window* is enabled, you can specify the time units for those cursors.

8. By default, SimVision displays a warning message when you try to save a command script that that may not restore properly. This can happend if you have used an input file to set up the simulation environment. If you do not want to see this warning message, disable the *Warn me when saving scripts that may be problematic due to input files* option.

9. By default, SimVision generates an autosave file every 8 minutes, so that you can restore your session if SimVision or the simulator should exit abnormally. When this feature is enabled, you can set a different autosave time interval. You can disable *Enable autosave* if you do not want SimVision to save this information.

Signal Options

To customize the signal list:

1. Choose *Edit – Preferences* from the menu bar, and select *Signal Options* from the list on the left.

2. When you enable *Display signal transition values*, SimVision displays two values when the value of an object changes during the current clock cycle. For example, if a

signal transitions from 0 to 1 in the current clock cycle, SimVision displays the transition as `` `b0 -> `b1 ``.

3. Enable *Display signal value underscores* to display an underscore between every eight characters of an octal, hexadecimal, or binary value. Disable this option to display the value with no underscores.

4. Enable *Display Real values with full precision* if you want to display real values with full precision in the signal list. Ordinarily, real values are limited to five decimal places.

5. Use *Display signal type icons* to enable or disable the signal type icons. You can disable icons if you need to conserve space in your Waveform windows.

6. When you enable *Highlight signal names that have been traced with Trace Sidebar*, signals that you have traced are highlighted in all windows in which they are displayed. In the Source Browser, they are displayed in green. In the other windows, they are underlined. Choose the *View – Clear Trace Highlights* in any window to clear the highlights in all windows.

7. By default, the color of z and x values do not change when you set the color of a waveform. This lets you identify these values in the waveform. If you want the x and z values to have the same color as the rest of the waveform, disable *Allow Z and X colors to override waveform color settings*.

8. Enable *Color signal names according to type* to use different colors for each signal type, then select the colors that you want to assign to those signal types.

9. Click *Default colors* at any time to return to the default color scheme.

VHDL Signal Options

To change the colors of waveforms:

1. Choose *Edit – Preferences* from the menu bar, expand *Signal Options* in the list on the left, and select *VHDL*. The Preferences form displays the VHDL color map.

2. To change a color setting, click on the color that you want to change, and choose the new color from the pop-up menu.

   If you want to mix your own color, click *More* in the pop-up menu. This opens the Select a New Color form. Slide the pointers to the left and right to adjust the amount of red, green, and blue that you want to use. The *Selection* fields show the hexadecimal number for the color and a color sample.

3. Click *OK* when you are satisfied with the color.

Verilog Signal Options

To change the colors of waveforms:

1. Choose *Edit – Preferences* from the menu bar, expand *Signal Options* in the list on the left, and select *Verilog*.

2. Enable *Show Signal Strength in Values* if you want the Waveform window to display the strength along with the signal values.

    Note: This setting does not affect windows that are already open, only new Waveform windows that you create.

3. Enable *Show colors by value* to color waveforms according to their values, and to choose the colors for each value.

4. Enable *Show colors by strength* to color waveforms according to their strength, and to choose the colors for each strength.

5. To change a color setting, click on the color that you want to change, and choose the new color from the pop-up menu.

    If you want to mix your own color, click *More* in the pop-up menu. This opens the Select a New Color form. Slide the pointers to the left and right to adjust the amount of red, green, and blue that you want to use. The *Selection* fields show the hexadecimal number for the color and a color sample.

6. Click *OK* when you are satisfied with the color.

Verilog AMS Signal Options

To customize Verilog AMS waveforms:

1. Choose *Edit – Preferences* from the menu bar, expand *Signal Options* in the list on the left, and select *Verilog AMS*.

2. Choose how you want to display AMS branches, as follows:

    ❑   Enable *Show Potential* to display potential quantities.

    ❑   Enable *Show Flow (if available)* to display flow quantities.

3. Choose how you want to format potential and flow values, as follows:

    ❑   Enable *Show Scale Factor and Unit* to use scalar multipliers and nature units, such as `-30.646mV`.

❑   Enable *Show Floating Point* to use floating-point format, such as `-0.0306461`.

Signal Tracing

To customize the signal tracing behavior:

1. Choose *Edit – Preferences* from the menu bar, and select *Signal Tracing* in the list on the left, or click *Options*, ⚙, in the Trace Signal sidebar.

2. Enable *Include external drivers/loads when tracing* if you want to include these signals in the trace path.

3. By default, the Signal Tracer sets the color of traced signals to green in any Source Browser windows that are open, and it underlines the signals in other windows. If you do not want traced signals to be highlighted, disable the *Highlight signals that have been traced with Trace Sidebar* option.

4. When tracing with the Source Browser or Waveform window, enable *When tracing an X value use Automatic Driver Tracing* if you want to trace the source of the X value in one step.

5. When tracing with the Source Browser or Waveform window, enable *When tracing a non-X value use Automatic Driver Tracing* if you want to trace the source of a non-X value in one step.

6. For all trace operations, enable *Automatic X Trace: Stop tracing if there are multiple X values* if you want to choose which path you want to take.

7. When tracing with the Schematic Viewer only, enable *Automatic X Trace: Follow all if there are multiple X values* if you want to see all possible causes of an X.

8. Specify a value for *Traced signal edges to go back in time looking for a match*.

9. Choose the mode in which you want the Signal Trace sidebar to operate, either *Simple Trace* or *Full Trace*.

10. Choose how you want signal names to be displayed in the Trace Signals sidebar—either the signal name only, the full hierarchical path, or the full path plus the database name.

11. When in simple trace mode, by default, the Trace Signals sidebar traces signal into modules and stops only when it finds more than one driver. Enable *Stop at module boundaries while tracing* if you want to stop whenever the path reaches a module boundary.

12. When in full trace mode, by default, the Trace Signals sidebar traces X values one step at a time, and stops to let you choose whether to continue tracing. Enable *Automatically*

trace X values as far as possible if you want to continue tracing until the Signal Tracer finds the source of the X value or reaches a decision point.

13. When in full trace mode, enable *Automatic X Tracing skip directly to the cause of the X*, when you want the Signal Tracer to display only the direct cause of the X value, and skip any intervening steps.

Keyboard Shortcuts

SimVision defines two sets of keyboard shortcuts:

■ Application Keyboard Shortcuts perform the same functions as menus and buttons. For example, in the Waveform window, you can select a set of signals and create a group by clicking *Group*, ![icon], or by choosing *Edit – Create – Group*. However, you can also use the keyboard shortcut, `Control-G`, to create the group.

■ Waveform Keyboard Shortcuts perform the same functions as thhe buttons and menu choices that manage the waveform display, such as zooming in and out, and scrolling the display. You can define your own keyboard shortcuts for these commands.

SimVision comes with a set of default keyboard shortcut definitions. However, you can define your own shortcuts and bind your own key sequences to any SimVision command.

☀ *Tip*

To see a list of your current keyboard shortcuts, choose *Help – Keyboard Shortcuts*. Whenever you change the shortcut definitions, the new definitions appear in the Help display.

Application Keyboard Shortcuts

There are many menus and commands, and it can be hard to find the one you are looking for. A search field at the top of the tab lets you search for a menu command. For example, suppose you want to define a shortcut for the *Edit – Preferences* menu choice.

To search for this menu choice:

✝ Enter a glob-style string in the search field. As you type, any menus that do not have *preferences* as a menu choice are removed from the Menus list, and any commands that do not contain the string are removed from the Commands list, as shown in Figure 3-1 on page 50. The list is filtered as you type, so you do not usually need to enter the entire string before finding the menu entry you are looking for.

Figure 3-1 Finding the Preferences Menu Choice
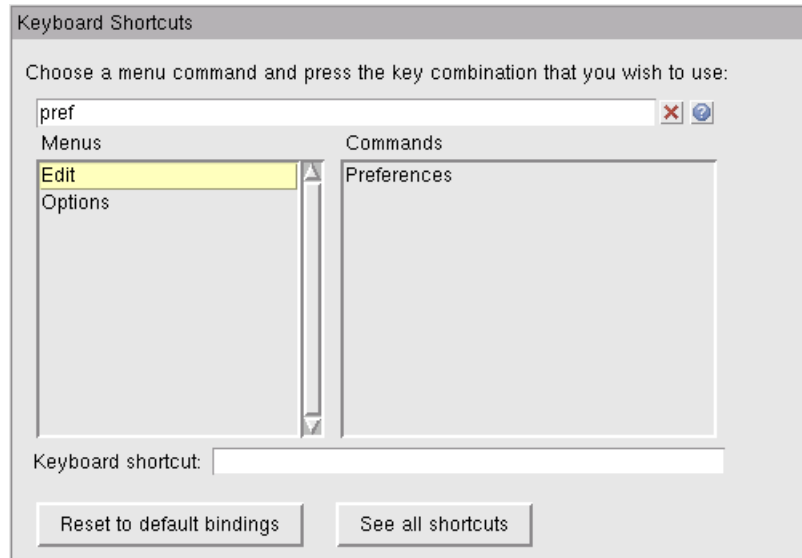
To create your own keyboard shortcuts:

1. Choose *Edit – Preferences* from the menu bar, expand *Signal Options* in the list on the left side of the Preferences form, then select *Keyboard Shortcuts*.

2. Select the name of the menu in the *Menus* area of the window. When you choose a menu, its menu entries are listed in the *Commands* area of the window.

3. In the *Commands* area, select the menu entry for which you want to define a shortcut.

4. In the *Keyboard shortcut* field, enter the key sequence that you want to bind to the command.

5. At any time, you can click *Reset to default bindings* to remove the shortcuts that you have defined.

6. Click *See all shortcuts* to pop up a list of all keyboard shortcuts that are defined. In the keyboard shortcut list, click *Save shortcuts*, 🖫 , to write the shortcuts to a text file, which you can print for a quick reference.

Waveform Keyboard Shortcuts

To define keyboard shortcuts for waveform commands:

1. Choose *Edit – Preferences* from the menu bar, expand *Waveform Window* in the list on the left side of the Preferences form, then choose *Keyboard Shortcuts*.

2. Place the cursor anywhere in the text field for the definition you want to change, and press the `Del` or `Back Space` key. This removes the old definition.

3. With the cursor still in the text field, press the desired key combination. The names of the keys that you press appear in the text field.

4. At any time, you can revert back to the default key bindings by clicking *Reset to default bindings*, or replace the key bindings with the Signalscan bindings by clicking *Reset to Signalscan bindings*.

5. Click *See all shortcuts* to pop up a list of all keyboard shortcuts that are defined. In the keyboard shortcut list, click *Save shortcuts*, 🖫 , to write the shortcuts to a text file, which you can print for a quick reference.

Auto-Probe Signals

To specify how you want signals to be probed in the Watch window:

1. Choose *Edit – Preferences* from the menu bar and select *Auto-Probe Signals* in the list on the left side of the Preferences form.

2. Enable *Do not ask about probing signals sent to the Watch Window* if you do not want to see the pop-up dialog box that appears you send unprobed objects to the Watch window. You can also disable this option from the pop-up dialog box

3. Enable *Always probe objects sent to the Watch Window* if you want to probe all objects that you send to the Watch window.

4. Enable *Always add contributors of Assertions sent to the Waveform Window* if you want to see the contributing signals for an assertion.

5. Choose one of the following types of signal to represent assertions that you send to the Waveform window:

   ❑ *State trace*

   ❑ *Transaction*

   ❑ *Single Failure Probe*

Warning Pop-Ups

The Waveform window pops up a warning message when the primary cursor no longer tracks simulation time.

To disable this warning pop-up:

1. Choose *Edit – Preferences* from any SimVision window, then choose *Warning Pop-Ups* from the list on the left side of the window.

2. Enable *Do not warn if the cursor is out of sync with simulation time*.You can also enable this preference by clicking a link in the warning pop-up box displayed when simulation time is no longer synchronized with the primary cursor.

Simulation Settings

To specify simulator settings:

1. Choose *Edit – Preferences* from any SimVision window, then choose *Simulation settings* from the list on the left side of the window.

2. Enable *Prompt before Reinvoke* if you want SimVision to prompt you for command-line options when you reinvoke the simulation. If this button is disabled, SimVision reinvokes the simulator with the original command-line options that you used.

3. Enable *Respond to simvision commands sent from a simulator* if you want to accept commands sent to SimVision from the simulator. When this command is disabled, SimVision ignores commands sent by the simulator. You might want to disable a command for security reasons if you are accessing a simulation that is running on another system.

4. By default, SimVision does not create save retention probes or waveform groupss for low-power simulations. Enable these options if you want to create these probes and groups.

5. By default, SimVision continues running in post-processing mode when you type `exit` at the simulator prompt in the Console window. Enable *Exit SimVision when exiting simulator* if you do not want to continue running SimVision when you exit the simulator.

6. By default, Specman opens the Source Browser when it stops at breakpoint. Enable this option if you do not want the Source Browser to be opened at those times.

7. From the *SystemC Debugger* drop-down menu, choose the debugging mode that you want to use, either *DDD integration* or *native SimVision gdb debugger*.

8. From the *Quick Thread Debugging* menu, choose whether you always want to use Quick Threads with the PLI Debugger, or whether you want to use Quick Threads only when SystemC is present.

9. By default, SimVision saves SystemC breakpoints and restores them in a subsequent session. Disable *Restore SystemC breakpoints between sessions* if you do not want the breakpoints to be saved and restored.

Class Debug

To customize the display of classes in the Class Browser:

1. Choose *Edit – Preferences* from the meny bar and select Class Debug from the list on the left side of the Preferences window.

2. Enable *Display UVM Base Classes* if you want to display them in the Class Browser.

3. Enable *Display Class Hierarchy Starting from Derive Classes* if you do want the class browser to include the derived classes.

4. Enable *Use Fully Quaified Class Names* if you want to the Class Browser to display the full names of class objects. When this option is disabled, the Class Browser displays short names.

5. From the *Members and Methods* menu, choose the types of objects that you want to display:

❏   *Show derived only*

❏   *Show inheritance as super*

❏   *Show flattened view*

Waveform Window

To customize the display of waveforms:

1. Choose *Edit – Preferences* from the menu bar and select *Waveform Window* from the list on the left side of the Preferences window.

2. Enable *Waveform banding* if you want the background of the Waveform window to alternate between bands of gray and black. Waveform banding can help you visually align signal names and values with their corresponding waveforms. Disable this option if you want the background to be all black.

3. Enable *Search Toolbar on by default* if you want to display the toolbar that lets you search for signal names, groups, markers, expressions, and values.

4. Enable *Show Times on Markers* if you want the current simulation time to appear in markers. You can choose not to display simulation times if you want to keep the size of markers as small as possible.

5. Enable *Show Times on Cursors* if you want the current simulation time to appear in cursors. You can choose not to display simulation times if you want to keep the size of cursors as small as possible.

6. Disable *Snap Cursor to Signal Transition* if you do not want cursors or makers to jump to the nearest signal transition as you move it through simulation time.

7. Enable *Zoom Out Full on Initial Add* if you want the initial view of the simulation database to include the entire simulation timeline. By default, SimVision displays a portion of the timeline. The amount of time displayed depends on the total amount of simulation time.

8. Enable *Enable Undo When Adding Signals* if you want to be able to click *Undo* to remove any signals that you have just added. This button is disabled by default, because an undo operation after you have added many signals can take a long time.

9. Enable *Show signal value tooltip* if you want SimVision to pop up a tooltip showing the signal value when you hover over a signal in the waveform area.

10. When you enable *Clear window on Load Signals*, SimVision replaces the contents of the current Waveform window with the signals that you add to it. When this option is disabled, signals are appended to the list of signals already displayed in the window. This option is disabled by default.

11. From the *Signal Radix* drop-down menu, choose the radix that you want to use when adding signals to the Waveform window. By default, the Waveform window displays values in the radix in which they were recorded to the simulation database.

12. Select one of the following actions to take when you double-click on a waveform:

- ❑ *No action*

- ❑ *Go to Cause*

- ❑ *Display in Trace Signals Sidebar*

- ❑ *Trace X in Trace Signals Sidebar*

- ❑ *Tcl Command in the SimVision Console*

- ❑ *Tcl Command in the Simulator Console*

If you choose *Tcl Command in the SimVision Console*, you must specify a SimVision command to execute. If you choose *Tcl Command in the Simulator Console*, you must specify a simulator command to execute.

Waveform Display

To set display options in the Waveform window:

1. Choose *Edit – Preferences* from the menu bar, expand *Waveform Window* in the list on the left side of the Preferences form, and choose *Display*.

2. Set the *Sequence Time Width* to the number of pixels you want to use between ticks when you expand sequence time. The default is 30 pixels.

3. Set the *Waveform height* for the number of pixels you want to use for the height of a digital waveform. The default is 12 pixels.

4. Set the *Waveform space* to the number of pixels you want to use between waveforms. The default is 6 pixels.

5. Set the *Transaction height* to the initial height of a transaction. The default is 3 times the height of a digital trace. The height of transactions is adjustable after they are displayed.

6. Set *Dynamic Arrays height* to the number of elements you inititally want to see when displaying dynamic arrays. By default, no elements are displayed. The height is adjustable after they are displayed.

Analog Waveform

To set the height of analog waveforms:

1. Choose *Edit – Preferences* from the menu bar, expand *Waveform Window* in the list on the left side of the Preferences form, then choose *Analog*.

2. Set the *Analog height* to the initial height of an analog waveform. The default is 3 times the height of a digital trace. The height of analog waveforms is adjustable after the waveform is displayed.

3. From the *Analog Format* menu, choose the format that you want to use when displaying analog signal values—either *Linear interpolation* or *Sample/Hold*.

4. Set the *Analog to digital threshold low* and *Analog to digital threshold high* fields to specify how analog signals are converted to digital signals. The low threshold specifies the value at which a signal is converted to 0; the high threshold specifies the value at which a signal is converted to 1.

5. Enable *Use data sampling* to reduce the number of data points when drawing analog waveforms. This improves performance but can remove interesting features of the waveform, such as spikes, when waveforms are zoomed out. This option is disabled by

default, and analog waveforms are drawn at each data point. This provides the most accuracy at all zoom levels, but can be slow when drawing waveforms with many data points.

6. Disable *Use automatic colors for Analog signals* if you do not want SimVision to assign a unique color to each analog signal that you add to the Waveform window.

Design Browser

To set general Design Browser preferences:

1. Choose *Edit – Preferences* from any SimVision window, then choose *Design Browser* from the list on the left side of the window.

2. Disable *Incremental Signal Filtering* if you do not want the Design Browser window to update the signal list incrementally. When this option is enabled, the signal list updates each time you enter a character of the search string in the signal list *Filter* field. When this option is disabled, the signal list is updated only after you press Return in the *Filter* field.

3. Enable *Show Module/Unit Names* if you want to display the module or design unit name next to each scope. By default, this option is off, and only the scope name is displayed.

   Note: *Show Module/Unit Names* does not apply to the *e* unit tree. The *e* unit tree displays the short names for units at all times.

4. Disable *Show UVM/OVM Verification Hierarchy* if you do not want to see UVM/OVM objects in the Design Browser. This option is enabled by default.

5. Enable *Show relative path to database* if you want the Design Browser to display the full path to opened databases. This is convenient when you have multiple databases opened and they have the same or similar names but are stored at different locations.

6. Disable *Show pop-up tool tips* if you do not want information about a button or icon displayed when you hover the cursor over the item.

7. Choose when you want to display signal values: *Never*, *Always*, and *When connected to simulator*.

8. Select one of the following actions to take when you double-click on a signal:

   ❑ *Expand/Collapse Arrays, etc.*

   ❑ *Send Object to Waveform window*

   ❑ *Send Object to Source Browser*

❑ *Break on Change*

❑ *Tcl Command in the SimVision Console*

❑ *Tcl command in the Simulator Console*

If you choose *Tcl Command in the SimVision Console*, you must specify a SimVision command to execute. If you choose *Tcl Command in the Simulator Console*, you must specify a simulator command to execute.

❑ *Group library cells into folders*

This lets you hide library cell instances at each level of the hierarchy, making it easier to navigate the design. See *Displaying and Hiding Library Cells in the Design Browser* for more details.

❑ *Find box searches only expanded nodes/items*

Performs searches in expanded nodes only. The option is off by default. See *Searching Only Expanded Nodes* for more details.

Design Browser Scope View

To set scope view preferences:

1. Choose *Edit – Preferences* from any SimVision window, expand *Design Browser* from the list on the left side of the window, then choose *Scope View*.

2. Choose the types of Verilog, VHDL, and SystemC, and *e* objects you want to display in the scope view.

3. Enable *Icons* to display icons that indicate object type, or disable this option to display only the object name.

4. Enable *Tree View* to display the scopes of the design hierarchically. Disable this option to display only the objects at the current level of hierarchy.

5. Choose whether you want to sort the objects in the scope view by *Name* or *Declaration type*.

Note: You might need to scroll the Preferences window to see the following option.

6. Enable *Track between target source browser and target watch list* if you want the Source Browser to display the source code for the scope that you select in the Design Browser sidebar.

Design Browser Signal List

To set signal list preferences:

1. Choose *Edit – Preferences* from any SimVision window, expand *Design Browser* from the list on the left side of the window, then choose *Signal List*.

2. Set the signal filter defaults for your design language by choosing the types of objects you want to display in the signal list. You can override these defaults by enabling and disabling the signal list filter buttons in the Design Browser window.

3. Select the radix that you want to use when displaying signal values.

4. Enable *Show Strength in Values* if you want to display the signal strength with the signal values.

5. Enable *Filter using regular expressions* if you want to include wildcard characters in the filter string.

6. Choose the order in which the signals should be displayed in the signal list—by name, declaration type, or the order in which they are declared.

Source Browser

To set Source Browser options:

1. Choose *Edit – Preferences* from any SimVision window, and choose *Source Browser* from the list on the left side of the window.

2. Enter the command to invoke the editor in the *Editor Command* field, using substitution strings to specify a file (`%F`) and a line number (`%L`), if your editor accepts those arguments. For example, to set `vi` as your editor, enter the following command:

```
xterm -e vi %F
```

Click *Default Editor* to reset the editor to the default.

3. Enter values for the following preferences:

❑ In the *Source Browser Tab Size* field, enter the size of a tab stop.

❑ In the *Maximum History Count* field, enter the number of scopes you want to save in the history list. See *Using the Source Browser* for information on using the history list.

These settings take effect when you click *OK* or *Apply.*

4. In the *Mouse Actions for Signals Double Click* field, choose the behavior you want when you double-click on a signal in the Source Browser. The drop-down menu provides several options, such as displaying the signal definition, or sending the signal to a specific window. The default double-click behavior is *Show Driver*.

   If you choose *Run a Tcl Command in the SimVision Console*, you must specify a SimVision command to execute. If you choose *Run a Tcl Command in the Simulator Console*, you must specify a simulator command to execute.

5. In the *Mouse Actions for Signals Drag and Drop* field, choose the behavior you want when you drop a signal into the Source Browser. The drop-down menu provides several options, such as displaying the signal definition, or sending the signal to a specific window.

   If you choose *Run a Tcl Command in the SimVision Console*, you must specify a SimVision command to execute. If you choose *Run a Tcl Command in the Simulator Console*, you must specify a simulator command to execute.

6. Disable the *Show Popup Tooltips on Signals in the Source Code* if you do not want the Source Browser to display the signal name, value, and probe status when you hover the cursor over a signal name. This information appears in a tooltip in the source area of the window.

7. Disable *Highlight Syntax in the Source Browser* if you do not want to display keywords, comments, and output strings in different colors. You can specify these colors in the Source Browser Colors Preferences tab.

Note: You may need to scroll or resize the window to see the following options.

8. Enable *Display values in the Source Browser* if you want the Source Browser to show object values on a separate line in between each line of source code.

9. Disable *Show Signal Strength in Values* if you do not want to see the strength of a signal value.

10. From the *Value Radix* drop-down list, select a radix for values displayed when you place the cursor over a signal. By default, the Source Browser displays values in the radix in which they were recorded.

Source Browser Colors

You can change the colors used by the Source Browser to display text elements such as comments, keywords, and strings in the source code.

To change the colors used by the Source Browser:

1. Choose *Edit – Preferences* from the menu bar, then select *Colors* from the list on the left side of the Preferences form, below *Source Browser*.

2. From the *Item* drop-down menu, select the text element whose color you want to change.

3. Click *Color* to open the *Color* menu. Choose a color from the menu, click *Default* to reset the text element to its default value, or click *More* to create a custom color with the Select a New Color form.

    When you select the new color, SimVision updates the sample in the Preferences form to show how the Source Browser will look when you apply your changes.

4. At any time, if you decide that you do not want to use the preferences you have set, click *Revert to Default Colors*.

Source Browser Files

You can specify the file types that the Source Browser recognizes when highlighting syntax.

To specify Source Browser file types:

1. Choose *Edit – Preferences* from the menu bar, then select *Files* from the list on the left side of the Preferences form, below *Source Browser*.

2. Click a row in the *Extensions* column, then add or remove a file extension.

Measurement Window

The Measurement window displays a table of measurements for selected signals. By default, the Measurement window includes no measurements, only the signal or variable name.

To choose the types of measurements displayed by default in a Measurement window:

1. Choose *Edit – Preferences* from the menu bar, then select *Measurement Window* from the list on the left side of the Preferences form.

    A list of *Available measurement* types is shown on the left, and a list of *Default measurements* is shown on the right.

2. Select the type of measurement you want to include in your Measurement window from the list of *Available measurements*, then click the right arrow to add your selections to the list of *Default measurements*. Select the type of measurement you want to remove from the list of *Default measurements*, then click the cross-hatch button.

Buttons to the right of the *Default measurements* list let you rearrange the list, as follows:

🔼	Move the selected measurement up in the list.
🔽	Move the selected measurement down in the list.
✖	Remove the selected measurement from the list of *Default measurements*.

In the Measurement window, *Default measurements* are displayed from left to right, beginning with the first measurement in the list. Therefore, moving a measurement up in the list has the effect of moving it to the left in the Measurement window. Conversely, moving the measurement down in the list, has the effect of moving it to the right in the window.

Schematic Tracer

To set Schematic Tracer options:

1. Choose *Edit – Preferences* from any SimVision window, and choose *Schematic Tracer* from the list on the left side of the window.

2. Choose when you want the Schematic Tracer to show signal values, as follows:

    ❏ *When connected to simulator*

    ❏ *Always*

    ❏ *Never*

3. Enable *Show RTL Logic when available* to display logic elements in all new Schematic Tracer windows, by default.

4. Disable *Show celldefine objects* if you do not want the Schematic tracer to display any objects declared with `celldefine` in the source code. These objects are displayed by default.

5. Enable *Show Cell Shapes, if available*, to display ASIC library cells using the shapes that you have specified in a cell map file. If you enable this option, you must also specify a filename in the *Cell Shapes Mapfile* field.

6. Enable *Fill instances added to the schematic* if you want the Schematic Tracer to automatically add any subscopes, blocks, and internal signals for the instances that you add to the schematic.

7. Enable *Fill parent module after an Ascend operation* if you want the Schematic Tracer to automatically add the subscopes, blocks, and internal signals of any parent modules encountered during an ascend operation.

8. Enable *Collapse sub module after an Ascend operation* if you do not want to see the details of submodules when you ascend to their parent module.

9. Enable *Highlight logic that has been sent to the Schematic* if you want to highlight objects that you add to the schematic. These highlights are cleared when you add new objects or when you choose *Edit – Clear Highlights* from the menu bar.

10. By default, the Schematic Tracer does not highlight any logic it adds as the result of a fill operation. Enable *Highlight new logic after a fill operation* if you want this logic highlighted.

11. Enable *Show selected scope in the status bar area* if you want the Schematic Tracer to display the full name of the selected scope in the status area of the window.

12. Enable *Show internal pins when using cell mapping* if you want the Schematic Tracer to display these pins.

13. Disable *Show thumbnail image of schematic in panner* if you do not want to see the thumbnail image in the panner. When you disable this preference, the panner is blank, except for the control box. You can still resize the box to zoom in and out, and move the box to scroll the display.

14. Disable *Center and Zoom after adding objects* if you do not want the Schematic Tracer to zoom in on the objects that you add.

15. In the *Cell Shapes Mapfile* field, specify the name of an ASIC library cell map file. This file defines the shapes that the Schematic Tracer uses when displaying library cells.

16. If you have instances that contain many cells and you want the cells to have a different color and line thickness, specify a number in the *Number of Cells to use special Inst symbol*. When an instance has more than that number of cells, the cells are drawn in a special color and line thickness. This can help you identify those cells more easily. The default, 0, draws the cells with the same color and line thickness as the instance.

17. In the *Don't expand buses wider than field*, specify the maximum size of a bus that you want to be able to expand. The Schematic Tracer will not expand buses larger than the specified size. The default is 32765.

18. From the *Double Click* menu, choose the action you want to take when you double-click on a schematic element. If you choose *Tcl Command in the SimVision Console*, you

must specify a SimVision command to execute. If you choose *Tcl Command in the Simulator Console*, you must specify a simulator command to execute.

Schematic Tracer Colors

To choose the colors used in the Schematic Tracer window:

1. Choose *Edit – Preferences* from any SimVision window. If necessary, expand the *Schematic Tracer* entry on the left side of the window, then choose *Colors*.

2. From the *Item* drop-down list, choose the type of schematic element whose color you want to change.

3. Click *Color* to open the *Color* menu. Choose a color from the menu, or click *Default* to reset the text element to its default value, or click *More* to create a custom color with the Select a New Color form.

When you select the new color, SimVision updates the sample in the Preferences form to show how the SchematicTracer will look when you apply your changes.

4. At any time, if you decide that you do not want to use the preferences you have set, click *Revert to Default Colors*.

Schematic Tracer Logic Cone

To specify how the Schematic Tracer performs trace operations:

1. Choose *Edit – Preferences* from any SimVision window. If necessary, expand the *Schematic Tracer* entry on the left side of the window, then choose *Logic Cone*.

2. Set *Max number of levels* to the number of levels of hierarchy that you want to trace before stopping. You might want to set this preference to some number when you have a large design with many levels of hierarchy. The default value is 0, which removes the limitation and tracing continues through all levels of the hierarchy.

3. Enable *Stop at Flops/Latches* if you want tracing to stop when the Schematic Tracer reaches a flip-flop or latch. By default, it traces through these objects.

4. Enable *Stop at module boundaries* if you want the Schematic Tracer to stop at module boundaries. By default, it traces through module boundaries.

5. Enable *Stop at primitives* if you want the Schematic Tracer to stop when it reaches a primitive. By default, it traces through these objects.

FSM Window

To choose the colors used in the FSM window:

1. Choose *Edit – Preferences* from any SimVision window. If necessary, expand the FSM entry on the left side of the window, then choose *Colors*.

2. From the *Item* drop-down list, choose the type of FSM element whose color you want to change.

3. Click *Color* to open the *Color* menu. Choose a color from the menu, or click *Default* to rest the element to its default value, or click *More* to create a custom color with the Select a new Color form.

    When you select the new color, SimVision updates the sample in the Preferences form to show how the FSM will look when you apply your changes.

4. At any time, if you decide that you do not want to use the preferences you have set, click *Revert to Default Colors*.

Memory Viewer

To set the Memory Viewer preferences:

1. Choose *Edit – Preferences* from any SimVision window, and choose *Memory Viewer* on the left side of the window.

2. From the *Radix* drop-down menus, choose the radix that you want the Memory Viewer to use when displaying values and addresses. Choose *As Recorded* to display the values and addresses in the radix in which they were recorded in the simulation database.

3. Enable *Size to fit* to let the Memory Viewer adjust the width and number of columns used to display memories, or disable that button and specify a fixed number of columns.

4. From the *Font Size* menu, choose to display the memory values in *Small*, *Medium*, or *Large* font.

5. In the *Double Click* field, specify the action that you want the Memory Viewer to take when you double-click on a memory element. The *Edit Deposit* action is the default, which lets you specify a value to deposit at the memory location.

    If you choose *Tcl Command in the SimVision Console*, you must specify a SimVision command to execute. If you choose *Tcl Command in the Simulator Console*, you must specify a simulator command to execute.

Memory Viewer Colors

To choose the colors used in Memory Viewer windows:

1. Choose *Edit – Preferences* from any SimVision window. If necessary, expand the *Memory Viewer* entry on the left side of the window, then choose *Colors*.

2. From the *Item* drop-down menu, select the type of value whose color you want to change:

    ❑   *Value has changed*—Choose the foreground and background colors to indicate that the value changed during the current sample time, but its ending value is the same as its starting value.

    ❑   *Value is changing*—Choose the foreground and background colors to indicate that the value is changing during the current sampled time.

    ❑   *Value is different*—Choose the foreground and background colors to indicate that the value changed during the current sample time, and its ending value is different from its starting value.

3. Click *Color* to open the *Color* menu. Choose a color from the menu, click *Default* to reset the text element to its default value, or click *More* to create a custom color with the Select a New Color form.

    When you select the new color, SimVision updates the sample in the Preferences form to show how the Memory Viewer will look when you apply your changes.

4. At any time, if you decide that you do not want to use the preferences you have set, click *Revert to Default Colors*.

Formal Verifier

To set Formal Verifier options:

1. Choose *Edit – Preferences* from any SimVision window, then choose *Formal Verifier* from the list on the left side of the window.

2. Set the *Debug/Witness Waveform Option* from the drop-down menu to *Static*, *Dynamic*, or *Full.*

Simulation Cycle Debug

Note: The Simulation Cycle Debugger is not available in Verilog-XL.

To set Simulation Cycle Debugger preferences:

1. Choose *Edit – Preferences* from any SimVision window, and choose *Simulation Cycle Debug Options* from the list on the left side of the window.

2. Specify the number of process or behavioral blocks you want to display in the Simulation Cycle Debugger. The maximum number is 100, because displaying more items can result in poor performance.

Plug-Ins

SimVision provides plug-in applications, which you can add to or remove from your SimVision windows.

To add or remove a plug-in application:

1. Choose *Edit – Preferences* from any SimVision window, and open the *Plug-Ins* page.

2. Enable *SV Data Browser* to add the *SystemVerilog Data Browser* to the *Windows – Tools* menu. For information, see Using the Data Browser, see *Debugging SystemVerilog*.

3. Enable *Group Scope* to add the *Group Scope* button, , to your Design Browser and Waveform windows. This button creates a group containing all of the signals in the scope that you have selected. For more information, see Creating a Group for a Scope.

4. Enable *Interleave and Compare* to add an *Interleave* button to your Design Browser and Waveform windows. This button creates a comparison of two signals that you have selected and displays it in the Waveform window. For more information about SimCompare, see Using SimCompare.

For information on how to write plug-in applications and add them to the Preferences page, see Writing Plug-In Applications.

Layouts

To specify preferences for custom window layouts:

1. Choose *Edit – Preferences* from any SimVision window, and open the *Layouts* page.

2. Enable *Ignore properties window when saving layouts* if you do not want the Properties window to be restored with this layout. This option is disabled by default.

3. Disable *Ignore iconfied windows when saving layouts* if you do not want iconified windows to be restored with this layout. This option is enabled by default.

4. Disable *Adjust window sizes for resolution when restoring layouts* if you do not want the windows to be resized when they are restored to a window with a different resolution. This option is enabled by default.

5. Choose from a list of available layouts to specify the *Default layout to load when starting SimVision*. This list includes the built-in layouts supplied by SimVision and any custom layouts that you have defined. You can override this setting with the `-layout` command-line option when starting SimVision.

Transaction Explorer

The Transaction Explorer preferences tabs let you customize the behavior of the TxE graphical user interface.

For more information, see <u>TxE Graphical User Interface</u> in the *Transaction Explorer Reference*.

4

Customizing the Layout of SimVision Windows

SimVision provides built-in layouts for particular uses, such as SystemC/C/C++ debugging and embedded software tracing. If these built-in layouts do not meet your needs, you can create your own custom window layouts.

For example, you might have a set of windows that you typically use together. You might want to re-size them or position them in relation to each other. You might want to link them, so that when you move one window, they all maintain their relative positions. When you have found the configuration of windows that works best for you, you can save your layout and restore it in subsequent SimVision sessions.

A window layout defines the following characteristics of each window in the configuration:

■ Window name

■ Window geometry

■ Linked state of the window, that is, whether the window should maintain its relative position to other windows

■ Sidebar information, such as:

❑ Sidebar state—hidden, partially visible, or fully visible

❑ Currently-selected sidebar

❑ Position and size of any sidebars that have been torn from the window

■ Toolbar information, such as:

❑ Position of the toolbars in each window

❑ State of toolbar buttons (hidden or shown)

■ State of the window—visible or iconified

■   Some window-specific information, such as the state of the callstack in the Source Browser

■   Screen resolution, to ensure that the window is sized properly when it is restored

Custom layouts that you create are saved in your `$HOME/.simvision` directory, in a subdirectory named `layout`. Therefore, the layouts you define are available in any SimVision session that you run under `$HOME`.

For More Information

■   <u>Creating a Window Layout</u> on page 70

■   <u>Using a Window Layout</u> on page 70

■   <u>Setting a Default Layout</u> on page 71

■   <u>Managing Layouts in the Properties Window</u> on page 71

Creating a Window Layout

To create a layout:

1. Open any windows you want to use together, resize them, and position them as you would like them to appear.

2. Enable the *Link*, ⊞ , in any windows that should maintain their relative positions when they are moved.

3. Choose *Windows – Layout – Save layout*. SimVision opens the Save New Layout form.

4. Enter a name for your layout and click *OK*.

Using a Window Layout

You can choose a window layout when you start SimVision, switch layouts at any time during a SimVision session, or set a default layout to be used whenever you run SimVision.

To choose a layout at startup:

➡   Invoke `simvsion` with the `-layout` option and specify the layout name. For example, to use the SystemC/C/C++ Debugging layout:

```
simvision -layout SystemC
```

To use a custom layout, called `mylayout`:

```
simvision -layout mylayout
```

If you do not specify a layout at startup, SimVision uses the layout you used in the most recent session.

To switch layouts during a session:

➡ Choose the name of the layout from the *Windows – Layout* menu in any SimVision window. This menu displays up to 20 of the most recently-saved layouts. To see all of the layouts, use the Layouts page of the Properties window,

To choose a layout from the Properties window:

➡ Open the *Layouts* page of the Properties window, select a layout, and click *Use selected layout*, ✔ .

Setting a Default Layout

You can make a layout your default, so that it is loaded automatically at startup.

To set a default layout from the Preferences window:

1. Choose *Edit – Preferences* from any SimVision window and open the *Layouts* page.

2. Select a layout from the *Default layout to load* drop-down menu. The default layout is used whenever you run SimVision, unless you specify a different layout at startup.

To set a default layout from the Properties window:

1. Open the *Layouts* page of the Properties window.

2. Enable the *Default* button for the layout. This layout is also set as your default layout in the Preferences window.

Managing Layouts in the Properties Window

To open the Layouts page of the Properties window, do one of the following:

➡ Open the drop-down menu on the *Properties* button and choose *Layouts*.

➡ Click *Properties* and select *Layouts* on the left side of the window.

➡ Choose *Windows – Layouts – Layout Properties*.

The *Layouts* page of the Properties window contains a list of all layouts that you have defined, as well as the Simvision bult-in layouts, as shown in <u>Figure 4-1</u> on page 72.

Figure 4-1 Layouts Page of the Properties Window
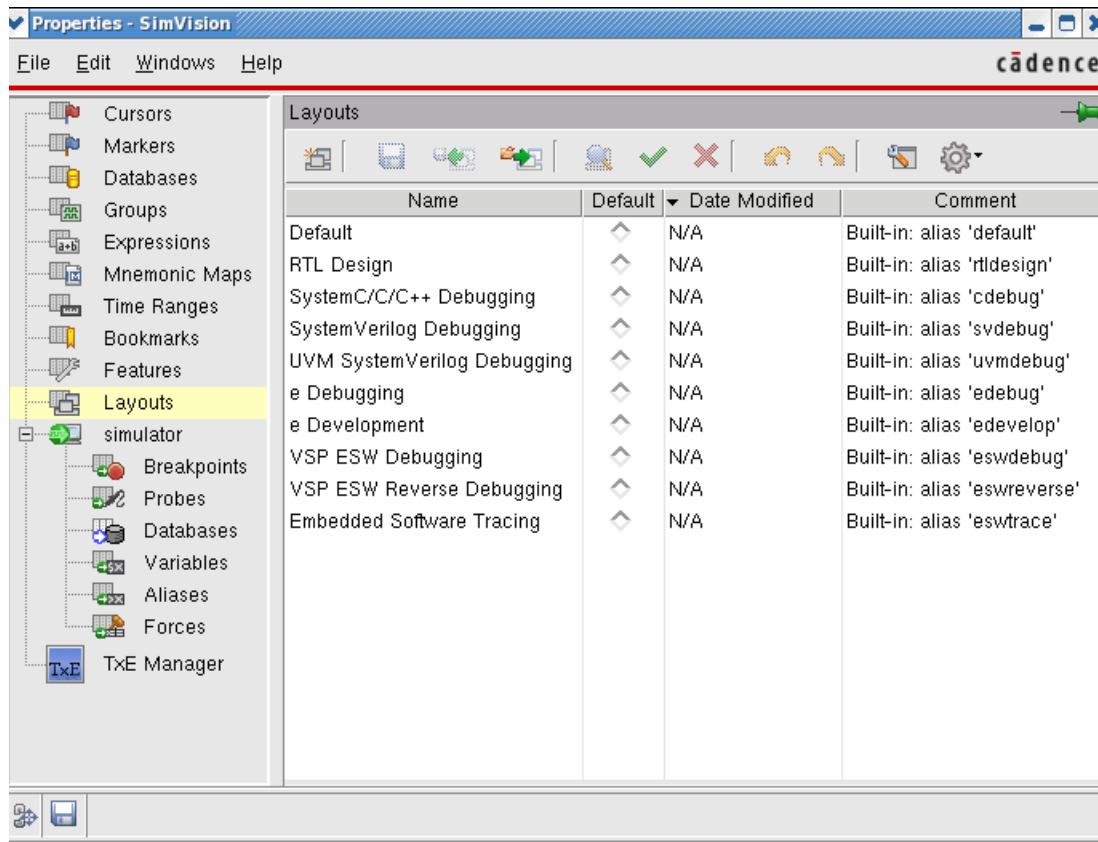
For each layout, the Properties window displays its name, a button to indicate whether it is the default layout, the date it was last modified, and a comment. You can edit the *Name* field to change a layout name, enable or disable the *Default* button, and edit the *Comment* field. These changes become effective immediately.

The toolbar at the top of the page lets you perform the following functions:

Save the current window configuration as a new custom layout.

Overwrite the selected layout with the current window configuration (custom layouts only).

Export a layout to a file

Import a layout from a file

Shows a preview of the selected layout

Use the selected layout as the current layout

Delete the selected layout (custom layouts only)

Undo the last action

Redo the last action

Open the Windows form to let you configure the window inthe layout

Set preferences for saving and restoring layouts (See <u>Layouts</u> on page 66 for information on layout preferences)

For More Information

■ <u>Creating a Layout in the Properties Window</u> on page 73

■ <u>Overwriting a Layout Definition</u> on page 74

■ <u>Importing and Exporting Layouts</u> on page 74

■ <u>Previewing a Layout</u> on page 74

■ <u>Configuring the Windows in a Layout</u> on page 76

Creating a Layout in the Properties Window

To create a layout from the Properties window:

1. Open any windows you want to use together, resize them, and position them as you would like them to appear.

2. Open the *Layouts* page of the Properties window.

3. Click *New Layout*, 🔲 . An entry is added to the list of layouts, named *New Layout*.

4. Edit the layout name and press `Return`.

5. If you want this to be the default layout, enable the *Default* button.

6. Edit the *Comment* field, if desired. The comment appears only in the Properties page, but it can be helpful if you have many layouts, or if you are sharing layouts with others.

Overwriting a Layout Definition

To overwrite a layout:

1. Open any windows you want to use together, resize them, and position them as you would like them to appear.

2. Open the *Layouts* page of the Properties window and select the layout you want to overwrite.

3. Click *Overwrite layout*, 🖫 .

Importing and Exporting Layouts

Custom layouts can be shared by importing and exporting them.

To export a layout:

1. Select the layout in the Properties window and click *Export*, ▣← .

2. In the Export Layout form, navigate to the directory where you want to store the layout file, enter a name in the *File name* field, and click *Save*. SimVision layout files have a default file extension of `.svl`.

To import a layout:

1. Click Import, ▣← .

2. In the Import Layout form, navigate to the directory that contains the layout file, select the file from the list, and click *Open*.

    Note: The Import Layout form filters the file list to show directories and only those files with the `.svl` extension. If the layout file you want to import does not have the default extension, change *Files of type* to *All Files (*)*.

After a layout is imported, its name appears in the Layouts page of the Properties window.

Previewing a Layout

If you have several layouts to choose from, you can preview them before deciding which one to use.
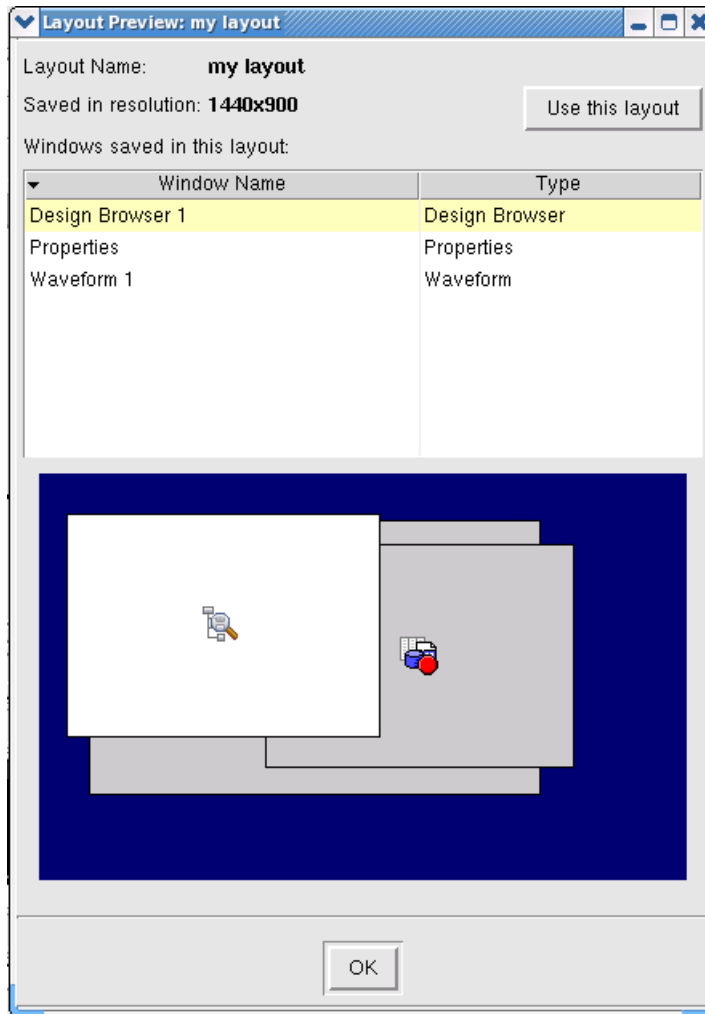
To preview a layout:

1. Select the layout in the Properties window and click *Preview*, ![icon] . This opens the Layout Preview window, as shown in <u>Figure 4-2</u> on page 75.

Figure 4-2 Layout Preview Window

The Layout Preview window contains the following information:

❑   Layout name

❑   Screen resolution used when the layout was saved. SimVision uses this resolution to resize the windows when they are restored, if you have enabled *Adjust window geometries for resolution* in the Layout Preferences window.

❑   A list of the windows that make up the layout, and their types. When you select a window name in the list, the window is brought forward in the window canvas below.

❑ A canvas that displays the relative position and size of the windows in the layout. When you click on a window in the canvas, it is brought to the front, and its name is selected in the window list above.

2. Examine the layout, and if you decide you want to use it, click *Use this layout*.

Configuring the Windows in a Layout

You might want to configure the windows that are open before creating a new layout or before changing a layout.
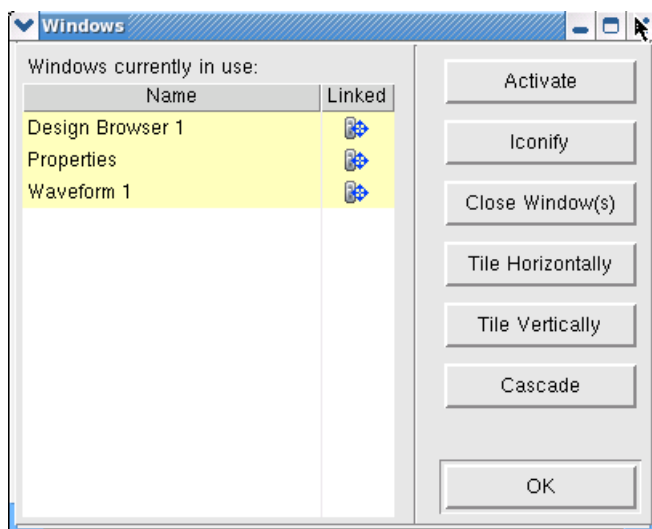
To configure the windows that are open:

1. From the Layouts page of the Properties window, click *Configure application windows*, 🖎 . This opens the Windows form, as shown in <u>Figure 4-3</u> on page 76. In this figure, the Design Browser, Waveform, and Watch windows are linked.

💡 *Tip*

> You can also open the Windows form from any SimVision window by choosing *Windows – Windows*.

Figure 4-3 Windows Form

The Windows form displays a list of all open windows.

2. Use the form to configure the windows, as follows:

❑ Select a window and click *Activate* to bring the window to the foreground.

❑ Select a window and click *Iconify* to minimize, the window. You can set a preference to include or exclude iconfied windows when saving a layout. In this way, you can use *Iconify* to add or remove a window from a layout.

❑ Select one or more windows and click *Close Window(s)*.

❑ Select two or more windows and click *Tile Horizontally*, *Tile Vertically*, or *Cascade* to arrange the windows accordingly.

❑ Enable the *Link*, , for any windows that should maintain their relative positions when they are moved.

❑ Click *OK* when you are satisfied with your window settings.

5

Customizing Toolbars

SimVision lets you choose the toolbars and buttons that you want to display in any SimVision window. You can also create new toolbars and buttons to perform functions that are not already available through the SimVision toolbars.

For More Information

■ Adding and Removing Toolbars and Buttons on page 79

■ Creating a Custom Toolbar on page 81

■ Adding a Custom Toolbar to Other Windows on page 85

■ Writing a Window-Independent Script on page 85

■ Modifying Custom Toolbars on page 86

## Adding and Removing Toolbars and Buttons

Each window type has its own set of toolbars and buttons. However, the basic procedure for adding and removing toolbars and buttons is the same, regardless of the window type.

To add or remove a toolbar or button:

1. Choose *View – Toolbars – Customize* from any SimVision window. This opens the Customize Toolbars form, as shown in Figure 5-1 on page 80.

Figure 5-1 Customize Toolbars Form
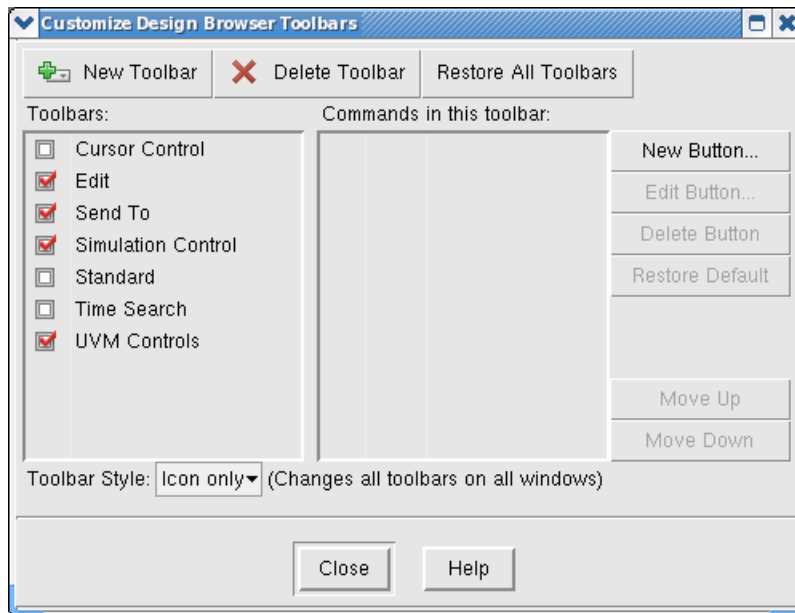
2. Enable or disable the check mark for a toolbar to add or remove it in the window.

Tip

> You can also add and remove toolbars by right-clicking over a blank space or separator in the toolbar area, and then enabling and disabling toolbars in the pop-up menu.

3. Highlight a toolbar in the *Toolbars* list, and the Customize Toolbars form displays the commands available in that toolbar, including the buttons that represent those commands, as shown in Figure 5-2 on page 81.

Figure 5-2 Displaying Commands for a Toolbar
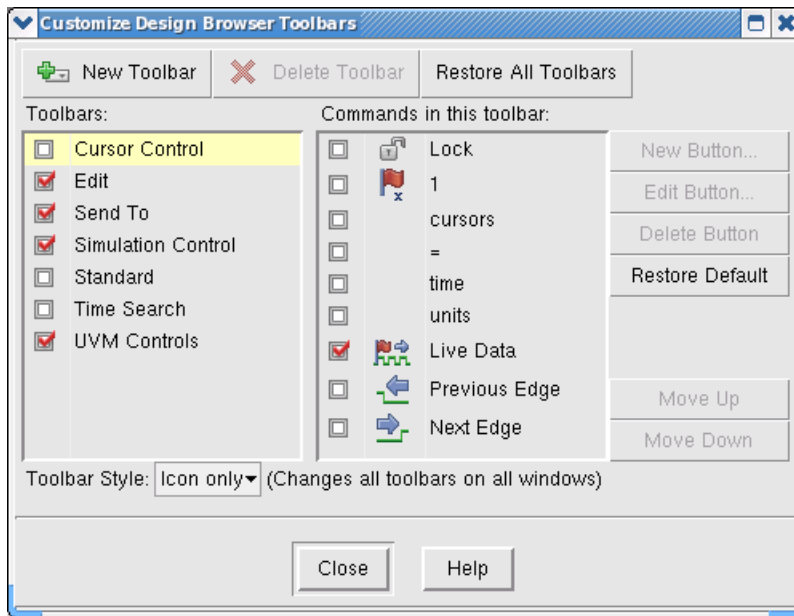
4. Enable or disable the check mark for a command to add or remove its button in the toolbar for that window. At any time, you can return the toolbar's button settings to their default values by clicking the *Restore Default* button.

5. At any time, you can restore all toolbars to their default settings by clicking *Restore all Toolbars*. If a toolbar that you have defined is in one of the default toolbar locations, your toolbar will be moved.

When you modify the toolbars and buttons for a window type, SimVision applies those changes to the current window and to any new windows of that type. Any other windows of that type that are already opened are not affected by your changes.

Creating a Custom Toolbar

The amount of customization that you can perform on the SimVision toolbars is limited. That is, you can enable and disable buttons on the toolbar, but you cannot add or delete buttons and you cannot rearrange their order in the toolbar. However, you do have complete control over the contents and layout of the toolbars that you create.

To create a toolbar:

1. Click *New Toolbar*, and SimVision adds a *User Toolbar* to the list of toolbars, as shown in <u>Figure 5-3</u> on page 82. At this point, the toolbar is empty; no commands have been defined.
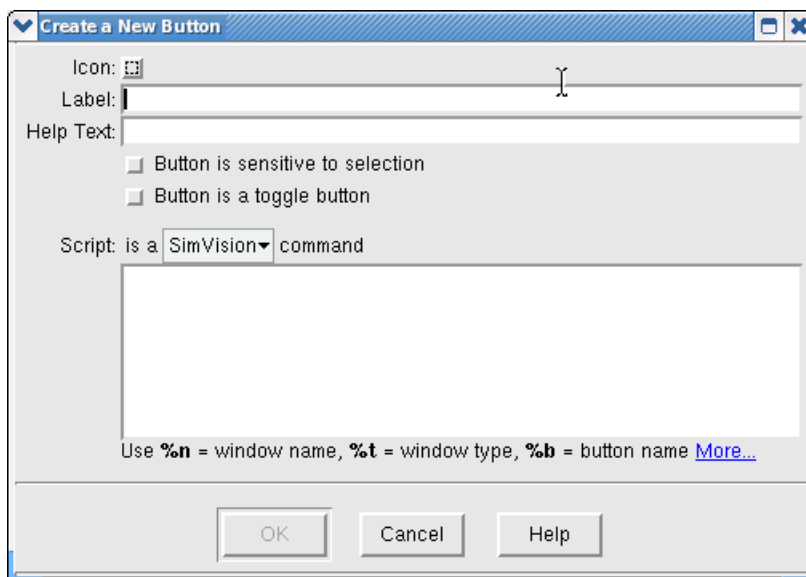
Figure 5-3 Adding a User Toolbar to the List of Toolbars
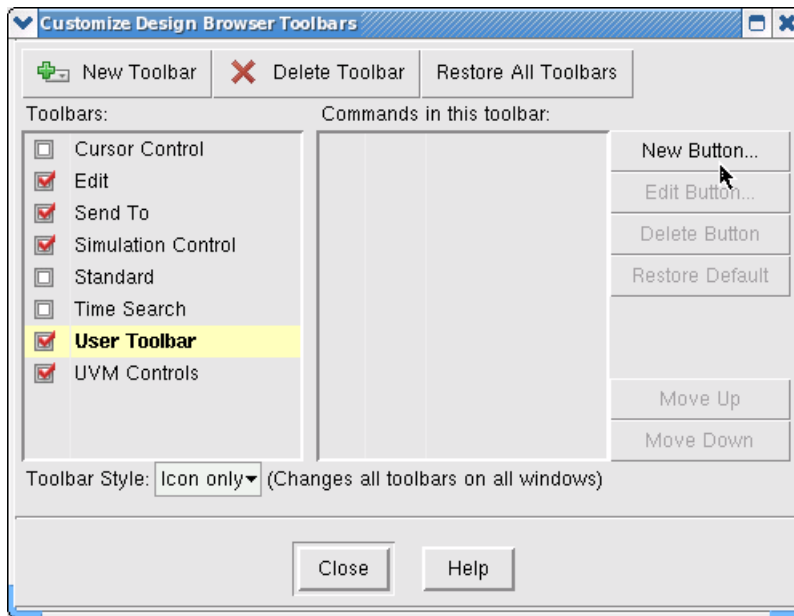
You can create additional toolbars by clicking *New Toolbar*. Additional toolbars are assigned the names *User Toolbar_1*, *User Toolbar_2*, and so on.

2. Click *New Button* to define a button for the toolbar. This opens the Create a New Button form, as shown in

Figure 5-4 Create New Button Form

3. Click *Icon*. This displays a menu of icons.

Choose an icon in one of the following ways:

❑ Choose an icon from the menu.

❑ Click *No Icon* to specify that you do not want to use an icon for the button. In this case, the label is used for the button.

❑ Click *More* to use your own GIF file for an icon. If you use your own GIF file, SimVision sizes the icon for you.

4. In the *Label* field, enter the string you want to display below the icon when the *Toolbar Style* is set to *Text only* or *Both* in the General Options tab of the Preferences form.

5. In the *Help Text* field, enter the string you want to display in the status area of the window and in a tooltip when the mouse cursor hovers over the button.

6. Enable *Button is sensitive to selection* if this button is enabled when objects have been selected in a SimVision window. Otherwise, the button is disabled.

7. Enable *Button is a toggle button* if you want this button to toggle. That is, the button appears to push in when the user clicks it once, then appears to pop out when the user clicks it again. Otherwise, this button appears to push in and pop out with a single mouse click.

Note: You can set this option only at the time the button is created. After that, you cannot change this option setting.

8. In the *Script* drop-down list, choose the console that you want to send the button commands to, as follows:

SimVision	Sends the commands to the *SimVision* console, which accepts SimVision commands, described in the *SimVision Tcl Command Reference*.
Simulator	Sends the commands to the *simulator* console, which accepts simulator commands, described in the *Help* for your simulator.
gdb	Sends commands to the *gdb* console, which accepts `gdb` debugger commands. See the *SystemC Simulation User Guide* for information on GDB support.

Tip

You can send Tcl/Tk commands to the *SimVision* console, to the *simulator* console for the Incisive simulator, or to the *Specman* console. You cannot send Tcl/Tk commands to the *gdb* console or to the *simulator* console for the Verilog-XL simulator.

9. In the text area for the *Script* field, enter the commands that you want to send to the specified console. For example, if you want to send a `describe` command to the simulator when you click the button, enter the following command in the text area:

```
describe test_drink.top.dispense test_drink.top.vending.current_state
```

You can use the substitution strings listed in Table 5-1 on page 84 to pass arguments to your commands.

Table 5-1 Substitution Strings for Command Scripts

`%b`   Button name

`%c`   All selected scopes

`%i`   All selected signals

`%n`   Window name

`%s`   All selected items

`%t`   Window type

`%{prompt "`*string*`"}`

            Opens a dialog box and prompts the user for a string. The string is passed as an argument to the command.

For example, the following Tcl command uses `%n` to pass the window name to the `puts` command:

```
puts "%n"
```

You could also prompt the user for the name of a signal, then pass that name to the `describe` command, as follows:

```
describe %{prompt "Signal"}
```

10. Click *OK* to add the button to the Commands area of the Customize Toolbars form.

11. Click *Close*. The button is added to the toolbar.

Tip

> If the toolbar does not appear in the window, open the Customize Toolbars form again, and make sure the toolbar is enabled in the list of toolbars.

Adding a Custom Toolbar to Other Windows

The toolbars that you create can be added to any SimVision window.

To add your toolbars to other SimVision windows:

1. In the window to which you want to add the toolbar, choose *View – Toolbars – Customize*. This opens the Customize Toolbars form for that window type.

2. Click *New Toolbar*, and SimVision adds the user toolbars that you have created to the Toolbars list.

3. Select the user toolbar that you want to add to the window. The buttons that you created for that toolbar are listed in the Commands area of the form.

4. Enable the check mark for the buttons that you want to display in the toolbar, then click *Close*. The toolbar and buttons now appear in the window.

Writing a Window-Independent Script

SimVision commands are window-specific. That is, a command to invoke a SimVision menu entry must be invoked from a particular window, such as a Design Browser or Waveform window. For example, suppose you want to create a button to save the user environment and then reinvoke the simulator. If you wrote the script in the following way, it would be tied to a particular window:

```
browser invoke -using "Design Browser 1" "File>SaveCommandScript"
browser invoke -using "Design Browser 1" "Simulation>ReinvokeSimulator"
```

As written, this script will work only if a Design Browser window named "Design Browser 1" exists.

However, there is no way to guarantee that a particular window exists when the user runs the script. You need to write the script so that it can run in any window.

Here are two ways to do this:

■ You can use the Tcl `set` command to return the type of the current window, as follows:

```
set type [window type %n]
```

The `%n` argument passes the name of the current window to the `window type` command. This command returns the type of the window, such as `browser` or `waveform`.

The window type is also the name of the command that you use to invoke the *Save Command Script* and *Reinvoke Simulator* commands. You can use the value of the `type` variable as the command name. For example:

```
$type invoke -using %n "File>SaveCommandScript"
$type invoke -using %n "Simulation>ReinvokeSimulator"
```

■ You can use the `window` command, as follows:

```
window invoke %n -type menu "File>SaveCommandScript"
window invoke %n -type menu "Simulation>ReinvokeSimulator"
```

The `window` command can invoke a command for any window type, specified by the `%n` argument.

Modifying Custom Toolbars

You can modify the toolbars that you create, as follows:

➤ Remove the check mark for a button to remove it from the toolbar. This does not delete the button definition. The button is still available to other windows that contain this toolbar.

➤ Click *Delete Button* to delete a button, if it does not appear in another toolbar. If the button is enabled and it appears in other toolbars, *Delete Button* disables the button. If you try to delete a disabled button that appears in other toolbars, SimVision displays an error message.

➤ Select a button and click *Edit Button* to modify the button definition in the Edit Button form. When you edit a button, it is changed in every window in which it appears.

➤ Select a button and click *Move Up* or *Move Down* to rearrange the order in which the buttons appear in the toolbar.