SIEMENS EDA

Questa® SIM GUI Reference Manual
Including Support for Questa Base

Software Version 2024.3
Document Revision 9.2

SIEMENS

About Siemens Digital Industries Software

Siemens Digital Industries Software is a global leader in the growing field of product lifecycle management (PLM), manufacturing operations management (MOM), and electronic design automation (EDA) software, hardware, and services. Siemens works with more than 100,000 customers, leading the digitalization of their planning and manufacturing processes. At Siemens Digital Industries Software, we blur the boundaries between industry domains by integrating the virtual and physical, hardware and software, design and manufacturing worlds. With the rapid pace of innovation, digitalization is no longer tomorrow's idea. We take what the future promises tomorrow and make it real for our customers today. Where today meets tomorrow. Our culture encourages creativity, welcomes fresh thinking and focuses on growth, so our people, our business, and our customers can achieve their full potential.

Support Center: support.sw.siemens.com
Send Feedback on Documentation: support.sw.siemens.com/doc_feedback_form

Revision History ISO-26262

Revision	Changes	Status/ Date
9.2	Modifications to improve the readability and comprehension of the content. Approved by Matt Laney. All technical enhancements, changes, and fixes listed in the Release Notes are reflected in this document. Approved by Amr Hany.	Released September 2024
9.1	Modifications to improve the readability and comprehension of the content. Approved by Matt Laney. All technical enhancements, changes, and fixes listed in the Release Notes are reflected in this document. Approved by Tim Peeke.	Released June 2024
9.0	Modifications to improve the readability and comprehension of the content. Approved by Matt Laney. All technical enhancements, changes, and fixes listed in the Release Notes are reflected in this document. Approved by Tim Peeke.	Released February 2024

Author: In-house procedures and working practices require multiple authors for documents. All associated authors for each topic within this document are tracked within the Siemens documentation source. For specific topic authors, contact the Siemens Digital Industries Software documentation department.

Revision History: Released documents include a revision history of up to four revisions. For earlier revision history, refer to earlier releases of documentation on Support Center.

Table of Contents

Chapter 4
Window Reference..95

Chapter 5

Chapter 6
GUI Customization..309

Chapter 7
GUI Preferences...323

List of Figures

List of Tables

Questa SIM GUI Reference Manual

Chapter 1
Overview

The Questa SIM graphical user interface (GUI) provides access to numerous debugging tools and windows that enable you to analyze different parts of your design. All windows initially display within the Questa SIM Main window.

Figure 1. Graphical User Interface
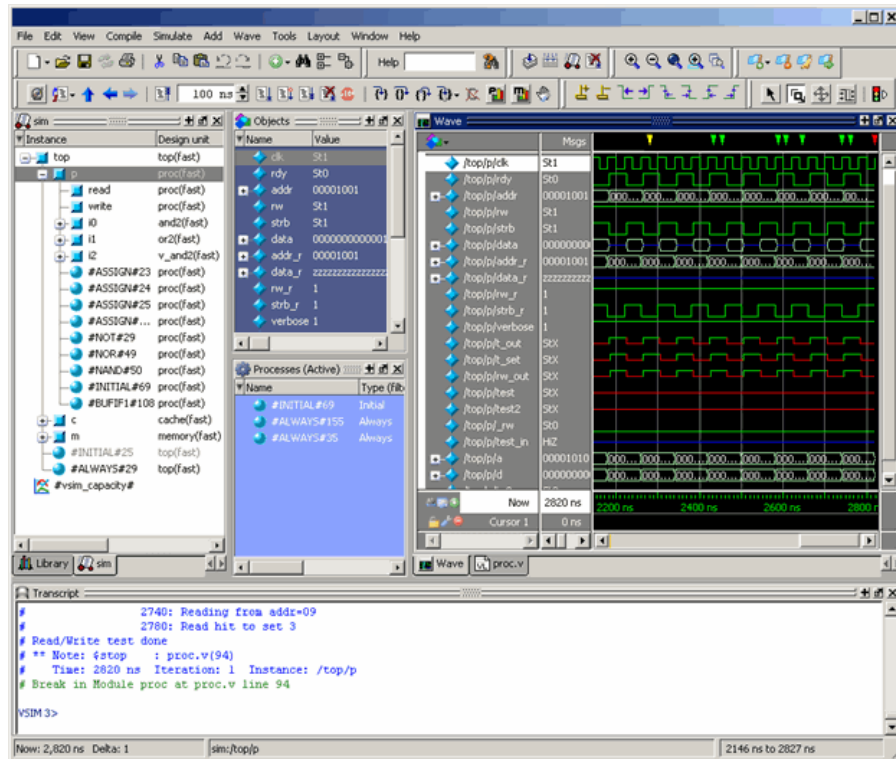
General GUI Features

General Visual Elements

General GUI Features

This section describes tasks common to more than one individual window and is organized into the following categories:

Window Management

The following tasks define actions you can take with the various windows.

Saving the Layout Upon Exit

By default when you exit Questa SIM, the current layout is saved for a given design so that it appears the same the next time you invoke the tool.

Resetting the Window Layout to the Default

The windows are customizable in that you can position and size them as you see fit, and Questa SIM will remember your settings upon subsequent invocations. You can restore Questa SIM windows and panes to their original settings by selecting **Layout > Reset** in the menu bar.

Copying Text from a Window Header

You can copy the title text in a window header by selecting it and right-clicking to display a popup menu. This is useful for copying the file name of a source file for use elsewhere

Selecting the Active Window

When the title bar of a window is highlighted - solid blue - it is the active window. All menu selections will correspond to this active window. You can change the active window in either of the following ways:

- (default) Click anywhere in a window or on its title bar.

- Move the mouse pointer into the window.

  To turn on this feature, select **Window > FocusFollowsMouse**. Default time delay for activating a window after the mouse cursor has entered the window is 300ms. You can change the time delay with the PrefMain(FFMDelay) preference variable.

Window Arrangement

The GUI provides features for moving and grouping the various windows.

Moving a Window or Tab Group

Relocating a window or tab group to a new location within the Main window.

Procedure

1. Click on the header handle in the title bar of the window or tab group.

Figure 2. Window Header Handle

2. Drag, without releasing the mouse button, the window or tab group to a different area of the Main window

   Wherever you move your mouse you will see a dark blue outline that previews where the window will be placed.

   If the preview outline is a rectangle centered within a window, it indicates that you will convert the window or tab group into new tabs within the highlighted window.

3. Release the mouse button to complete the move.

Moving a Tab out of a Tab Group

Removing a window from a tab group.

Procedure

1. Click on the tab handle that you want to move.

Figure 3. Tab Handle

2. Drag, without releasing the mouse button, the tab to a different area of the Main window

   Wherever you move your mouse you will see a dark blue outline that previews where the tab will be placed.

If the preview outline is a rectangle centered within a window, it indicates that you will move the tab into the highlighted window.

3. Release the mouse button to complete the move.

Undocking a Window from the Main Window

You can move a window to exist outside of the main window.

Procedure

Do either of the following:

- Follow the steps in Moving a Window or Tab Group, but drag the window outside of the Main window.

- Click on the Dock/Undock button for the window.

Figure 4. Window Undock Button

Column-Based Windows

This section describes tasks related to column-based windows throughout the GUI.

Customizing the Column Views

Customizing the Column Views

You can customize the display of columns column-based windows, and then save these views for later use.

Procedure

1. Right-click in the column headings and select **Configure Column Layout**. This displays the Configure Column Layout dialog box.

2. Click **Create**. This displays the Create Column Layout dialog box.

3. For Layout Name, enter a name for the layout for future reference.

4. For Column Selections, move columns to your desired state.

5. Click **OK**. This adds your new layout to the Layouts list.

6. Click **Done**.

Results

After applying your selections, the rearranged columns and custom layouts are saved and appear when you next open that column view in the window.

Bookmarks

You can create bookmarks that allow you to return to a specific view or place in your design for some of the windows. The bookmarks you make can be saved and automatically restored. Some of the windows that allow bookmarking include the Structure, Files, Wave, and Objects windows.

    Bookmark Actions

    Bookmarks Management

Bookmark Actions

The Bookmarks toolbar and the Bookmarks menu give you access to several bookmarking features.

- Add Bookmarks — Bookmarks are added to an active window by selecting **Bookmarks > Add Bookmark** or by clicking the **Add Bookmark** button. You will be prompted to automatically save and restore your bookmarks when you set the first bookmark. You can change the automatic save and restore settings in the Figure 6.

- Add Custom — Selecting **Add Custom** opens the **New Bookmark** dialog box with the context field(s) populated and a field for specifying an alias for the bookmark. Click and hold the **Add Bookmark** button to access this feature from the **Bookmarks** toolbar.

> **Note:**
> Aliases are mapped to the window in which a bookmark is set. You can use the same alias for different bookmarks as long as each alias is assigned to a bookmark set in a different window.

- Deleting Bookmarks — You can choose to delete the bookmarks from the currently active window or from all windows.

- Manage Bookmarks — Opens the **Manage Bookmarks** dialog box. Refer to Bookmarks Management for more information.

- Load Bookmarks — Loads the bookmarks saved in the *bookmarks.do* file. You can choose whether to load bookmarks for the currently active window or all the bookmarks saved in the *bookmarks.do* file. Bookmarks are automatically loaded from the saved *bookmarks.do* file when you start a new simulation session.

> **Note:**
> You must reload bookmarks for a window if you close then reopen that window during the current session.

- Jump to Bookmark — Shows the available bookmarks in the currently active window followed by a drop down list of bookmarks for each window. You can set the maximum number of bookmarks listed in the Figure 6.

Bookmarks Management

You can open the Manage Bookmarks dialog box with the **Manage Bookmarks** toolbar button or by selecting **Bookmarks > Manage Bookmarks**.

The dialog box can be kept open during your simulation (Figure 5).

Figure 5. Manage Bookmarks Dialog Box
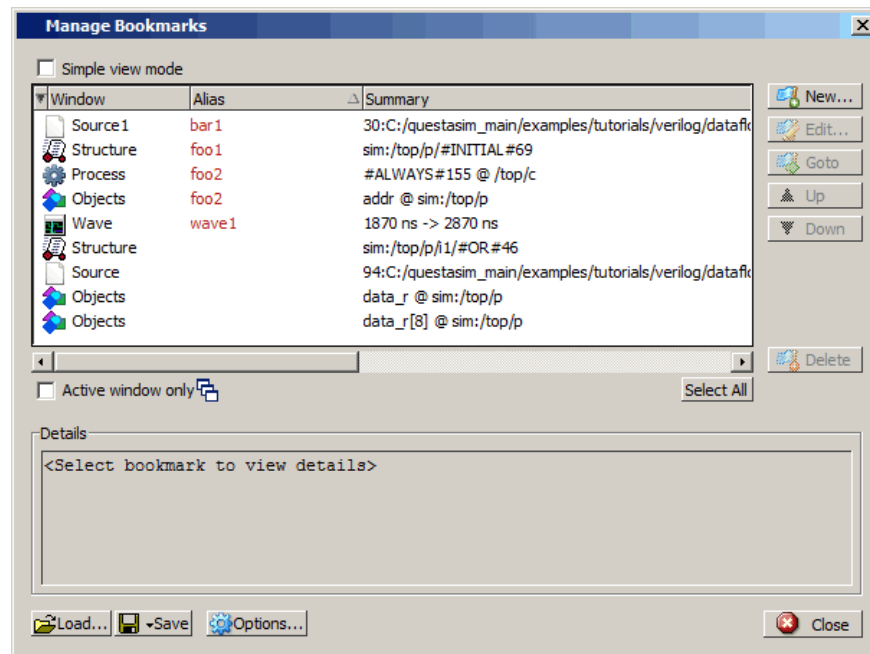
- **Simple view mode** changes the buttons from name and icon mode to icon only mode.

- Checking **Active window only** changes the display to show the bookmarks in the currently active window. Selecting a different window in the tool changes the display to the bookmarks set in that window.

- Selecting **New** opens the **New Bookmark** dialog box. The fields in the dialog automatically load the settings of the view in the currently active window. You can choose to name the bookmark with an alias to provide a more meaningful description. Aliases are displayed in the Alias column in the Manage Bookmarks dialog box.

- Selecting **Options** opens the **Bookmark Options** dialog box (Figure 6).

Figure 6. Bookmark Options Dialog Box
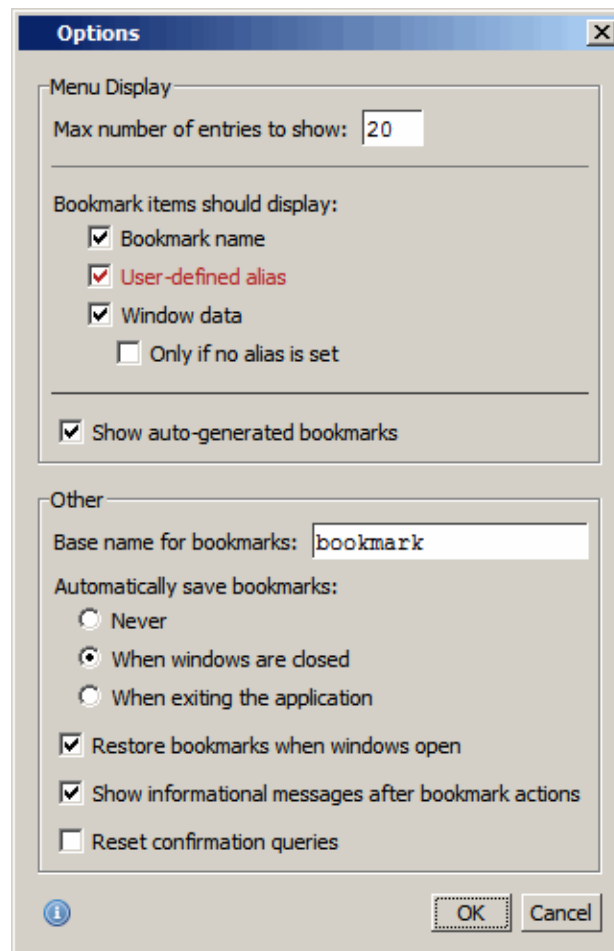
The **Menu Display** section allows you to:

◦ Set the number of bookmarks displayed in the Bookmarks menu or the Jump to Bookmark button menu.

◦ Select the types of information displayed for each bookmark.

The **Other** section allows you to:

◦ Specify a different base name for bookmarks.

◦ Choose whether you want to automatically save bookmarks and when they are saved.

◦ Automatically restore the bookmarks when windows are first loaded in the current session.

◦ **Show informational message after bookmark actions** sends bookmark actions to the transcript. For example:

```
# Bookmark(s) were restored for window "Source"
```

Saving and Reloading Formats and Content

You can use the write format restart command to create a single *.do* file that will recreate all debug windows and breakpoints (refer to Saving and Restoring Breakpoints in the User's Manual) when invoked with the do command in subsequent simulation runs. The syntax is:

write format restart <filename>

If the *modelsim.ini* variable is set to this *.do* filename, it will call the write format restart command upon exit. (Refer to ShutdownFile in the User's Manual for more information.)

Scribble Mode

On Linux®systems, you can capture an image of the desktop or a single window, make line drawings and take notes, then save the image for later review or to share with others. Scribble mode provides a quick way to make annotations directly on the GUI.

Figure 7. Scribble Mode
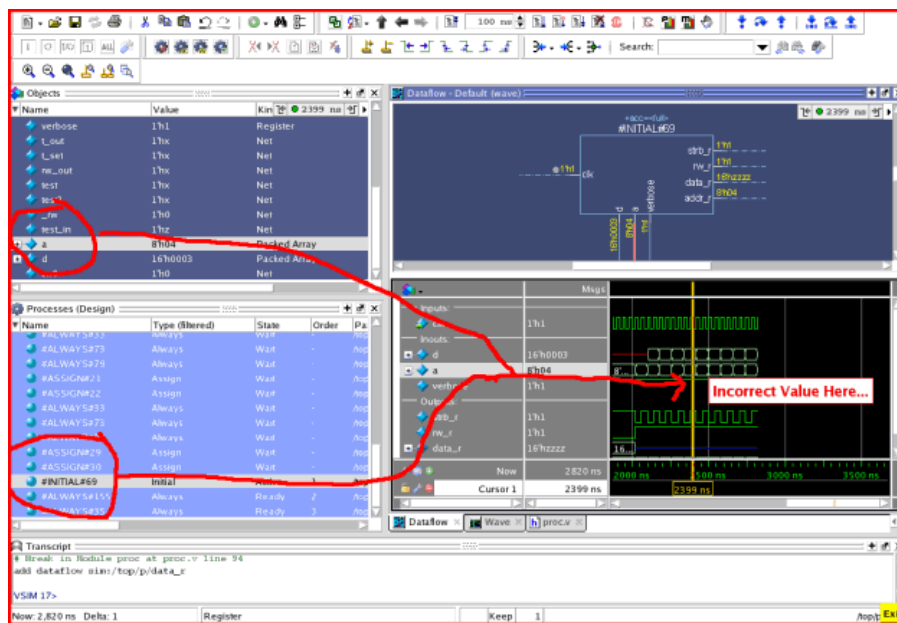
You can access this feature by selecting **Window > Enable Scribble Mode** from either the main window or a separate window. A new window with a red border is superimposed over a window or the desktop and access to all window and toolbar functions are suspended until you exit Scribble Mode. Use the left mouse button to draw and drag a text box into position. Use the right mouse button to open a menu with the following options:

Table 1. Scribble Mode Menu

Menu Item	Description
Add Text	Opens the Scribble Text dialog box for entering and formatting text
Attributes	Choose color and line thickness

Table 1. Scribble Mode Menu (continued)

Menu Item	Description
Clear Items	Choose which graphic elements to delete
Save Image	Saves the image as a bitmap to the current directory
Exit Scribble Mode	

Font Management

You may need to adjust font settings to accommodate the aspect ratios of wide screen and double screen displays or to handle launching Questa SIM from an X-session.

Refer to Setting GUI Preferences for more information.

Font Scaling

To change font scaling, select the Transcript window, then Transcript > Adjust Font Scaling. You will need a ruler to complete the instructions in the lower right corner of the dialog. When you have entered the pixel and inches information, click OK to close the dialog. Then, restart Questa SIM to see the change. This is a one time setting; you should not need to set it again unless you change display resolution or the hardware (monitor or video card). The font scaling applies to Windows and UNIX operating systems. On UNIX systems, the font scaling is stored based on the $DISPLAY environment variable.

Find and Filter Functions

Finding and/or filtering capabilities are available for most windows.

The Find mode toolbar is shown in Figure 8. The filtering function is denoted by a Contains field (Figure 9).

Figure 8. Find Mode

Figure 9. Filter Mode
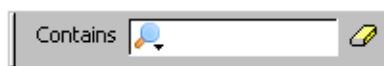
Windows that support both Find (Figure 8) and Filter modes (Figure 9) allow you to toggle between the two modes by doing any one of the following:

- Use the **Ctrl+M** hotkey.

- Click the "Find" or "Contains" words in the toolbar at the bottom of the window.

- Select the mode from the Find Options popup menu (see Find Options Popup Menu).

The last selected mode is remembered between sessions.

A "Find" toolbar will appear along the bottom edge of the active window when you do either of the following:

- Select **Edit > Find** in the menu bar.

- Click the **Find** icon in the Standard Toolbar.

All of the above actions are toggles - repeat the action and the Find toolbar will close.

The Find or Filter entry fields prefill as you type, based on the context of the current window selection. The find or filter action begins as you type.

There is a simple history mechanism that saves find or filter strings for later use. The keyboard shortcuts to use this feature are:

- **Ctrl+P** — retrieve previous search string

- **Ctrl+N** — retrieve next search string

Other hotkey actions include:

- **Esc** — closes the Find toolbar

- **Enter** — initiates a "Find Next" action

- **Ctrl+T** — search while typing (default is on)

The entry field turns red if no matches are found.

The graphic elements associated with the Find toolbar are shown in Table 2.

Note:
The Find Toolbar graphic elements are context driven. The actions available change for each window.

Table 2. Graphic Elements of Toolbar in Find Mode
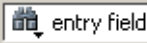
Graphic Element	Action
Find	opens the find toolbar in the active window
Close	closes the find toolbar
entry field Find entry field	allows entry of find parameters
Find Options	opens the Find Options popup menu at the bottom of the active window. The contents of the menu changes for each window.

Table 2. Graphic Elements of Toolbar in Find Mode (continued)

Graphic Element	Action
✎ Clear Entry Field	clears the entry field
▶ Execute Search	initiates the search
⬇ Toggle Search Direction	toggles search direction upward or downward through the active window
🔲 Find All Matches; Bookmark All Matches (for Source window only)	highlights every occurrence of the find item; for the Source window only, places a blue flag (bookmark) at every occurrence of the find item
Search For ▼ Search For	Click and hold the button to open a drop down menu with the following options: • Instance • Design Unit
☐ Aa Match Case	search must match the case of the text entered in the Find field
☐ {a} Exact (whole word)	searches for whole words that match those entered in the Find field
☐ Σ Regular Expression	Searches for a regular expression; Source window only.
☐ ⬆ Wrap Search	Searches from cursor to bottom of window then continues search from top of the window.

Structure Window Search Features

The Structure window Find bar supports hierarchical searching to limit the regions of a search. A forward slash (/) character is used to separate the search words. A double slash (//) is used to specify a recursive search from the double slash down the hierarchy (Figure 10). Refer to Finding Items in the Structure Window for more information.

Figure 10. Find Mode Popup Displaying Matches
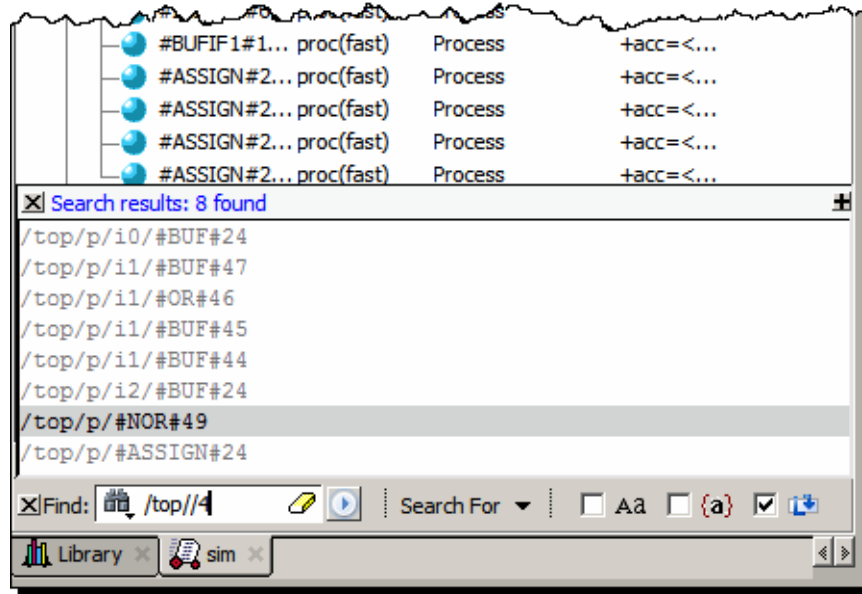
Filter Mode Options

By entering a string in the "Contains" text entry box you can filter the view of the selected window down to the specific information you are looking for.

Table 3. Graphic Elements of Toolbar in Filter Mode

Button	Name	Shortcuts	Description
Contains	Filter Regular Expression	None	A drop down menu that allows you to set the wildcard mode. A text entry box for your filter string.
(pencil icon)	Clear Filter	None	Clears the text entry box and removes the filter from the active window.

Wildcard Usage

There are three wildcard modes:

- glob-style — Allows you to use the following special wildcard characters:

  * — matches any sequence of characters in the string

  ? — matches any single character in the string

  [<chars>] — matches any character in the set <chars>

  \<x> — matches the single character <x>, which allows you to match on any special characters (*, ?, [,], and \)

  Refer to Finding Items in the Structure Window and the Tcl documentation for more information:

Help > Tcl Man Pages Tcl Commands > string > string match

- regular-expression — (Source window only) allows you to use wildcard characters based on Tcl regular expressions. For more information refer to the Tcl documentation:

  Help > Tcl Man Pages Tcl Commands > re_syntax

- exact — indicates that no characters have special meaning, thus disabling wildcard features.

The string entry field of the Contains toolbar item is case-insensitive, If you need to search for case-sensitive strings in the Source window select "regular-expression" and prepend the string with (?c).

Find Options Popup Menu

When you click the Find Options icon 🔍 in the Find entry field it will open a Find Options popup menu (Figure 11).

Figure 11. Find Options Popup Menu
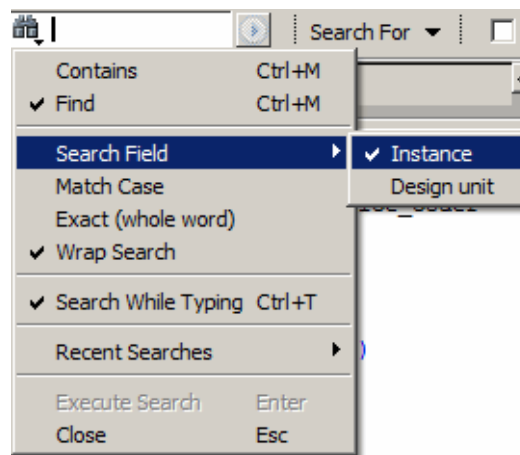
The Find Options menu displays the options available to you as well as hot keys for initiating the actions without the menu.

General Visual Elements

This section describes elements that are used by multiple windows.

Elements of the Main Window

Design Object Icons and Their Meanings

Window Time Display

Elements of the Main Window

Definitions of the GUI terminology used in this manual.

The Main window is the primary access point in the GUI. Figure 12 shows an example of the Main window during a simulation run.

Figure 12. Main Window of the GUI
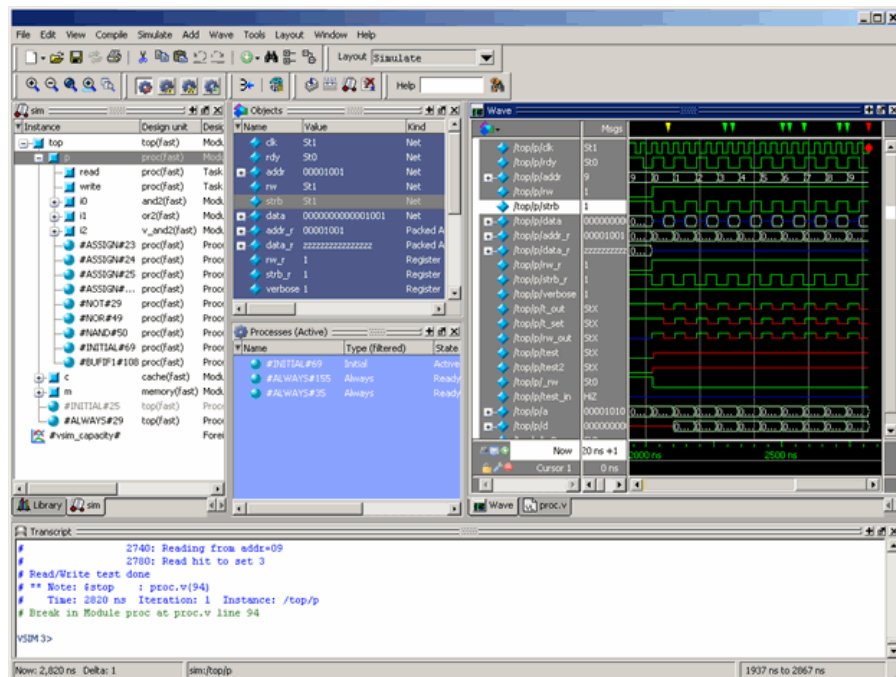
The Main window contains a menu bar, toolbar frame, windows, tab groups, and a status bar, which are described in the following sections.

Menu Bar

The menu bar provides access to many tasks available for your workflow. Figure 13 shows the selection in the menu bar that changes based on whichever window is currently active.

The menu items that are available and how certain menu items behave depend on which window is active. For example, if the Structure window is active and you choose Edit from the menu bar, the Clear command is disabled. However, if you click in the Transcript window and choose Edit, the Clear command is enabled. The active window is denoted by a blue title bar

Figure 13. Main Window — Menu Bar
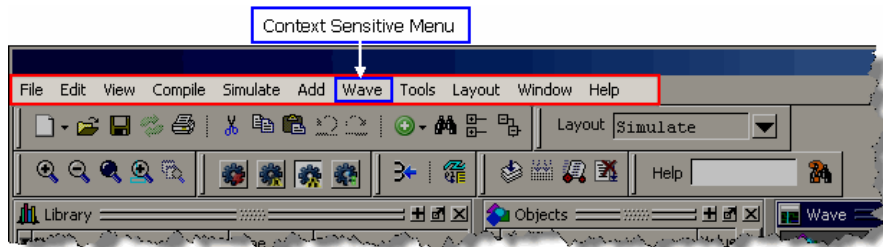
Toolbar Frame

The toolbar frame contains several toolbars that provide quick access to various commands and functions.

Figure 14. Main Window — Toolbar Frame
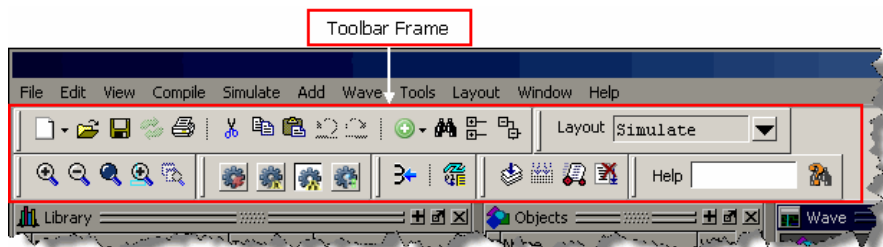
Toolbar

A toolbar is a collection of GUI elements in the toolbar frame and grouped by similarity of task. There are many toolbars available within the GUI, refer to the section Toolbars for more information about each toolbar. Figure 15 highlights the Compile toolbar in the toolbar frame.

Figure 15. Main Window — Toolbar
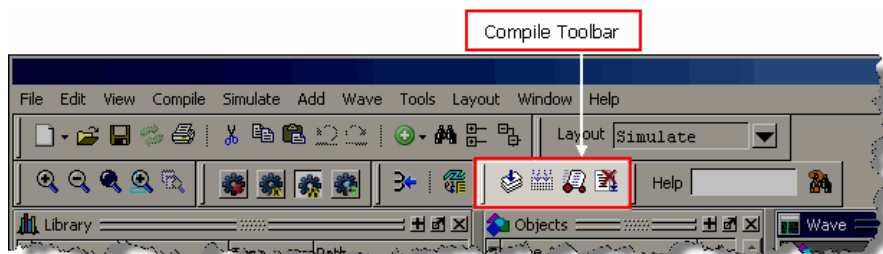
Window

Questa SIM can display over 40 different windows you can use with your workflow. This manual refers to all of these objects as windows, even though you can rearrange them such that they appear as a single window with tabs identifying each window.

Figure 16 shows an example of a layout with five windows visible; the Structure, Objects, Processes, Wave and Transcript windows.

Figure 16. GUI Windows
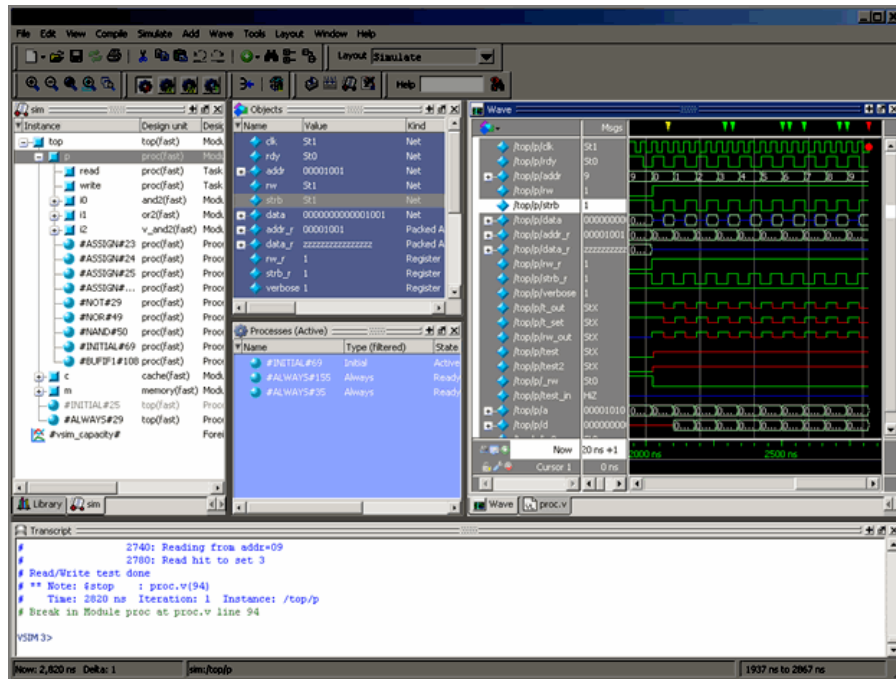
Tab Group

You can group any number of windows into a single space called a tab group, allowing you to show and hide windows by selecting their tabs. Figure 17 shows a tab group of the Library, Files, Capacity and Structure windows, with the Structure (sim) window visible.

Figure 17. GUI Tab Group
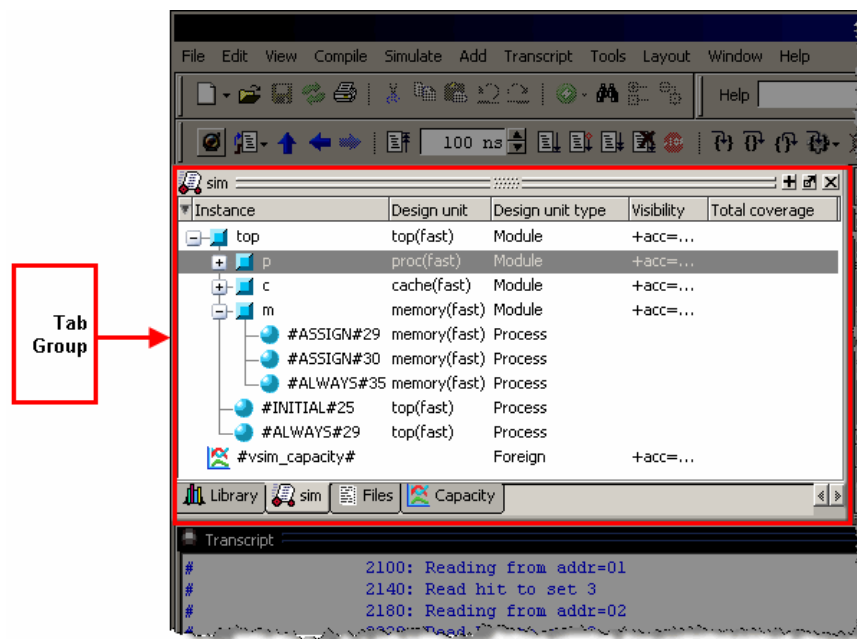
Pane

Some windows contain panes, which are separate areas of a window display containing distinct information within that window. One way to tell if a window has panes is whether you receive different popup menus (right-click menu) in different areas. Windows that have panes include the Wave, Source, and List windows. Figure 18 shows the Wave window with its the three panes.

Figure 18. Wave Window Panes
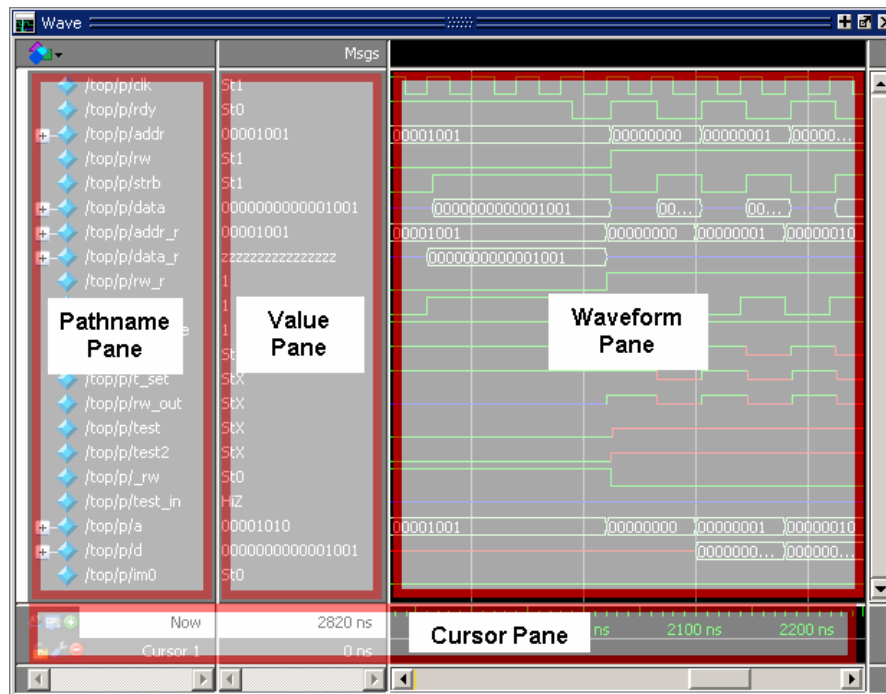
Main Window Status Bar

Fields at the bottom of the Main window provide the following information about the current simulation:

Figure 19. Main Window Status Bar

Table 4. Information Displayed in Status Bar

Field	Description
Project	name of the current project
Now	the current simulation time
Delta	the current simulation iteration number
Profile Samples	the number of profile samples collected during the current simulation

Table 4. Information Displayed in Status Bar (continued)

Field	Description
Memory	the total memory used during the current simulation
environment	name of the current context (object selected in the active Structure window)
line/column	line and column numbers of the cursor in the active Source window
Total Coverage	the aggregated coverage, as a percent, of the top level object in the design
coverage mode	recursive or non-recursive

Design Object Icons and Their Meanings

The color and shape of icons convey information about the language and type of a design object.

Table 5 shows the icon colors and the languages they indicate.

Table 5. Design Object Icons

Icon color	Design Language
light blue	Verilog or SystemVerilog
dark blue	VHDL
green	SystemC
magenta	PSL
orange	virtual object

The following table presents icon shapes and the design object types they indicate:

Table 6. Icon Shapes and Design Object Types

Icon shape	Example	Design Object Type
Square		any scope (VHDL block, Verilog named block, SC module, class, interface, task, function, and so forth.)
Square and red asterix		SystemVerilog object, OVM, and UVM test bench scope or object
Circle		process
Diamond		valued object (signals, nets, registers, SystemC channel, and so forth.)

Table 6. Icon Shapes and Design Object Types (continued)

Icon shape	Example	Design Object Type
Diamond and yellow pulse on red dot		an editable waveform created with the waveform editor
Diamond and red asterix		valued object (abstract)
Diamond and green arrow		indicates mode (In, Inout, Out) of an object port
Triangle		assertions (SV, PSL)
Triangle		caution sign on comparison object
Chevron		cover directives (SV, PSL)
Star		transaction; The color of the star for each transaction depends on the language of the region in which the transaction stream occurs: dark blue for VHDL light blue for Verilog and SystemVerilog green for SystemC

Window Time Display

There are two basic time designations used to control display of object values in many simulator windows.

- Now — The end-of-simulation time or the time at which the simulation has stopped.

- Current Time — The current time displayed in an open window. The time may be any time between 0 and the end of simulation, and is set in several ways:

  ◦ by moving a wave cursor

  ◦ by executing a causality trace that traces back through simulation time to find an event

  ◦ by interacting with the Current Time Label. Refer to Current Time Label for more information.

A number of windows are dynamically linked to update when the time setting in one is changed, including:

- Dataflow

- FSM

- Objects

- Schematic

- • Source

- • Watch

Current Time Label

The Current Time Label allows you to interact with and change the simulation time displayed in several windows. The Current Time Label displays the Now (end of simulation) or Current Time, and allows you to search the time line for a transition on a selected object in the window (Figure 20).

Figure 20. Current Time Label
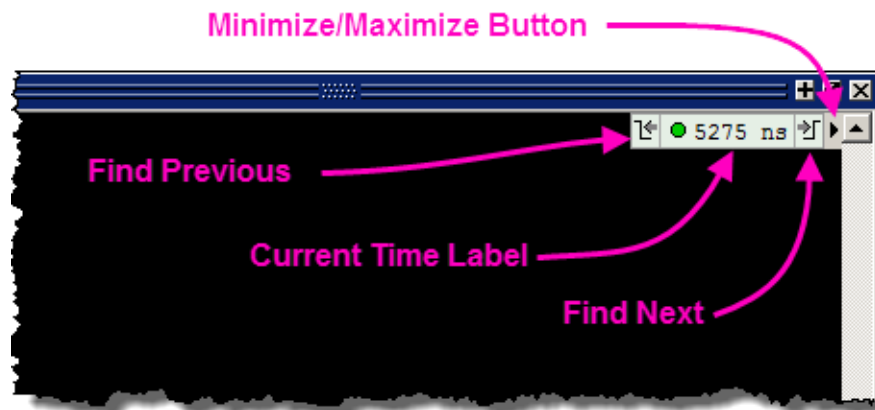
When you run a simulation and it comes to an end, the Current Time Label displays the Now time which is the end-of-simulation time. When you select a cursor in the Wave window the Current Time Label automatically changes to display the time of the current active cursor and updates the other windows to the same time.

The Current Time Label includes a minimize/maximize button that allows you to hide or display the label.

When you select an object in a window displaying the Current Time label, you can jump to the previous or next transition for that object, with respect to the current time, by clicking the Find Previous or Find Next button.

To change the display from showing the Current Time to showing the Now time (default), or vice versa, click the time display in the Current Time label to open a drop down menu (Figure 21) or select **View > Time Mode**.

Figure 21. Enter Current Time Value
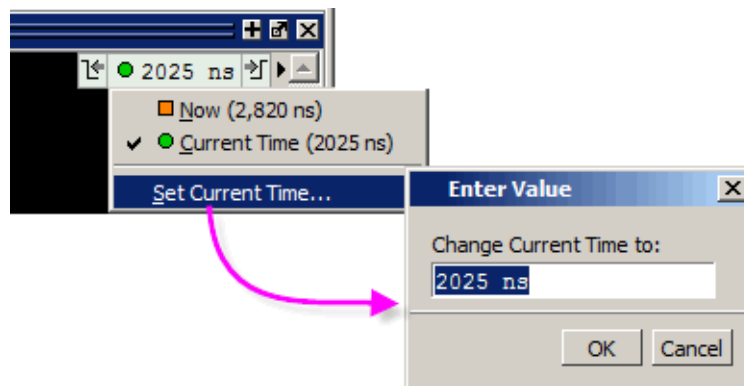
Chapter 2
Menus

The selections in the menu bar of the main window change dynamically, based on which window is currently active. As a result, some menu items change their names or become unavailable (dimmed). This section describes the menu items provided at the highest level provided for a given window.

Window-specific Menu

File Menu

Edit Menu

View Menu

Compile Menu

Simulate Menu

Add Menu

Tools Menu

Layout Menu

Bookmarks Menu

Window Menu

Help Menu

Window-specific Menu

Every window has a window-specific menu that appears in the menu bar between the Add and Tools menus.

The menu options available pertain only to that window and are described in the window-specific section of the chapter Window Reference.

File Menu

The File menu provides access to a number of simulation-wide options.

Table 7. File Menu — Item Description

Menu Item	Description
New	• Folder — create a new folder in the current directory
	• Source — create a new VHDL, Verilog or other source file
	• Project — create a new project
	• Library — create a new library and mapping
	• Debug Archive — archive debug data for post-simulation analysis. Refer to the archive write command for more information.
	• Regression — create a new Run Manager Database

Table 7. File Menu — Item Description (continued)

Menu Item	Description
Open	Open a file of any type.
Load	• DO File — load and run a script file (.*do* or .*tcl*) • Debug Archive — load archived debug data for post-simulation analysis. Refer to the archive load command for more information. • UCDB Testanalysis — load coverage test association data for the currently loaded, merged UCDB (unless the merged UCDB contains less than 100 tests: in that case, the data is loaded automatically). Refer to the coverage loadtestassoc command for more information.
Close	Close an opened file
Import	• Library — import FPGA libraries • EVCD — import an extended VCD file previously created with the ModelSim Waveform Editor. This item is enabled only when a Wave window is active • Memory Data — initialize a memory by reloading a previously saved memory file. • Testplan — import a verification management testplan. Only applies to the Verification Management Browser window. • Column Layout — apply a previously saved column layout to the active window • Hierarchy configuration — import a results analysis hierarchy configuration.
Export	• Waveform — export a created waveform • Tabular list — writes List window data to a file in tabular format • Event list — writes List window data to a file as a series of transitions that occurred during simulation • TSSI list — writes List window data to a file in TSSI format • Image — saves an image of the active window • Memory Data — saves data from the selected memory in the Memory List window or an active Memory Data window to a text file • Column Layout — saves a column layout from the active window • Hierarchy configuration — exports a results analysis hierarchy configuration. • HTML — opens up a dialog where you can specify the name of an HMTL file and the directory where it is saved • VCD — exports selected signals in Wave window to VCD file.
Save	These menu items change based on the active window.

Table 7. File Menu — Item Description (continued)

Menu Item	Description
Save as	
Report	Produce a textual report based on the active window
Change Directory	Opens a browser for you to change your current directory. Not available during a simulation, or if you have a dataset open.
Use Source	Specifies an alternative file to use for the current source file. This mapping only exists for the current simulation. This option is only available from the Structure window.
Source Directory	Control which directories are searched for source files.
Datasets	Manage datasets for the current session.
Environment	Set up how different windows should be updated, by dataset, process, and/or context. This is only available when the Structure, Locals, Processes, and Objects windows are active.
Page Setup Print Print Postscript	Manage the printing of information from the selected window.
Recent Directories	Display a list of recently opened working directories. You may change the number of directories displayed by changing the value of the PrefMain(recentDirectoryLimit) preference setting. Select **Tools > Edit Preferences**, then the **By Name** tab. The default value is 8.
Recent Projects	Display a list of recently opened projects. You may change the number of projects displayed by changing the value of the PrefMain(recentProjectLimit) preference setting. Select **Tools > Edit Preferences**, then the **By Name** tab. The default value is 8.
Close Window	Close the active window
Quit	Quit the application

Edit Menu

The Edit menu provides access to a number of options for manipulating information in the GUI.

Table 8. Edit Menu — Item Description

Menu Item	Description
Undo Redo	Alter your previous edit in a Source window.
Cut	Use or remove selected text.

Table 8. Edit Menu — Item Description (continued)

Menu Item	Description
Copy Paste	
Delete	Remove an object from the Wave and List windows
Clear	Clear the Transcript window
Select All Unselect All	Change the selection of items in a window
Expand	Expand or collapse hierarchy information
Goto	Goto a specific line number in the Source window
Find	Open the find toolbar. Refer to the section Find and Filter Functions for more information
Replace	Find and replace text in a Source window.
Signal Search	Search the Wave or List windows for a specified value, or the next transition for the selected object
Find in Files	search for text in saved files
Previous Coverage Miss Next Coverage Miss	Find the previous or next line with missed coverage in the active Source window

View Menu

The View menu provides to options for manipulating the GUI windows.

Table 9. View Menu — Item Description

Menu Item	Description
Window Name	Displays the selected window
New Window	Open additional instances of the Wave, List, or Dataflow windows
Sort	Change the sort order of the Wave window
Filter	Filters information from the Objects and Structure windows.
Justify	Change the alignment of data in the selected window.
Properties	Displays file property information from the Files or Source windows.

Compile Menu

The Compile menu provides access to options related to the compile step of your simulation run.

Table 10. Compile Menu — Item Description

Menu Item	Description
Compile	Compile source files
Compile Options	Set various compile options.
SystemC Link	Collect the object files created in the different design libraries, and uses them to build a shared library (.so) in the current work library
Compile All	Compile all files in the open project. Disabled if you do not have a project open
Compile Selected	Compile the files selected in the project tab. Disabled if you do not have a project open
Compile Order	Set the compile order of the files in the open project. Disabled if you do not have a project open
Compile Report	report on the compilation history of the selected file(s) in the project. Disabled if you do not have a project open
Compile Summary	report on the compilation history of all files in the project. Disabled if you do not have a project open

Simulate Menu

The Simulate menu provides access to options related to the simulation step of your simulation run

Table 11. Simulate Menu — Item Description

Menu item	Description
Design Optimization	Open the Design Optimization dialog to configure simulation optimizations
Start Simulation	Load the selected design unit
Runtime Options	Set various simulation runtime options
Run	• Run <default> — run simulation for one default run length; change the run length with **Simulate > Runtime Options**, or use the Run Length text box on the toolbar • Run -All — run simulation until you stop it • Continue — continue the simulation • Run -Next — run to the next event time

Table 11. Simulate Menu — Item Description (continued)

Menu item	Description
	• Step — single-step the simulator • Step -Over — execute without single-stepping through a subprogram call • Restart — reload the design elements and reset the simulation time to zero; only design elements that have changed are reloaded; you specify whether to maintain various objects (logged signals, breakpoints, and so on)
Break	Stop the current simulation run
End Simulation	Quit the current simulation run

Add Menu

The Add menu provides access to options for populating windows with data from other windows.

Table 12. Add Menu — Item Description

Menu Item	Description
To Wave	Add information to the Wave window
To List	Add information to the List window
To Log	Add information to the Log file
To Schematic	Add information to the Schematic window
To Dataflow	Add information to the Dataflow window
Window Pane	Add an additional pane to the Wave window. You can remove this pane by selecting **Wave > Delete Window Pane**.

Tools Menu

The Tools menu provides access to various tools available within the GUI.

Table 13. Tools Menu — Item Description

Menu Item	Description
Waveform Compare	Access tasks for waveform comparison. Refer to Waveform Compare in the User's Manual.
Code Coverage	Access tasks for code coverage. Refer to Code Coverage Analysis Window in the GUI Reference Manual.

Table 13. Tools Menu — Item Description (continued)

Menu Item	Description
Functional Coverage	Access menus for configuring cover directives and filtering functional coverage items. Refer to Functional Coverage Control Options and Filtering Functional Coverage Data in the User's Manual.
Toggle Coverage	Add toggle coverage tracking. Refer to Toggle Coverage in the User's Manual.
Coverage Save	Save the coverage metrics to a UCDB file.
Coverage Report	Save the coverage metrics to report file.
Coverage Configuration	Access menus for configuring coverage related items, such as goal, weight and colorization. Refer to Viewing Test Data in the GUI in the User's Manual.
Profile	Access tasks for the memory and performance profilers. Refer to Classic Performance Profiler in the User's Manual.
Breakpoints	Manage breakpoints
Trace	Perform signal trace actions.
Dataset Snapshot	Enable periodic saving of simulation data to a *.wlf* file.
C Debug	Access tasks for the C Debug interface. Refer to C Debug in the User's Manual.
JobSpy	Access tasks for the JobSpy utility. Refer to Monitoring Simulations with JobSpy in the User's Manual.
Tcl	Execute or debug a Tcl macro.
Wildcard Filter	Refer to Using the WildcardFilter Preference Variable in the Command Reference Manual.
Edit Preferences	Set GUI preference variables. Refer to Setting GUI Preferences for more information.

Layout Menu

The Layout menu provides access to options for manipulating the configuration of the GUI.

Table 14. Layout Menu — Item Description

Menu Item	Description
Reset	Reset the GUI to the default appearance for the selected layout.
Save Layout As	Save your reorganized view to a custom layout. Refer to the section Simulator GUI Layout Customization for more information.

Table 14. Layout Menu — Item Description (continued)

Menu Item	Description
Configure	Configure the layout-specific behavior of the GUI. Refer to the section Configure Window Layouts Dialog Box for more information.
Delete	Delete a customized layout. You can not delete any of the five standard layouts.
Layout Name	Select a standard or customized layout.

Bookmarks Menu

The Bookmarks menu provides access to options related to bookmarking the Wave window.

Table 15. Bookmarks Menu — Item Description

Menu Item	Description
Add	Clicking this button bookmarks the current view of the Wave window.
Add Custom	Opens the New Bookmark dialog box.
Manage	Opens the Manage Bookmarks dialog box.
Delete All	• Active Window Only • All Windows.
Reload from File	• Active Window Only • All Windows.

Window Menu

The Window menu provides access to options for organizing and controlling features of GUI windows.

Table 16. Window Menu — Item Description

Menu Item	Description
Cascade Tile Horizontally Tile Vertically	Arrange all undocked windows. These options do not impact any docked windows.
Icon Children Icon All Deicon All	Minimize (Icon) or Maximize (Deicon) undocked windows. These options do not impact any docked windows.

Table 16. Window Menu — Item Description (continued)

Menu Item	Description
Show Toolbar	Toggle the appearance of the Toolbar frame of the Main window
Show Window Headers	Toggle the appearance of the window headers. Note that you will be unable to rearrange windows if you do not show the window headers.
FocusFollowsMouse	Mouse pointer makes window active when pointer hovers in the window briefly. Refer to Selecting the Active Window for more information.
Keyboard Shortcuts	Opens the Keyboard Shortcuts dialog box. Refer to User-Defined Keyboard Shortcuts for more information.
Add Toolbar Button	Add a button to the toolbar frame.
Toolbars	Toggle the appearance of available toolbars. Similar behavior to right-clicking in the toolbar frame.
window name	Make the selected window active.
Windows	Display the Windows dialog box, which allows you to activate, close or undock the selected window(s).

Help Menu

The Help menu provides access to various types of documentation.

Table 17. Help Menu — Item Description

Menu Item	Description
About	Display Questa SIM application information.
Release Notes	Display the current Release Notes in the Questa SIM Notepad editor. You can find past release notes in the *<install_dir>/docs/rlsnotes/* directory.
Welcome Window	Display the Important Information splash screen. By default this window is displayed on startup. You can disable the automatic display by toggling the **Don't show this dialog again** radio button.
Command Completion	Toggles the command completion dropdown box in the transcript window. When you start typing a command at the Transcript prompt, a dropdown box appears which lists the available commands matching what has been typed so far. You may use the Up and Down arrow keys or the mouse to select the desired command. When a unique command has been entered, the command usage is presented in the drop down box.

Table 17. Help Menu — Item Description (continued)

Menu Item	Description
Register File Types	Associate files types (such as *.v*, *.sv*, *.vhd*, *.do*) with the product. These associations are typically made upon install, but this option allows you to update your system in case changes have been made since installation.
Questa SIM Documentation - InfoHub	Open the HTML-based portal for all PDF and HTML documentation.
Questa SIM Documentation - PDF Bookcase	Open the PDF-based portal for the most commonly used PDF documents.
Tcl Help	Open the Tcl command reference (man pages) in Windows help format.
Tcl Syntax	Open the Tcl syntax documentation in your web browser.
Tcl Man pages	Open the Tcl/Tk manual in your web browser.
Technotes	Open a technical note in the Questa SIM Notepad editor.

Chapter 3
Toolbars

The Main window contains a toolbar frame that displays context-specific toolbars. The following sections describe the toolbars and their associated buttons.

You can determine the name of a toolbar in the GUI by right-clicking on the toolbar and looking for the bolded name in the pop-up menu.

ATV Toolbar

The ATV toolbar allows you to control aspects of the Assertion Thread Viewer.

Figure 22. ATV Toolbar

Table 18. ATV Toolbar Buttons

Button	Name	Shortcuts	Description
	View Grid	**Menu: ATV > View Grid**	creates horizontal grid lines from the assertion expression through the Thread Viewer pane
	Ascending Expressions	**Menu: ATV > Ascending Expressions**	changes the display of expressions in the Expressions pane from descending order (default) to ascending order
	Annotate Local Vars	**Menu: ATV > Annotate Local Vars**	displays annotation of local variables in the Thread Viewer pane
	Show Local Vars	**Menu: ATV > Show Local Vars**	displays the Local Variables pane at the bottom of the viewer
	Show Design Objects	**Menu: ATV > Show Design Objects**	displays the Design Objects pane at the bottom of the ATV viewer

Analysis Toolbar

The Analysis (coverage) toolbar allows you to control aspects of the Code Coverage Analysis window.

Figure 23. Analysis Toolbar
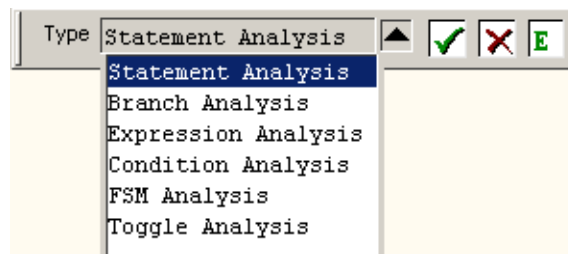
Table 19. Analysis Toolbar Buttons
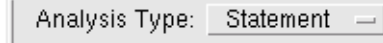
Button	Name	Shortcuts	Description
	Type	**Command**: view canalysis	A dropdown box that allows you to specify the type of code coverage to view in the Code Coverage Analysis Window.

Table 19. Analysis Toolbar Buttons (continued)

Button	Name	Shortcuts	Description
✓	Covered	**Menu**: N/A	All covered (hit) items are displayed.
✗	Missed	**Menu**: N/A	All missed items (not executed) are displayed.
E	Excluded	**Menu**: N/A	Restores the precision to the default value (2).

Bookmarks Toolbar

The Bookmark toolbar allows you to manage your bookmarks of the Wave window

Figure 24. Bookmarks Toolbar

Table 20. Bookmarks Toolbar Buttons

Button	Name	Shortcuts	Description
	Add Bookmark	Command (Wave window only): bookmark add wave Menu Wave window only: **Add > To Wave > Bookmark**	Clicking this button bookmarks the current view of the active window. Click and hold the button to open a drop down menu with the following options: • Add Current View • Add Custom ... • Set Default Action
	Delete All Bookmarks	Command (Wave window only): bookmark delete wave -all	Removes all bookmarks, after prompting for your confirmation. Click and hold the button to open a drop down menu with the following options: • Active Window • All Windows
	Manage Bookmarks	None	Displays the Manage Bookmarks dialog box.
	Reload from File	None	Reloads bookmarks from the *bookmarks.do* file. • Set Default Action

Table 20. Bookmarks Toolbar Buttons (continued)

Button	Name	Shortcuts	Description
	Jump to Bookmark	Command (Wave window only): bookmark goto wave <name>	Displays bookmarks grouped by window. Select the bookmark you want to display.

Column Layout Toolbar

The Column Layout toolbar allows you to specify the column layout for the active window.

Figure 25. Column Layout Toolbar
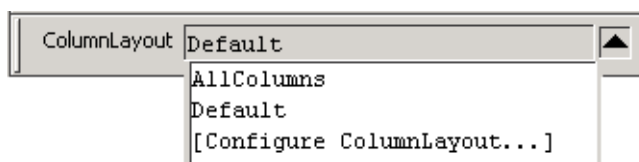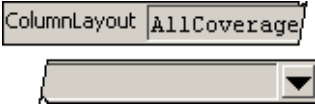
Table 21. Change Column Toolbar Buttons

Button	Name	Shortcuts	Description
	Column Layout	Menu:**View > Verification Management Browser > Configure Column Layout**	A dropdown box that allows you to specify the column layout for the active window.

The Column Layout dropdown menu allows you to select pre-defined column layouts for the active window. For example:

- AllColumns — displays all available columns.

- Default — displays only columns that are displayed by default.

- CodeCoverageRanked — displays all columns related to ranked code coverage results.

- FunctionalCoverageRanked - displays all columns related to ranked functional coverage results.

- AllCoverage — displays all columns related to coverage statistics.

- TestData — displays all columns containing data about the test, including information about how and when the coverage data was generated.

- Testplan — displays all columns containing data about the testplan, including Testplan Section / Coverage Link, Type, Goal, % of Goal, Weight.

- AllCoverageRanked — displays all columns related to ranking results.

- FunctionalCoverage — displays all columns related to functional coverage statistics.

- CodeCoverage — displays all columns related to code coverage statistics.

- [Configure ColumnLayout…] — opens the Configure Column Layout dialog, which allows you to create, edit, remove, copy, or rename a column layout. See Column Layout Configuration.

Compile Toolbar

The Compile toolbar provides access to compile and simulation actions.

Figure 26. Compile Toolbar

Table 22. Compile Toolbar Buttons

Button	Name	Shortcuts	Description
	Compile	Command: vcom or vlog Menu: **Compile > Compile**	Opens the Compile Source Files dialog box.
	Compile All	Command: vcom or vlog Menu: **Compile > Compile All**	Compiles all files in the open project.
	Simulate	Command: vsim Menu: **Simulate > Start Simulation**	Opens the Start Simulation dialog box.
	Break	Menu: **Simulate > Break** Hotkey: Break	Stop a compilation, elaboration, or the current simulation run.

Coverage Toolbar

The Coverage toolbar provides tools for filtering code coverage data in the Structure and Instance Coverage windows.

Figure 27. Coverage Toolbar
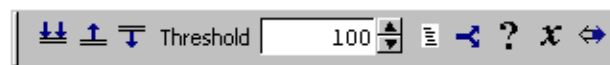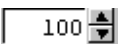
Table 23. Coverage Toolbar Buttons

Button	Name	Shortcuts	Description
	Enable Filtering	None	Enables display filtering of coverage statistics in the Structure and Instance Coverage windows.

Table 23. Coverage Toolbar Buttons (continued)

Button	Name	Shortcuts	Description
	Threshold Above	None	Displays all coverage statistics above the Filter Threshold for selected columns.
	Threshold Below	None	Displays all coverage statistics below the Filter Threshold for selected columns
100	Filter Threshold	None	Specifies the display coverage percentage for the selected coverage columns
	Statement	None	Applies the display filter to all Statement coverage columns in the Structure and Instance Coverage windows.
	Branch	None	Applies the display filter to all Branch coverage columns in the Structure and Instance Coverage windows.
	Condition	None	Applies the display filter to all Condition coverage columns in the Structure and Instance Coverage windows.
	Expression	None	Applies the display filter to all Expression coverage columns in the Structure and Instance Coverage windows.
	Toggle	None	Applies the display filter to all Toggle coverage columns in the Structure and Instance Coverage windows.

Dataflow Toolbar

The Dataflow toolbar provides access to various tools to use in the Dataflow window.

Figure 28. Dataflow Toolbar

Table 24. Dataflow Toolbar Buttons

Button	Name	Shortcuts	Description
	Trace Input Net to Event	Menu: **Tools > Trace > Trace next event**	Move the next event cursor to the next input event driving the selected output.
	Trace Set	Menu: **Tools > Trace > Trace event set**	Jump to the source of the selected input event.
	Trace Reset	Menu: **Tools > Trace > Trace event reset**	Return the next event cursor to the selected output.

Table 24. Dataflow Toolbar Buttons (continued)

Button	Name	Shortcuts	Description
	Trace Net to Driver of X	Menu: **Tools > Trace > TraceX**	Step back to the last driver of an unknown value.
	Expand Net to all Drivers	None	Display driver(s) of the selected signal, net, or register.
	Expand Net to all Drivers and Readers	None	Display driver(s) and reader(s) of the selected signal, net, or register.
	Expand Net to all Readers	None	Display reader(s) of the selected signal, net, or register.
	Show Wave	Menu: **Dataflow > Show Wave**	Display the embedded wave viewer pane.

FSM Toolbar

The FSM toolbar provides access to tools that control the information displayed in the FSM Viewer window.

Figure 29. FSM Toolbar
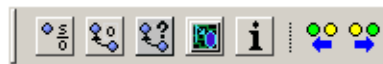
Table 25. FSM Toolbar Buttons

Button	Name	Shortcuts	Description
	Show State Counts	Menu: **FSM View > Show State Counts**	(only available when simulating with -coverage) Displays the coverage count over each state.
	Show Transition Counts	Menu: **FSM View > Show Transition Counts**	(only available when simulating with -coverage) Displays the coverage count for each transition.
	Show Transition Conditions	Menu: **FSM View > Show Transition Conditions**	Displays the conditions of each transition.
	Track Wave Cursor	Menu: **FSM View > Track Wave Cursor**	The FSM Viewer tracks your current cursor location.
	Enable Info Mode Popups	Menu: **FSM View > Enable Info Mode Popups**	Displays information when you mouse over each state or transition

Table 25. FSM Toolbar Buttons (continued)

Button	Name	Shortcuts	Description
	Previous State	None	Steps to the previous state in the FSM Viewer window.
	Next State	None	Steps to the next state in the FSM Viewer window.

Help Toolbar

The Help toolbar provides a way for you to search the HTML documentation for a specified string. The HTML documentation will be displayed in a web browser.

Figure 30. Help Toolbar

Table 26. Help Toolbar Buttons

Button	Name	Shortcuts	Description
	Search Documentation	None	A text entry box for your search string.
	Search Documentation	Hotkey: Enter	Activates the search for the term you entered into the text entry box.

Layout Toolbar

The Layout toolbar allows you to select a predefined or user-defined layout of the graphical user interface.

Refer to the section Simulator GUI Layout Customization for more information.

Figure 31. Layout Toolbar
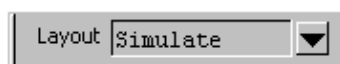
Table 27. Layout Toolbar Buttons
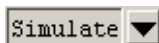
Button	Name	Shortcuts	Description
	Change Layout	Menu: **Layout > <layout_name>**	A dropdown box that allows you to select a GUI layout. • NoDesign • Simulate

Table 27. Layout Toolbar Buttons (continued)

Button	Name	Shortcuts	Description
			• Coverage • VMgmt
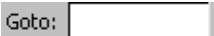
Memory Toolbar

The Memory toolbar provides access to common functions.

Figure 32. Memory Toolbar

Table 28. Memory Toolbar Buttons

Button	Name	Shortcuts	Description
	Split Screen	Menu: **Memory > Split Screen**	Splits the memory window.
	Goto Address		Highlights the first element of the specified address.

Mode Toolbar

The Mode toolbar provides access to tools for controlling the mode of mouse navigation.

Figure 33. Mode Toolbar

Table 29. Mode Toolbar Buttons

Button	Name	Shortcuts	Description
	Select Mode	Menu: **Dataflow > Mouse Mode > Select Mode** or **Schematic > Mouse Mode > Select Mode**	Set the left mouse button to select mode and middle mouse button to zoom mode.
	Zoom Mode	Menu: **Dataflow > Mouse Mode > Zoom Mode** or **Schematic > Mouse Mode > Zoom Mode**	Set left mouse button to zoom mode and middle mouse button to pan mode.

Table 29. Mode Toolbar Buttons (continued)

Button	Name	Shortcuts	Description
✛	Pan Mode	Menu: **Dataflow > Mouse Mode > Pan Mode** or **Schematic > Mouse Mode > Pan Mode**	Set left mouse button to pan mode and middle mouse button to zoom mode.
↥↥	Two Cursor Mode	Menu: **Wave > Mouse Mode > Two Cursor**	Sets two cursors in Wave window. First cursor moves with LMB, second cursor with MMB.
⬚	Edit Mode	Menu: **Wave or Dataflow > Mouse Mode > Edit Mode**	Set mouse to Edit Mode, where you drag the left mouse button to select a range and drag the middle mouse button to zoom.
⬚	Stop Drawing	None	Halt any drawing currently happening in the window.

Objectfilter Toolbar

The Objectfilter toolbar provides filtering of design objects appearing in the Objects window.

Figure 34. Objectfilter Toolbar
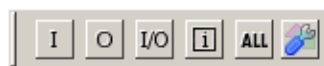
Table 30. Objectfilter Toolbar Buttons

Button	Name	Shortcuts	Description
I	View Inputs Only	None	Changes the view of the Objects Window to show inputs.
O	View Outputs Only	None	Changes the view of the Objects Window to show outputs.
I/O	View Inouts Only	None	Changes the view of the Objects Window to show inouts.
i	Vies Internal Signals	None	Changes the view of the Objects Window to show Internal Signals.
ALL	Reset All Filters	None	Clears the filtering of Objects Window entries and displays all objects.
🔧	Change Filter	None	Opens the Filter Objects dialog box.

Precision Toolbar

The Precision toolbar allows you to control the precision of the data in the Verification Management windows (Browser, Tracker, Results Analysis).

Figure 35. Precision Toolbar
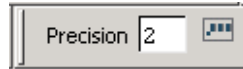
Table 31. Precision Toolbar Buttons

Button	Name	Shortcuts	Description
Precision 2	Set Precision for VMgmt	Menu: Verification Browser > Set Precision	A text entry box that allows you to control the precision of the data in the Verification Browser window.
	Restore Default Precision	None	Restores the precision to the default value (2).

Process Toolbar

The Process toolbar contains three toggle buttons (only one can be active at any time) that controls the view of the Process window.

Figure 36. Process Toolbar

Table 32. Process Toolbar Buttons

Button	Name	Shortcuts	Description
	View Active Processes	Menu: **Process > Active**	Changes the view of the Processes Window to only show active processes.
	View Processes in Region	Menu: **Process > In Region**	Changes the view of the Processes window to only show processes in the active region.
	View Processes for the Design	Menu: **Process > Design**	Changes the view of the Processes window to show processes in the design.
	View Process hierarchy	Menu: **Process > Hierarchy**	Changes the view of the Processes window to show process hierarchy.

Profile Toolbar

The Profile toolbar provides access to tools related to the profiling windows (Ranked, Calltree, Design Unit, and Structural).

Figure 37. Profile Toolbar

Table 33. Profile Toolbar Buttons

Button	Name	Shortcuts	Description
	Collapse Sections	Menu: **Tools > Profile > Collapse Sections**	Toggle the reporting for collapsed processes and functions.
	Profile Cutoff	None	Display performance and memory profile data equal to or greater than set percentage.
	Refresh Profile Data	None	Refresh profile performance and memory data after changing profile cutoff.
	Save Profile Results	Menu: **Tools > Profile > Profile Report**	Save profile data to output file (prompts for file name).

Schematic Toolbar

The Schematic toolbar provides access to tools for manipulating highlights and signals in the Dataflow and Schematic windows.

Figure 38. Schematic Toolbar

Table 34. Schematic Toolbar Buttons

Button	Name	Shortcuts	Description
	Trace Input Net to Event	Menu: **Tools > Trace > Trace next event**	Move the next event cursor to the next input event driving the selected output.
	Trace Set	Menu: **Tools > Trace > Trace event set**	Jump to the source of the selected input event.
	Trace Reset	Menu: **Tools > Trace > Trace event reset**	Return the next event cursor to the selected output.

Table 34. Schematic Toolbar Buttons (continued)

Button	Name	Shortcuts	Description
	Trace Net to Driver of X	Menu: **Tools > Trace > TraceX**	Step back to the last driver of an unknown value.
	Expand Net to all Drivers	None	Display driver(s) of the selected signal, net, or register.
	Expand Net to all Drivers and Readers	None	Display driver(s) and reader(s) of the selected signal, net, or register.
	Expand Net to all Readers	None	Display reader(s) of the selected signal, net, or register.
	Remove All Highlights	Menu: **Dataflow > Remove Highlight** or **Schematic > Edit > Remove Highlight**	Clear the highlighting identifying the path you have traversed through the design. Click and hold the button to open a drop down menu with the following options: • Remove All Highlights • Remove Selected Highlights • Set Default Action
	Delete Content	Menu: **Dataflow > Delete** or **Schematic > Edit > Delete** **Schematic > Edit > Delete All**	Delete the selected signal. Click and hold the button to open a drop down menu with the following options: • Delete Selected • Delete All • Set Default Action
	Regenerate	Menu: **Dataflow > Regenerate** or **Schematic > Edit > Regenerate**	Redraws the current schematic view to better take advantage of the available space. For example, after adding or removing elements.
	Enable 1-Click Mode		Enables single-click sprout expansion. Default is double-click to sprout.
	Show Wave	Menu: **Schematic > Show Wave**	Display the embedded Wave Viewer pane.

Simulate Toolbar

The Simulate toolbar provides various tools for controlling your active simulation.

Figure 39. Simulate Toolbar

Table 35. Simulate Toolbar Buttons
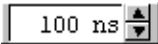
Button	Name	Shortcuts	Description
	Source Hyperlinking	None	Toggles display of hyperlinks in design source files.
	Show Cause	Command: find drivers -active	Traces a selected wave signal from the current time back to the first sequential element. See #unique_60_Connect_42_id8f7d4c37-c85a-4c3f-8dbb-c005e1bb1757 for more information. • Set Default Action
	Environment Up	Command: env … Menu: **File > Environment**	Changes your environment up one level of hierarchy.
	Environment Back	Command: env -back Menu: **File > Environment**	Change your environment to its previous location.
	Environment Forward	Command: env -forward Menu: **File > Environment**	Change your environment forward to a previously selected environment.
	Restart	Command: restart Menu: **Simulate > Run > Restart**	Reload the design elements and reset the simulation time to zero, with the option of maintaining various settings and objects.
	Run Length	Command: run Menu: **Simulate > Runtime Options**	Specify the run length for the current simulation.
	Run	Command: run Menu: **Simulate > Run > Run** *<default_run_length>*	Run the current simulation for the specified run length.
	Continue Run	Command: run -continue Menu: **Simulate > Run > Continue**	Continue the current simulation run until the end of the specified run length or until it hits a breakpoint or specified break event.
	Run All	Command: run -all Menu: **Simulate > Run > Run -All**	Run the current simulation forever, or until it hits a breakpoint or specified break event.
	Break	Menu: Simulate > Break	Immediate stop of a compilation, elaboration, or simulation run.

Table 35. Simulate Toolbar Buttons (continued)

Button	Name	Shortcuts	Description
		Hotkey: Break	Similar to hitting a breakpoint if the simulator is in the middle of a process.
	Stop -sync	None	Stop simulation the next time time/delta is advanced.
	C Interrupt	Command: cdbg interrupt Menu: **Tools > C Debug > C Interrupt**	Reactivate the C debugger when stopped in HDL code.
	Performance Profiling	Menu: **Tools > Profile > Performance**	Enable collection of statistical performance data.
	Memory Profiling	Menu: **Tools > Profile > Memory**	Enable collection of memory usage data.
	Edit Breakpoints	Menu: **Tools > Breakpoint**	Enable breakpoint editing, loading, and saving.

Show Cause Button

Clicking this button initiates a trace on a signal event back to the first sequential driving process. When you click-and-hold the button you can access additional options via a dropdown menu, as shown in Figure 40.

Figure 40. Show Cause Dropdown Menu
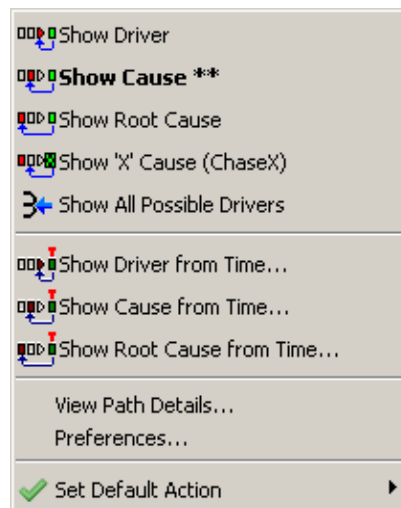
- **Show Cause**— Refer to Trace to the First Sequential Process in the User's Manual.

- **Show Driver** — Refer to Tracing to the Immediate Driving Process in the User's Manual.

- **Show Root Cause** — Refer to Tracing to the Root Cause in the User's Manual.

- **Show 'X' Cause (ChaseX)** — Refer to Tracing to the Root Cause of an 'X' in the User's Manual.

- **Show All Possible Drivers** — Refer to Finding All Possible Drivers in the User's Manual.

- **Show Cause from Time** — Refer to Tracing from a Specific Time in the User's Manual.

- **Show Driver from Time** — Refer to Tracing from a Specific Time in the User's Manual.

- **Show Root Cause from Time** — Refer to Tracing from a Specific Time in the User's Manual.

- **View Path Details** — Refer to Causality Path Details in the User's Manual.

- **Preferences** — Refer to Causality Traceback Preferences in the User's Manual.

- **Set Default Action** — Selecting one of the items from the dropdown menu sets that item as the default action when you click the Event Traceback button. The title of the selection is shown in bold type in the Event Traceback dropdown menu and two asterisks (**) are placed after the title to indicate the current default action. For example, **Show Cause** is the default action in Figure 40.

For more information, refer to Using Causality Tracebackin the User's Manual.

Source Toolbar

The Source toolbar allows you to perform several activities on Source windows.

Figure 41. Source Toolbar

Table 36. Source Toolbar Buttons

Button	Name	Shortcuts	Description
	Previous Zero Hits	None	Jump to previous line with zero coverage.
	Next Zero Hits	None	Jump to next line with zero coverage.
	Source Annotation	Menu: **Source > Show Annotation**	Allows Debugging with Source Annotation in every open source file.
	Clear Bookmarks	Menu: **Source > Clear Bookmarks**	Removes any bookmarks in the active source file.

Standard Toolbar

The Standard toolbar contains common buttons that apply to most windows.

Figure 42. Standard Toolbar

Table 37. Standard Toolbar Buttons

Button	Name	Shortcuts	Description
	New File	Menu: **File > New > Source**	Opens a new Source text file. The icon changes to reflect the default file type set with the Set Default Action menu pick from the dropdown menu. Click and hold the button to open a dropdown menu with the following options: • VHDL • Verilog • SystemC • SystemVerilog • Do • Other • Set Default Action
	Open	Menu: **File > Open**	Opens the Open File dialog
	Save	Menu: **File > Save**	Saves the contents of the active window or Saves the current wave window display and signal preferences to a DO file script.
	Reload	Command: Dataset Restart Menu: **File > Datasets**	Reload the current dataset.
	Print	Menu: **File > Print**	Opens the Print dialog box.
	Cut	Menu: Edit > Cut Hotkey: Ctrl+x	
	Copy	Menu: **Edit > Copy** Hotkey: Ctrl+c	
	Paste	Menu: **Edit > Paste** Hotkey: Ctrl+v	
	Undo	Menu: **Edit > Undo**	

Table 37. Standard Toolbar Buttons (continued)

Button	Name	Shortcuts	Description
		Hotkey: Ctrl+z	
	Redo	Menu: **Edit > Redo** Hotkey: Ctrl+y	
	Add Selected to Window	Menu: **Add > to Wave** Hotkey: Ctrl+w	Clicking adds selected objects to the Wave window. Refer to Add Selected to Window Button for more information about the dropdown menu selections.[1] • Set Default Action
	Find	Menu: **Edit > Find** Hotkey: Ctrl+f (Windows) or Ctrl+s (UNIX)	Opens the Find dialog box.
	Collapse All	Menu: **Edit > Expand > Collapse All**	

1. You can set the default insertion location in the Wave window from menus and hotkeys with the PrefWave(InsertMode) preference variable.

Add Selected to Window Button

This button is available when you have selected an object in any of the following windows: Dataflow, List, Locals, Memory, Objects, Process, Schematic, Structure, Watch, and Wave windows. Using a single click, the objects are added to the Wave window. However, if you click-and-hold the button you can access additional options via a dropdown menu.

Figure 43. Add Selected to Window Dropdown Menu
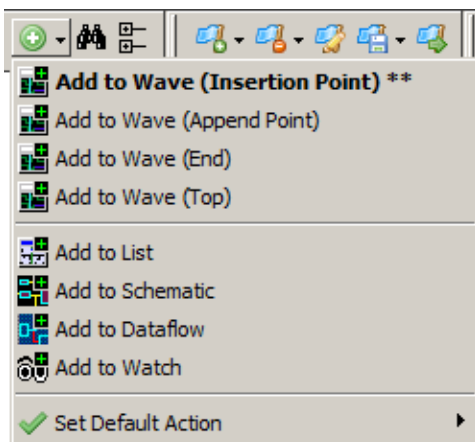
- Add to Wave (Anchor Location) — Adds selected signals above the Insertion Point Bar in the Pathname Pane by default.

- Add to Wave (Append Point) — Adds selected signals below the insertion pointer in the Pathname Pane.

- Add to Wave (End) — Adds selected signals after the last signal in the Wave Window.

- Add to Wave (Top) — Adds selected signals above the first signal in the Wave window.

- Add to List — Adds selected objects to the List Window.

- Add to Dataflow — Adds selected objects to the Dataflow Window.

- Add to Schematic — Adds selected objects to the Schematic Window.

- Add to Watch — Adds selected objects to the Watch Window.

- Set Default Action — Selecting one of the items from the dropdown menu sets that item as the default action when you click the **Add Selected to Window** button. The title of the selection is shown in bold type in the **Add Selected to Window** dropdown menu and two asterisks (******) are placed after the title to indicate the current default action. For example, **Add to Wave (Anchor Location)** is the default action in Figure 43.

- You can change the default.

Step Toolbar

The Step toolbar allows you to step through your source code.

Figure 44. Step Toolbar

Table 38. Step Toolbar Buttons

Button	Name	Shortcuts	Description
⬇	Step Into	Command: step Menu: **Simulate > Run > Step**	Step the current simulation to the next statement.
↪	Step Over	Command: step -over Menu: **Simulate > Run > Step-Over**	Execute HDL statements, treating them as simple statements instead of entered and traced line by line.
⬆	Step Out	Command: step -out	Step the current simulation out of the current function or procedure.

Table 38. Step Toolbar Buttons (continued)

Button	Name	Shortcuts	Description
	Step Into Current	Command: step -current	Step the simulation into the current instance, process, or thread.
	Step Over Current	Command: step -over -current	Step the simulation over the current instance, process, or thread.
	Step Out Current	Command: step -out -current	Step the simulation out of the current instance, process, or thread.

Wave Toolbar

The Wave toolbar allows you to perform specific actions in the Wave window.

Figure 45. Wave Toolbar

Table 39. Wave Toolbar Buttons

Button	Name	Shortcuts	Description
	Show Drivers Show Drivers (source only)	None	Display driver(s) of the selected signal, net, or register in the Dataflow , Schematic and Source windows. Display drivers only in the Source window The source window is not shown if there are no drivers.
	Show Readers Show Readers (source only)	None	Display reader(s) of the selected signal, net, or register in the Dataflow window. Display drivers only in the Source window The source window is not shown if there are no readers.
	Add Contributing Signals	Menu: **Add > To Wave > Contributing Signals**	Creates a group labeled Contributors: <name>, where <name> is the name of the currently selected signal. This group contains the inputs to the process driving <name>.

Table 39. Wave Toolbar Buttons (continued)

Button	Name	Shortcuts	Description
Search: [ ] [▼]	Wave Search Box	Click the box when in transition mode (Falling Edge, Rising Edge, or Any Transition) to cycle through these options.	Text-entry box for the search string. Dropdown button displays previous search strings. Long search times result in the display of a stop icon you can use to cancel the search.
	Search Previous/Next	Previous: Shift+Enter Next: enter	Searches for the next occurrence of the string, either backward or forward in time, from the cursor.
	Search Options		Dropdown button to: • Change Mode: value, rising edge, falling edge, or any transition. • Display the Figure 93 for advanced search behavior. • Clear the value history.

Using the Wave Search Box

You can use the Wave Search box to search the timeline of a selected signal for transitions to a specific value, or just for transitions themselves.

1. Select a signal in the left-hand side of the Wave window.

2. Place a wave cursor where you want to begin the search.

3. Enter a value in the Wave Search box. The value must be of a compatible format type for the signal you selected.

   Alternatively you can use the Search Options button to change the mode from "value" to Rising Edge, Falling Edge, or Any Transition. This populates the Wave Search box with the selected transition type.

4. Use the Search Reverse or Search Forward buttons to locate the next occurrence of the value (or transition).

   For large simulations, if the search takes a long time, the Wave Search box will display a stop icon you can click to stop the search.

Wave Compare Toolbar

The Wave Compare toolbar allows you to quickly find differences in a waveform comparison.

Figure 46. Wave Compare Toolbar
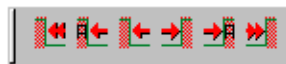
Table 40. Wave Compare Toolbar Buttons

Button	Name	Shortcuts	Description
	Find First Difference	None	Find the first difference in a waveform comparison
	Find Previous Annotated Difference	None	Find the previous annotated difference in a waveform comparison
	Find Previous Difference	None	Find the previous difference in a waveform comparison
	Find Next Difference	None	Find the next difference in a waveform comparison
	Find Next Annotated Difference	None	Find the next annotated difference in a waveform comparison
	Find Last Difference	None	Find the last difference in a waveform comparison

Wave Cursor Toolbar

The Wave Cursor toolbar provides various tools for manipulating cursors in the Wave window.

Figure 47. Wave Cursor Toolbar

Table 41. Wave Cursor Toolbar Buttons

Button	Name	Shortcuts	Description
	Insert Cursor	None	Adds a new cursor to the active Wave window.
	Delete Cursor	Menu: **Wave > Delete Cursor**	Deletes the active cursor.
	Find Previous Transition	Menu: **Edit > Signal Search** Hotkey: Shift + Tab	Moves the active cursor to the previous signal value change for the selected signal.

Table 41. Wave Cursor Toolbar Buttons (continued)

Button	Name	Shortcuts	Description
	Find Next Transition	Menu: **Edit > Signal Search** Hotkey: Tab	Moves the active cursor to the next signal value change for the selected signal.
	Find Previous Falling Edge	Menu: **Edit > Signal Search**	Moves the active cursor to the previous falling edge for the selected signal.
	Find Next Falling Edge	Menu: **Edit > Signal Search**	Moves the active cursor to the next falling edge for the selected signal.
	Find Previous Rising Edge	Menu: **Edit > Signal Search**	Moves the active cursor to the previous rising edge for the selected signal.
	Find Next Rising Edge	Menu: **Edit > Signal Search**	Moves the active cursor to the next rising edge for the selected signal.

Wave Edit Toolbar

The Wave Edit toolbar provides easy access to tools for modifying an editable wave.

Figure 48. Wave Edit Toolbar

Table 42. Wave Edit Toolbar Buttons

Button	Name	Shortcuts	Description
	Insert Pulse	Menu: **Wave > Wave Editor > Insert Pulse** Command: wave edit insert_pulse	Insert a transition at the selected time.
	Delete Edge	Menu: **Wave > Wave Editor > Delete Edge** Command: wave edit delete	Delete the selected transition.
	Invert	Menu: **Wave > Wave Editor > Invert** Command: wave edit invert	Invert the selected section of the waveform.
	Mirror	Menu: **Wave > Wave Editor > Mirror** Command: wave edit mirror	Mirror the selected section of the waveform.

Table 42. Wave Edit Toolbar Buttons (continued)

Button	Name	Shortcuts	Description
	Change Value	Menu: **Wave > Wave Editor > Value** Command: wave edit change_value	Change the value of the selected section of the waveform.
	Stretch Edge	Menu: **Wave > Wave Editor > Stretch Edge** Command: wave edit stretch	Move the selected edge by increasing/decreasing waveform duration.
	Move Edge	Menu: **Wave > Wave Editor > Move Edge** Command: wave edit move	Move the selected edge without increasing/decreasing waveform duration.
	Extend All Waves	Menu: **Wave > Wave Editor > Extend All Waves** Command: wave edit extend	Increase the duration of all editable waves.

Wave Expand Time Toolbar

The Wave Expand Time toolbar provides access to enabling and controlling wave expansion features.

Figure 49. Wave Expand Time Toolbar

Table 43. Wave Expand Time Toolbar Buttons

Button	Name	Shortcuts	Description
	Expanded Time Off	Menu: **Wave > Expanded Time > Off**	turns off the expanded time display (default mode)
	Expanded Time Deltas Mode	Menu: **Wave > Expanded Time > Deltas Mode**	displays delta time steps
	Expanded Time Events Mode	Menu: **Wave > Expanded Time > Events Mode**	displays event time steps
	Expand All Time	Menu: **Wave > Expanded Time > Expand All**	expands simulation time over the entire simulation time range, from 0 to current time
	Expand Time at Active Cursor	Menu: **Wave > Expanded Time > Expand Cursor**	expands simulation time at the simulation time of the active cursor

Table 43. Wave Expand Time Toolbar Buttons (continued)

Button	Name	Shortcuts	Description
	Collapse All Time	Menu: **Wave > Expanded Time > Collapse All**	collapses simulation time over entire simulation time range
	Collapse Time at Active Cursor	Menu: **Wave > Expanded Time > Collapse Cursor**	collapses simulation time at the simulation time of the active cursor

Zoom Toolbar

The Zoom toolbar allows you to change the view of the Wave window.

Figure 50. Zoom Toolbar

Table 44. Zoom Toolbar Buttons

Button	Name	Shortcuts	Description
	Zoom In	Menu: **Wave > Zoom > Zoom In** Hotkey: i, I, or +	Zooms in by a factor of 2x.
	Zoom Out	Menu: **Wave > Zoom > Zoom Out** Hotkey: o, O, or -	Zooms out by a factor of 2x.
	Zoom Full	Menu: **Wave > Zoom > Zoom Full** Hotkey: f or F	Zooms to show the full length of the simulation.
	Zoom in on Active Cursor	Menu: **Wave > Zoom > Zoom Cursor** Hotkey: c or C	Zooms in by a factor of 2x, centered on the active cursor.
	Zoom between Cursors		Zooms in or out to show the range between the last two selected cursors.
	Zoom Other Window		Changes the view in additional instances of the Wave window to match the view of the active Wave window.

Toolbar Tabs

The default GUI toolbar format of multiple separate toolbars can be replaced with a row of tabs containing buttons grouped into common tasks such as editing, debugging, and so forth.

The buttons are context-driven and are either operative or dimmed, depending on which window is currently active. To change the GUI to Toolbar Tab format, set prefToolbar(newEnabled) to 1 (select the **Tools > Edit Preferences**, **By Name** tab, and expand the Toolbar object). You must restart the application for the change to take effect. On restarting the application, the GUI displays the toolbar tabs (Figure 51).

Figure 51. Toolbar Tabs and Overflow Menu
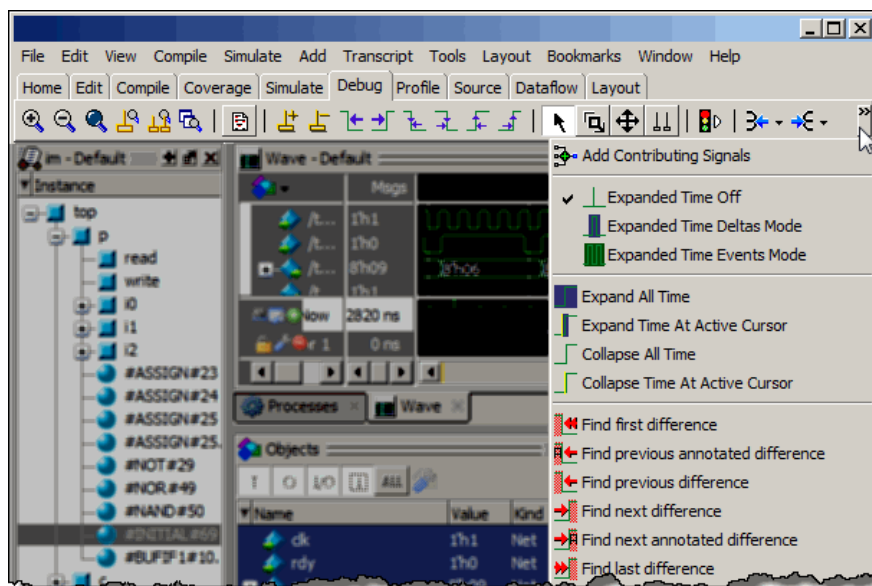
If the application window is too narrow to accommodate all of the buttons and widgets assigned to it, an overflow button will appear on the right edge of the toolbar. Clicking the overflow button opens a drop-menu (Figure 51) which displays the remaining buttons. You can change the overflow from a drop-menu to a scrolling menu by setting prefToolbar(newScrollable) to 1.

The following sections describe the tabs and their associated buttons.

- Compile Toolbar Tab
- Coverage Toolbar Tab
- Debug Toolbar Tab
- Edit Toolbar Tab
- Home Toolbar Tab
- Layout Toolbar Tab
- Profile Toolbar Tab
- Schematic and Dataflow Toolbar Tab
- Simulate Toolbar Tab
- Windows With Dedicated Toolbars
- Toolbar Visibility and Layout

Compile Toolbar Tab

The Compile Toolbar Tab provides access to compile actions.

Figure 52. Compile Tab
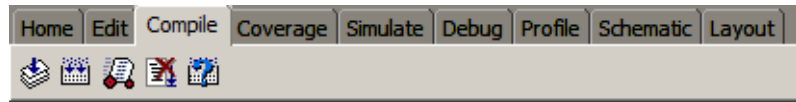
Table 45. Compile Toolbar Tab Buttons

Button	Name	Shortcuts	Description
	Compile	Command: vcom or vlog Menu: **Compile > Compile**	Opens the Compile Source Files dialog box.
	Compile All	Command: vcom or vlog Menu: **Compile > Compile All**	Compiles all files in the open project.
	Simulate	Command: vsim Menu: **Simulate > Start Simulation**	Opens the Start Simulation dialog box.
	Break	Menu: **Simulate > Break** Hotkey: Break	Stop a compilation, elaboration, or the current simulation run.
	Compile Out of Date		Recompile if changes have been made to any source files.

Coverage Toolbar Tab

The Coverage toolbar tab provides tools for filtering code coverage data in the Structure and Instance Coverage windows.

Figure 53. Coverage Tab
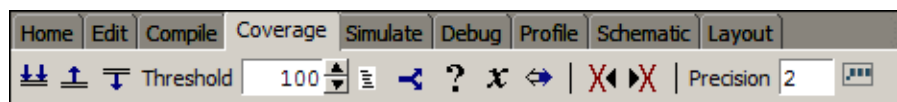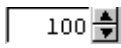
Table 46. Coverage Toolbar Tab Buttons

Button	Name	Shortcuts	Description
	Enable Filtering	None	Enables display filtering of coverage statistics in the Structure and Instance Coverage windows.
	Threshold Above	None	Displays all coverage statistics above the Filter Threshold for selected columns.
	Threshold Below	None	Displays all coverage statistics below the Filter Threshold for selected columns
100	Filter Threshold	None	Specifies the display coverage percentage for the selected coverage columns
	Statement	None	Applies the display filter to all Statement coverage columns in the Structure and Instance Coverage windows.
	Branch	None	Applies the display filter to all Branch coverage columns in the Structure and Instance Coverage windows.
?	Condition	None	Applies the display filter to all Condition coverage columns in the Structure and Instance Coverage windows.
x	Expression	None	Applies the display filter to all Expression coverage columns in the Structure and Instance Coverage windows.
	Toggle	None	Applies the display filter to all Toggle coverage columns in the Structure and Instance Coverage windows.
	Previous Zero Hits	None	Jump to previous line with zero coverage.
	Next Zero Hits	None	Jump to next line with zero coverage.
Precision 2	Set Precision for VMgmt	Menu: **Verification Browser > Set Precision**	A text entry box that allows you to control the precision of the data in the Verification Browser window.
	Restore Default Precision		Restores the precision to the default value (2).

Debug Toolbar Tab

The Debug toolbar tab provides tools for debugging in various windows.

Figure 54. Debug Tab

Table 47. Debug Toolbar Tab Buttons

Button	Name	Shortcuts	Description
	Zoom In	Menu: **Wave > Zoom > Zoom In** Hotkey: i, I, or +	Zooms in by a factor of 2x.
	Zoom Out	Menu: **Wave > Zoom > Zoom Out** Hotkey: o, O, or -	Zooms out by a factor of 2x.
	Zoom Full	Menu: **Wave > Zoom > Zoom Full** Hotkey: f or F	Zooms to show the full length of the simulation.
	Zoom in on Active Cursor	Menu: **Wave > Zoom > Zoom Cursor** Hotkey: c or C	Zooms in by a factor of 2x, centered on the active cursor.
	Zoom between Cursors		Zooms in or out to show the range between the last two selected cursors.
	Zoom Other Window		Changes the view in additional instances of the Wave window to match the view of the active Wave window.
	Source Annotation	Menu: **Source > Show Annotation**	Allows Debugging with Source Annotation in every open source file.
	Source Hyperlinking	None	Toggles display of hyperlinks in design source files.

Table 47. Debug Toolbar Tab Buttons (continued)

Button	Name	Shortcuts	Description
	Insert Cursor	None	Adds a new cursor to the active Wave window.
	Delete Cursor	Menu: **Wave > Delete Cursor**	Deletes the active cursor.
	Find Previous Transition	Menu: **Edot > Signal Search** Hotkey: Shift + Tab	Moves the active cursor to the previous signal value change for the selected signal.
	Find Next Transition	Menu: **Edit > Signal Search** Hotkey: Tab	Moves the active cursor to the next signal value change for the selected signal.
	Find Previous Falling Edge	Menu: **Edit > Signal Search**	Moves the active cursor to the previous falling edge for the selected signal.
	Find Next Falling Edge	Menu: **Edit > Signal Search**	Moves the active cursor to the next falling edge for the selected signal.
	Find Previous Rising Edge	Menu: **Edit > Signal Search**	Moves the active cursor to the previous rising edge for the selected signal.
	Find Next Rising Edge	Menu: **Edit > Signal Search**	Moves the active cursor to the next rising edge for the selected signal.
	Select Mode	Menu: **Dataflowor Schematic > ,Mouse Mode > Select Mode**	Set the left mouse button to select mode and middle mouse button to zoom mode.
	Zoom Mode	Menu: **Dataflow or Schematic > ,Mouse Mode > ZoomMode**	Set left mouse button to zoom mode and middle mouse button to pan mode.
	Pan Mode	Menu: **Dataflow or Schematic > ,Mouse Mode > Pan Mode**	Set left mouse button to pan mode and middle mouse button to zoom mode.
	Two Cursor Mode	Menu: **Wave > Mouse Mode > Two Cursor**	Sets two cursors in Wave window. First cursor moves with LMB, second cursor with MMB.
	Stop Drawing	None	Halt any drawing currently happening in the window.
	Show Cause	Command: find drivers -active	Traces a selected wave signal from the current time back to the first sequential element. See #unique_60_Connect_42_id8f7d 4c37-c85a-4c3f-8dbb-c005e1bb 1757 for more information.

Table 47. Debug Toolbar Tab Buttons (continued)

Button	Name	Shortcuts	Description
			• Set Default Action
	Show Drivers Show Drivers (source only)	None	Display driver(s) of the selected signal, net, or register in the Dataflow, Schematic and Source windows. Display drivers only in the Source window The source window is not shown if there are no drivers.
	Show Readers Show Readers (source only)	None	Display reader(s) of the selected signal, net, or register in the Dataflow window. Display drivers only in the Source window The source window is not shown if there are no readers.
	Add Contributing Signals	Menu: **Add > To Wave > Contributing Signals**	Creates a group labeled Contributors: <name>, where <name> is the name of the currently selected signal. This group contains the inputs to the process driving <name>.
	Expanded Time Off	Menu: **Wave > Expanded Time**	turns off the expanded time display (default mode)
	Expanded Time Deltas Mode	Menu: **Wave > Expanded Time > Deltas Mode**	displays delta time steps
	Expanded Time Events Mode	Menu: **Wave > Expanded Time > Events Mode**	displays event time steps
	Expand All Time	Menu: **Wave > Expanded Time > Expand All**	expands simulation time over the entire simulation time range, from 0 to current time
	Expand Time at Active Cursor	Menu: **Wave > Expanded Time > Expand Cursor**	expands simulation time at the simulation time of the active cursor
	Collapse All Time	Menu: **Wave > Expanded Time > Collapse All**	collapses simulation time over entire simulation time range
	Collapse Time at Active Cursor	Menu: **Wave > Expanded Time > Collapse Cursor**	collapses simulation time at the simulation time of the active cursor

Table 47. Debug Toolbar Tab Buttons (continued)

Button	Name	Shortcuts	Description
	Find First Difference	None	Find the first difference in a waveform comparison
	Find Previous Annotated Difference	None	Find the previous annotated difference in a waveform comparison
	Find Previous Difference	None	Find the previous difference in a waveform comparison
	Find Next Difference	None	Find the next difference in a waveform comparison
	Find Next Annotated Difference	None	Find the next annotated difference in a waveform comparison
	Find Last Difference	None	Find the last difference in a waveform comparison
Search:	Wave Search Box	Click the box when in transition mode (Falling Edge, Rising Edge, or Any Transition) to cycle through these options.	Text-entry box for the search string. Dropdown button displays previous search strings. Long search times result in the display of a stop icon you can use to cancel the search.
	Search Previous/Next	Previous: Shift+Enter Next: enter	Searches for the next occurrence of the string, either backward or forward in time, from the cursor.
	Search Options	Menu: **Edit > Signal**	Dropdown button to: • Change Mode: value, rising edge, falling edge, or any transition. • Display the Figure 93 for advanced search behavior. • Clear the value history.

Edit Toolbar Tab

The Edit toolbar tab provides tools for modifying an editable waveform.

Figure 55. Edit Tab

Table 48. Edit Toolbar Tab Buttons

Button	Name	Shortcuts	Description
	New File	Menu: **File > New > Source**	Opens a new Source text file. The icon changes to reflect the default file type set with the Set Default Action menu pick from the dropdown menu. Click and hold the button to open a dropdown menu with the following options: • VHDL • Verilog • SystemC • SystemVerilog • Do • Other • Set Default Action
	Insert Pulse	Menu: **Wave > Wave Editor > Insert Pulse** Command: wave edit insert_pulse	Insert a transition at the selected time.
	Delete Edge	Menu: **Wave > Wave Editor > Delete Edge** Command: wave edit delete	Delete the selected transition.
	Invert	Menu: **Wave > Wave Editor > Invert** Command: wave edit invert	Invert the selected section of the waveform.
	Mirror	Menu: **Wave > Wave Editor > Mirror** Command: wave edit mirror	Mirror the selected section of the waveform.
	Change Value	Menu: **Wave > Wave Editor > Value** Command: wave edit change_value	Change the value of the selected section of the waveform.
	Stretch Edge	Menu: **Wave > Wave Editor > Stretch Edge** Command: wave edit stretch	Move the selected edge by increasing/decreasing waveform duration.
	Move Edge	Menu: **Wave > Wave Editor > Move Edge** Command: wave edit move	Move the selected edge without increasing/decreasing waveform duration.

Table 48. Edit Toolbar Tab Buttons (continued)

Button	Name	Shortcuts	Description
	Extend All Waves	Menu: **Wave > Wave Editor > Extend All Waves** Command: wave edit extend	Increase the duration of all editable waves.
	Edit Mode	Menu: **Wave or Dataflow > Mouse Mode > Edit Mode**	Set mouse to Edit Mode, where you drag the left mouse button to select a range and drag the middle mouse button to zoom.

Home Toolbar Tab

The Home toolbar tab contains common buttons that apply to most windows.

Figure 56. Home Tab
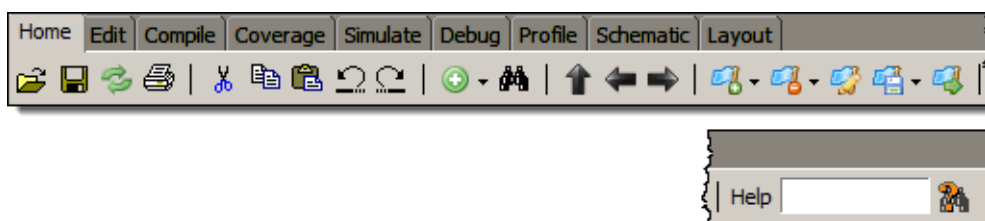
Table 49. Home Toolbar Tab Buttons

Button	Name	Shortcuts	Description
	Open	Menu: **File > Open**	Opens the Open File dialog
	Save	Menu: **File > Save**	Saves the contents of the active window or Saves the current wave window display and signal preferences to a DO file script.
	Reload	Command: Dataset Restart Menu: **File > Datasets**	Reload the current dataset.
	Print	Menu: **File > Print**	Opens the Print dialog box.
	Cut	Menu: **Edit > Cut** Hotkey: Ctrl+x	
	Copy	Menu: **Edit > Copy** Hotkey: Ctrl+c	
	Paste	Menu: **Edit > Paste**	

Table 49. Home Toolbar Tab Buttons (continued)

Button	Name	Shortcuts	Description
		Hotkey: Ctrl+v	
	Undo	Menu: **Edit > Undo** Hotkey: Ctrl+z	
	Redo	Menu: **Edit > Redo** Hotkey: Ctrl+y	
	Add Selected to Window	Menu: Add > to Wave Hotkey: Ctrl+w	Clicking adds selected objects to the Wave window. Refer to Add Selected to Window Button for more information about the dropdown menu selections.¹ • Set Default Action
	Find	Menu: **Edit > Find** Hotkey: Ctrl+f (Windows) or Ctrl+s (UNIX)	Opens the Find dialog box.
	Environment Up	Command: env.. Menu: **File > Environment**	Changes your environment up one level of hierarchy.
	Environment Back	Command: env -back Menu: **File > Environment**	Change your environment to its previous location.
	Environment Forward	Command: env -forward Menu: **File > Environment**	Change your environment forward to a previously selected environment.
	Add Bookmark	Command (Wave window only): bookmark add wave Menu (Wave window only): **Add > To > Wave > Bookmark**	Clicking this button bookmarks the current view of the active window. Click and hold the button to open a drop down menu with the following options: • Add Current View • Add Custom ... • Set Default Action
	Delete All Bookmarks	Command (Wave window only): bookmark delete wave -all	Removes all bookmarks, after prompting for your confirmation. Click and hold the button to open a drop down menu with the following options: • Active Window • All Windows

Table 49. Home Toolbar Tab Buttons (continued)

Button	Name	Shortcuts	Description
	Manage Bookmarks	None	Displays the Manage Bookmarks dialog box.
	Reload from File	None	Reloads bookmarks from the *bookmarks.do* file. • Set Default Action
	Jump to Bookmark	Command (Wave window only): bookmark goto wave <name>	Displays bookmarks grouped by window. Select the bookmark you want to display.
Help	Search Documentation	None	A text entry box for your search string.
	Search Documentation	Hotkey: Enter	Activates the search for the term you entered into the text entry box.

1. You can set the default insertion location in the Wave window from menus and hotkeys with the PrefWave(InsertMode) preference variable.

Layout Toolbar Tab

The Layout toolbar tab allows you to select a predefined or user-defined layout of the graphical user interface.

Refer to the section Simulator GUI Layout Customization for more information.

Figure 57. Layout Tab
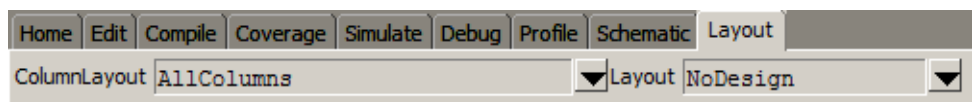
| Home | Edit | Compile | Coverage | Simulate | Debug | Profile | Schematic | Layout |

ColumnLayout `AllColumns` ▼ Layout `NoDesign` ▼

Table 50. Layout Toolbar Tab Buttons
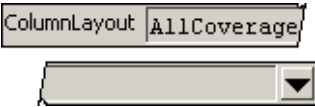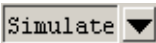
Button	Name	Shortcuts	Description
ColumnLayout `AllCoverage` ▼	Column Layout	Menu: **Verification Browser > Configure Column Layout**	A dropdown box that allows you to specify the column layout for the active window.
`Simulate` ▼	Change Layout	Menu: **Layout > <layoutName>**	A dropdown box that allows you to select a GUI layout. • NoDesign • Simulate

Table 50. Layout Toolbar Tab Buttons (continued)

Button	Name	Shortcuts	Description
			• Coverage
			• VMgmt

Profile Toolbar Tab

The Profile toolbar tab provides access to tools related to the profiling windows (Ranked, Calltree, Design Unit, and Structural.

Figure 58. Profile Tab

Table 51. Profile Toolbar Tab Buttons

Button	Name	Shortcuts	Description
	Collapse Sections	Menu: **Tools > Profile > Collapse Sections**	Toggle the reporting for collapsed processes and functions.
0.10	Profile Cutoff	None	Display performance and memory profile data equal to or greater than set percentage.
	Refresh Profile Data	None	Refresh profile performance and memory data after changing profile cutoff.
	Save Profile Results	Menu: **Tools > Profile > Profile Report**	Save profile data to output file (prompts for file name).
	Performance Profiling	Menu: **Tools > Profile > Performance**	Enable collection of statistical performance data.
	Memory Profiling	Menu: **Tools > Profile > Memory**	Enable collection of memory usage data.

Schematic and Dataflow Toolbar Tab

The Schematic toolbar tab provides access to tools for manipulating highlights and signals in the Dataflow and Schematic windows.

Figure 59. Schematic Tab

Table 52. Schematic Toolbar Tab Buttons

Button	Name	Shortcuts	Description
	Trace Input Net to Event	Menu: **Tools > Trace > Trace next event**	Move the next event cursor to the next input event driving the selected output.
	Trace Set	Menu: **Tools > Trace > Trace event set**	Jump to the source of the selected input event.
	Trace Reset	Menu: **Tools > Trace > Trace event reset**	Return the next event cursor to the selected output.
	Trace Net to Driver of X	Menu: **Tools > Trace > TraceX**	Step back to the last driver of an unknown value.
	Expand Net to all Drivers	None	Display driver(s) of the selected signal, net, or register.
	Expand Net to all Drivers and Readers	None	Display driver(s) and reader(s) of the selected signal, net, or register.
	Expand Net to all Readers	None	Display reader(s) of the selected signal, net, or register.
	Remove All Highlights	Menu: **Dataflow > Remove Highlight** or **Schematic > Edit > Remove Highlight**	Clear the highlighting identifying the path you have traversed through the design. Click and hold the button to open a drop down menu with the following options: • Remove All Highlights • Remove Selected Highlights • Set Default Action
	Delete Content	Menu: **Dataflow > Delete** or **Schematic > Edit > Delete** **Schematic > Edit > Delete All**	Delete the selected signal. Click and hold the button to open a drop down menu with the following options: • Delete Selected • Delete All • Set Default Action

Table 52. Schematic Toolbar Tab Buttons (continued)

Button	Name	Shortcuts	Description
	Regenerate	Menu: **Dataflow > Regenerate** or **Schematic > Edit > Regenerate**	Redraws the current schematic view to better take advantage of the available space. For example, after adding or removing elements.
	Enable 1-Click Mode		Enables single-click sprout expansion. Default is double-click to sprout.
	Show Wave	Menu: **Dataflow > Show Wave**	Display the embedded wave viewer pane.

Simulate Toolbar Tab

The Simulate Toolbar tab allows you to control aspects of simulation including running the simulation and stepping through your code.

Figure 60. Simulate Tab

Table 53. Simulate Toolbar Tab Buttons
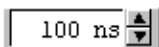
Button	Name	Shortcuts	Description
	Source Hyperlinking	None	Toggles display of hyperlinks in design source files.
	Step Into	Command: step Menu: **Simulate > Run > Step**	Step the current simulation to the next statement.
	Step Over	Command: step -over Menu: **Simulate > Run > Step -over**	Execute HDL statements, treating them as simple statements instead of entered and traced line by line.
	Step Out	Command: step -out	Step the current simulation out of the current function or procedure.
	Step Into Current	Command:step -current	Step the simulation into the current instance, process, or thread.
	Step Over Current	Command: step -over -current	Step the simulation over the current instance, process, or thread.

Table 53. Simulate Toolbar Tab Buttons (continued)

Button	Name	Shortcuts	Description
	Step Out Current	Command: step -out -current	Step the simulation out of the current instance, process, or thread.
	Restart	Command: restart Menu: **Simulate > Run > Restart**	Reload the design elements and reset the simulation time to zero, with the option of maintaining various settings and objects.
100 ns	Run Length	Command: run Menu: **Simulate > Runtime Options**	Specify the run length for the current simulation.
	Run	Command: run Menu: **Simulate > Run > Run<*default_run_length*>**	Run the current simulation for the specified run length.
	Continue Run	Command: run -continue Menu: **Simulate > Run > Continue**	Continue the current simulation run until the end of the specified run length or until it hits a breakpoint or specified break event.
	Run All	Command: run -all Menu: **Simulate > Run > Run -all**	Run the current simulation forever, or until it hits a breakpoint or specified break event.
	Simulate	Command: vsim Menu: **Simulate > Start Simulation**	Opens the Start Simulation dialog box.
	Break	Menu: Simulate > Break Hotkey: Break	Immediate stop of a compilation, elaboration, or simulation run. Similar to hitting a breakpoint if the simulator is in the middle of a process.
	Stop -sync	None	Stop simulation the next time time/delta is advanced.
	C Interrupt	Command: cdbg interrupt Menu: **Tools > C Debug > C Interrupt**	Reactivate the C debugger when stopped in HDL code.
	Edit Breakpoints	Menu: **Tools > Breakpoint**	Enable breakpoint editing, loading, and saving.

Windows With Dedicated Toolbars

Buttons that function only in a specific window are available in a bar at the top of the window.

Click the Show/Hide Toolbar button in the window title bar to display the window specific buttons (Figure 61). You can add buttons to any currently open window. Refer to Customizing Button and Tab Location for more information.

Figure 61. Window Specific Buttons
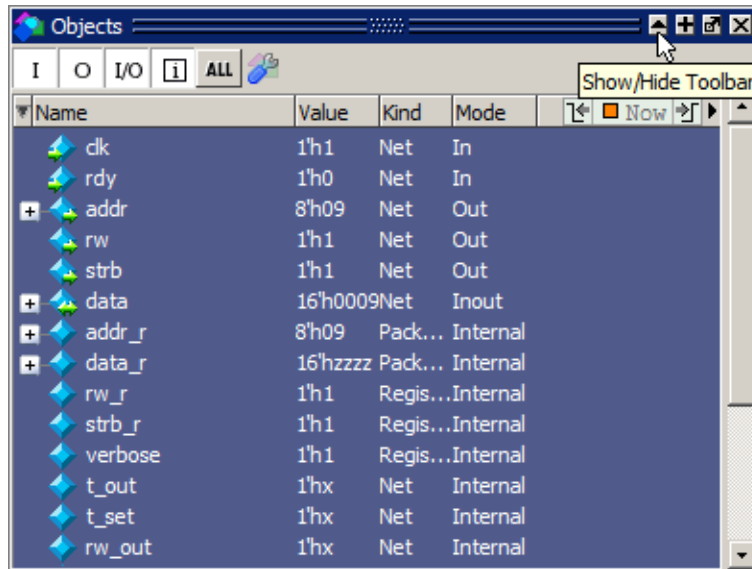
The following windows have default buttons or widgets:

- ATV Window

- FSM Viewer Window

- Memory Data Window

- Objects Window

- Processes Window

- Source Window

- Verification Results Analysis Window

Toolbar Visibility and Layout

You can customize the display of the Toolbar Tabs in the main GUI and undocked windows.

Procedure

1. Right-click in the tab or toolbar area of the GUI.

2. Select one of the following items from the popup menu (Table 54):

Table 54. Toolbar Tab Popup Menu

Popup Menu Item	Description
Hide Tabs	Hides all tabs except for the currently active tab.
Edit Toolbars	Opens the Toolbar Tab Widget Toolbox and locks the GUI into edit mode.
Remove <name> Tab	Removes the tab that is under the cursor.
Add Tab	Opens the Add/Create Tab dialog box. Use this dialog box to create a new user-defined tab or restore a previously defined tab.
Reset Tabs	Resets GUI to the default tabs and order of tabs. Removes user-defined tabs.
Reset <name> Toolbar	Resets the selected tab to the default buttons and widgets.
Reset All Toolbars	Resets all toolbar tabs to their default buttons and widgets. User-defined toolbar tabs default to an empty toolbar.

3. You can also remove all toolbars from the GUI by selecting **Window > Show Toolbar**. This removes toolbar tabs from the main GUI only.

Creating and Restoring Toolbar Tabs

You can create your own tabs and populate them with the buttons and widgets you want to use. In addition you can restore tabs you have deleted from the main GUI or an undocked window.

Procedure

Right-click in the toolbar area of the main GUI or the undocked window you want to modify and select **Add Tab** from the popup menu to open the **Add/Create Tab** dialog box.

- To create a user defined toolbar tab, enter a name in the **Tab Name** field then click **Add**. The new tab is added to the main GUI or the currently active undocked window.

- To restore an existing toolbar tab, select a tab name from the drop down list in the Tab Name field. The tab and all buttons currently registered to that tab are added to the GUI or undocked window.

Customizing Button and Tab Location

You can add and delete buttons from any Toolbar Tab or window and add buttons to windows that do not have default buttons associated with them.

Toolbar Tab Widget Toolbox

Toolbar Tab Widget Toolbox

The Widget Toolbox allows you to change the default layout of the Toolbar Tabs and buttons on each tab and window.

Figure 62. Toolbar Widget Toolbox
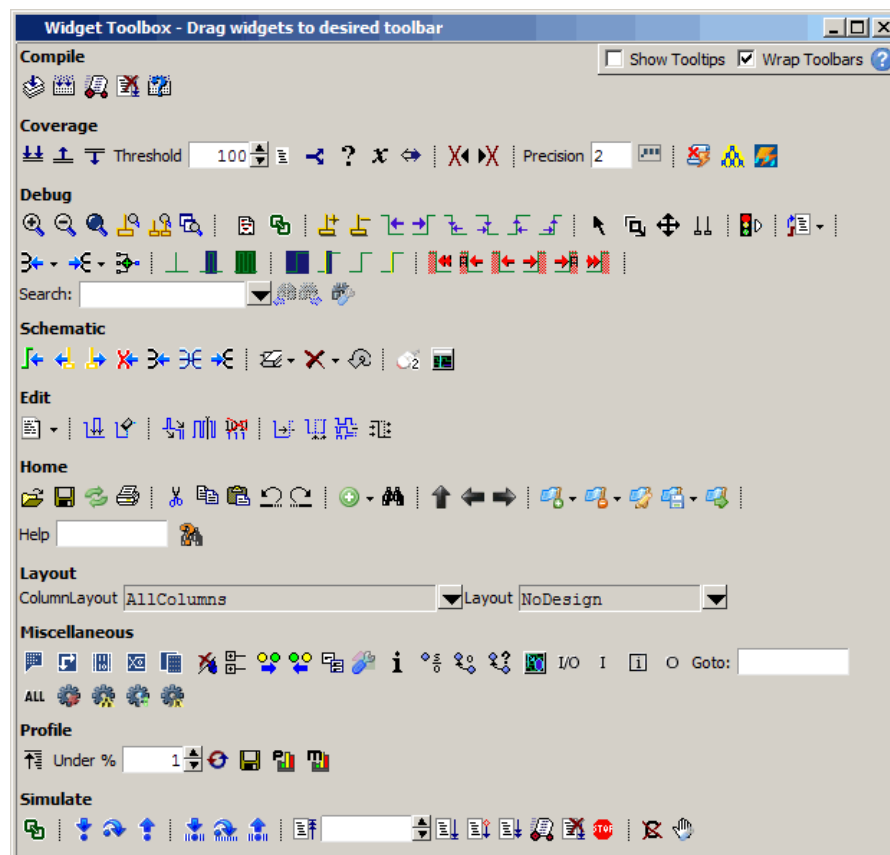
The GUI locks into edit mode when the Widget Toolbox is open preventing interaction with your design and allowing you to change the following items in the GUI:

- Button Location — Add or delete buttons from the main GUI or an undocked window by dragging and dropping. All currently open windows will display a toolbar area at the top of the window where you can add buttons. The toolbar area in a window will disappear if it does not contain any buttons when the Widget Toolbox closes. To delete buttons, drag the button to the Widget

Toolbox or outside of the main GUI and drop it. Refer to Windows With Dedicated Toolbars for more information on windows with dedicated toolbar buttons.

- Tab Ordering — Change the order of the tabs by dragging the tab to a new location in the GUI. You can also change tab order in undocked windows. Tab order in undocked windows is saved separately from the main GUI and reloaded every time you undock a modified window.

Tabbed Toolbar Mapping to Default Toolbars

The following table provides you with the toolbar tab locations of the 10.2 and earlier toolbars.

Table 55. Pre-10.3 Toolbar Mapping to Toolbar Tab Location

Default Toolbar	Toolbar Tab
Bookmarks Toolbar Help Toolbar Standard Toolbar	Home Toolbar Tab
Wave Edit Toolbar	Edit Toolbar Tab
Compile Toolbar	Compile Toolbar Tab
Coverage Toolbar	Coverage Toolbar Tab
Precision Toolbar	Coverage Toolbar Tab
Process Toolbar	
Simulate Toolbar Step Toolbar	Simulate Toolbar Tab
Mode Toolbar Source Toolbar Wave Toolbar Wave Compare Toolbar Wave Cursor Toolbar Wave Expand Time Toolbar Zoom Toolbar	Debug Toolbar Tab
Profile Toolbar	Profile Toolbar Tab
Dataflow Toolbar Schematic Toolbar	Schematic and Dataflow Toolbar Tab
Column Layout Toolbar Layout Toolbar	Layout Toolbar Tab
Analysis Toolbar	Code Coverage Analysis Window

Table 55. Pre-10.3 Toolbar Mapping to Toolbar Tab Location (continued)

Default Toolbar	Toolbar Tab
ATV Toolbar	ATV Window
FSM Toolbar	FSM Viewer Window
Memory Toolbar	Memory Data Window
Objectfilter Toolbar	Object Window Toolbar
Process Toolbar	Process Window Toolbar
Bookmarks Toolbar	Source Window Toolbar

Chapter 4
Window Reference

The following table summarizes all of the available windows.

Table 56. GUI Windows

Window	Description
ATV Window	Displays a graphical, time-based view of your SystemVerilog and PSL assertions.
Assertions Window	manages SystemVerilog and PSL assertions.
Call Stack Window	Displays the current call stack, allowing you to debug your design by analyzing the depth of function calls.
Capacity Window	Displays capacity data (memory usage) about SystemVerilog constructs.
Class Graph Window	Displays interactive relationships of SystemVerilog classes in graphical form.
Class Instances Window	Displays class instances.
Class Tree Window	Displays interactive relationships of SystemVerilog classes in tabular form.
Code Coverage Analysis Window	Displays missing code coverage, details, and code coverage exclusions.
Cover Directives Window	Manages SystemVerilog and PSL cover directives.
Covergroups Window	Manages SystemVerilog covergroups.
Coverage Details Window	contains details about coverage metrics based on selections in other coverage windows.
Dataflow Window	Displays "physical" connectivity and lets you trace events (causality).
Files Window	Displays the source files and their locations for the loaded simulation.
FSM List Window	Lists all recognized FSMs in the design.
FSM Viewer Window	Graphically represents a recognized FSM.
Instance Coverage Window	Displays coverage statistics for each instance in a flat, non-hierarchical view.
Library Window	Lists design libraries and compiled design units.

Table 56. GUI Windows (continued)

Window	Description
List Window	Shows waveform data in a tabular format.
Locals Window	Displays data objects that are immediately visible at the current execution point of the selected process.
Memory List Window Memory Data Window	Show memories and their contents.
Message Viewer Window	Allows easy access, organization, and analysis of Note, Warning, Errors or other messages written to transcript during simulation.
Objects Window	Displays all declared data objects in the current scope.
OVM-Aware Debug Windows	Assists in debugging OVM testbenches.
Processes Window	Displays all processes that are scheduled to run during the current simulation cycle.
Profiling Window	Displays performance and memory profiling data.
Projects	Provides access to information about Projects.
Schematic Window	Displays information about the design in schematic format.
Source Window	Provides a text editor for viewing and editing files.
Structure Window Also known as the "sim" window.	Displays hierarchical view of active simulation. Name of window is either "sim" or "<dataset_name>".
Transaction View Window	Displays the details of Questa Verification IP transaction instances. Not available for any other type of transactions.
Transcript Window	Keeps a running history of commands and messages and provides a command-line interface.
UVM Details Window	Displays stream and config database information about UVM items.
Verification Management Browser Window	Displays information about UCDB tests for the purpose of managing the verification process.
Verification Results Analysis Window Verification Test Analysis Window Verification Tracker Window Verification Trender Window	Displays information about UCDB tests and Verification Plan information for the purpose of managing the verification process.
Watch Window	Displays signal or variable values at the current simulation time.

Table 56. GUI Windows (continued)

Window	Description
Wave Window	Displays waveforms.

Assertions Window

ATV Window

Call Stack Window

Capacity-Object and Capacity-Line Windows

Class Graph Window

Class Instances Window

Class Tree Window

Code Coverage Analysis Window

Viewing Code Coverage Data and Current Exclusions

Cover Directives Window

Coverage Details Window

Covergroups Window

Dataflow Window

Dataflow Window Tasks

Files Window

FSM List Window

FSM Viewer Window

FSM Viewer Window Tasks

Instance Coverage Window

Library Window

List Window

List Window Tasks

Locals Window

Memory Data Window

Memory List Window

Saving Memory Formats in a DO File

Saving Memories to the WLF File

Message Viewer Window

Objects Window

Objects Window Tasks

Pending Exclusions Window

Processes Window

Profiling Windows

Schematic Window

Incremental Schematic Options Dialog Box

Source Window

Using the Source Window

Structure Window

Structure Window Tasks

Assertions Window

To access:

- **View > Coverage > Assertions**

- view assertions command

Use this window to view all embedded and external assertions that were successfully compiled and simulated during the current session.

Description

Figure 63. Assertions Window

Objects

Table 57. Assertions Window Columns

Column Title	Description
Active Count	The number of active assertion attempts at the current time. Enabled if the simulator is invoked with any of the following: • vsim -assertcounts • vsim -assertdebug • the AssertionCover variable set to 1 in your *modelsim.ini* file • the AssertionDebug variable set to 1 in your *modelsim.ini* file
Assertion Expression	Displays the actual assertion expression.
Assertion Type	Indicates the assertion type: Immediate or Concurrent.
Attempt Count	The number of times the assertion has been attempted. This is the number of assertion clocks for clocked assertions, or the number of passes and fails for unclocked assertions. This column is

Table 57. Assertions Window Columns (continued)

Column Title	Description
	populated when you specify -assertcounts or -assertdebug for vsim, or if either the AssertionCover or AssertionDebug *.ini* file variable is set to 1.
ATV	Indicates the status of assertion thread viewing: on or off. Only enabled if the simulator is invoked with the -assertdebug argument or the AssertionDebug *.ini* file variable is set to 1.
Cumulative Threads	The cumulative thread count for the assertion.
	This count is designed to highlight those directives that are starting too many attempts.
	For example, given the assertion:
	<pre>assert property ((@posedge clk) a |=> b);</pre>
	If 'a' is true throughout the simulation, then the above assertion will start a brand new attempt at every clock. An attempt, once started, will only be alive until the next clock. So this assertion will not appear abnormally high in the Memory and Peak Memory columns, but it will have a high count in the Cumulative Threads column.
Design Unit	Identifies the design unit to which the assertion is bound.
Design Unit Type	Identifies the HDL type of the design unit.
Disable Count	The number of assertion attempts that have been disabled due to either an *abort* expression becoming true (PSL) or a *disable if* expression becoming true (SVA).
	This column is populated when you specify -assertcounts or -assertdebug for vsim, or if either the AssertionCover or AssertionDebug *.ini* file variable is set to 1.
Enable	Identifies whether failure checking is active for the assertion.
EOS Note	Identifies when assertion directives are active at the end of simulation (EOS): on or off.
	Refer to the assertion active command.
	If the assert directive is strong, the EOS Note column will report both the "active at end of simulation" note along with a strong error message.
Failure Count	Total number of times the assertion has failed in the current simulation.
Failure Limit	The number of times the simulator will respond to a failure event on an assertion.
Failure Log	enabled — failure messages will be logged to the transcript.
	disabled — failure messages will not be logged to the transcript.
Formal Radius	Indicates that the property has been verified to a depth of x number of cycles in the formal analysis.

Table 57. Assertions Window Columns (continued)

Column Title	Description
	Shown as positive integer. Data appears in this column only during post-process analysis (Coverage View mode).
Formal Status	Indicates formal analysis has been performed. Data appears in this column only during post-process analysis (Coverage View mode). Displayed values include: • blank — no formal analysis has been performed • assumption — indicates that property was used as an assumption in the formal analysis • conflict — indicates conflict when inconsistent data merged in formal analysis • failure — indicates formal analysis has determined that property can fail • inconclusive — indicates formal analysis has not proved or falsified the property. See Proof Radius column for amount of formal analysis performed. • proof — indicates formal analysis has proven that property cannot fail for all legal stimulus • vacuous — indicates formal analysis has proven that the antecedent of the property cannot be reached; property needs to be examined closer
FPSA Actions	Displays a matrix of information relating the current action for each possible state: Failure, Pass, Start, and Antecedent. The field will always show four letters that indicate the current action: • C - Continue • B - Break • E - Exit • S - Subroutine Call where the order of the letters relates to the state: • F - Failure • P - Pass • S - Start • A - Antecedent For example, if you see CCSB, you can derive the following: • Failure state - Continue action • Pass state - Continue action • Start state - Subroutine Call action • Antecedent state - Break action

Table 57. Assertions Window Columns (continued)

Column Title	Description
Included	Indicates whether the Assertion is included in (check mark) or excluded from (X mark) aggregate statistics and reports.
Language	Identifies the HDL used to write the assertion.
Memory	Tracks the current memory used by the assertion.
Name	Identifies the assert directive you specified in the assertion code.
Pass Count	The total number of times the assertion has passed in the current simulation. This column is populated when you specify -assertcounts or -assertdebug for vsim, or if either the AssertionCover or AssertionDebug *.ini* file variable is set to 1. When the simulation is run without -assertcounts or -assertdebug, the Pass Count shows only the pass status (0 = never passed; 1 = passed any number of times).
Pass Log	enabled — pass messages are logged to the transcript. disabled — pass messages are not logged to the transcript. Only enabled if the simulator is invoked with the -assertdebug argument, or the AssertionDebug *.ini* file variable is set to 1.
Peak Active Count	The peak simultaneously active thread count that has occurred up to the current time. Peak Active Count will also be shown in reports created with the vcover report command. Only enabled if the simulator is invoked with the -assertdebug argument, or the AssertionDebug *.ini* file variable is set to 1.
Peak Memory	The peak memory used by the assertion.
Peak Memory Time	Indicates the simulation run time at which the peak memory usage occurred.
Vacuous Count	The number of assertion attempts that have succeeded vacuously, that is, if the left hand side of an implication is false on a clock edge. This column is populated when you specify -assertcounts or -assertdebug for vsim, or if either the AssertionCover or AssertionDebug *.ini* file variable is set to 1.

Table 58. Assertions Window Popup

Popup Menu Item	Description
Add	Add information about the selected assertions to the specified window.
View Source	Opens a source file for the selected assertion.
Enable ATV	Enables an assertion for use with the ATV Window.

Table 58. Assertions Window Popup (continued)

Popup Menu Item	Description
View ATV	Opens an ATV window for the selected assertion.
Report	Generates a report about the selected assertion.
Configure	Allows you to configure the simulators behavior for the selected assertion.
Enable	Enables checking of the assertion for failures during the simulation
Failure Log	Logs failure messages (PSL only) to the transcript.
Pass Log	Logs pass messages (PSL only) to the transcript. Only enabled if the simulator is invoked with the -assertdebug argument, or the AssertionDebug *.ini* file variable is set to 1.
... Action	Allows you to take specified actions when the selected assertions meet certain conditions.
Test Analysis	When a UCDB file is selected, allows you to Find Tests with: Least Coverage, Most Coverage, Zero Coverage, or Non-Zero Coverage; Rank Most Effective Tests; produce a Summary report.
XML Import Hint	Displays the XML Import Hint dialog box with information about the Link Type and Name.
Filter	Setup — Opens Filter Setup dialog Apply — Applies filter to selected item(s)
Expand	Expand or collapse the hierarchy.
Display Options	Allows you to control the appearance of information in the window.
Exclude Selected	Excludes selected item(s) from coverage statistics collection and reports
Exclude with Comment	Excludes selected item(s), with a comment, from coverage statistics collection and reports
Clear Exclusion	Clears coverage exclusion for selected item(s)

Usage Notes

Refer to Viewing Assertions in the Assertions Window in the User's Manual for more information.

ATV Window

To access:

- **Assertions > Add ATV**, when selecting an assertion in the Assertions Window.

- add atv command

Refer to "Viewing Assertion Threads in the ATV Window" in the User's Manual for detailed instructions.

Use this window to view the progress of assertion threads (PSL and SystemVerilog assertions) over time.

Description

This section describes GUI elements specific to this window.

The ATV window contains four panes (described in more detail in the User's Manual):

- Expression Pane — Shows a hierarchical representation of the assertion.

- Thread Viewer Pane — Shows the progress of the assertion threads over time for a given thread instance and starting time.

- Local Variables Pane — Shows any values related to local variables in the assertion.

- Design Objects Pane — Shows values of the design objects in the expression at that time.

The graphic symbols in the Thread Viewer Pane indicate the current state of the assertion or the cover directive.

Figure 64 shows the status information that is displayed when you hover the mouse over graphic symbols used to indicate clock, thread, and directive status.

Right-click in the window to display the popup menu and select an option.

Figure 64. ATV Window
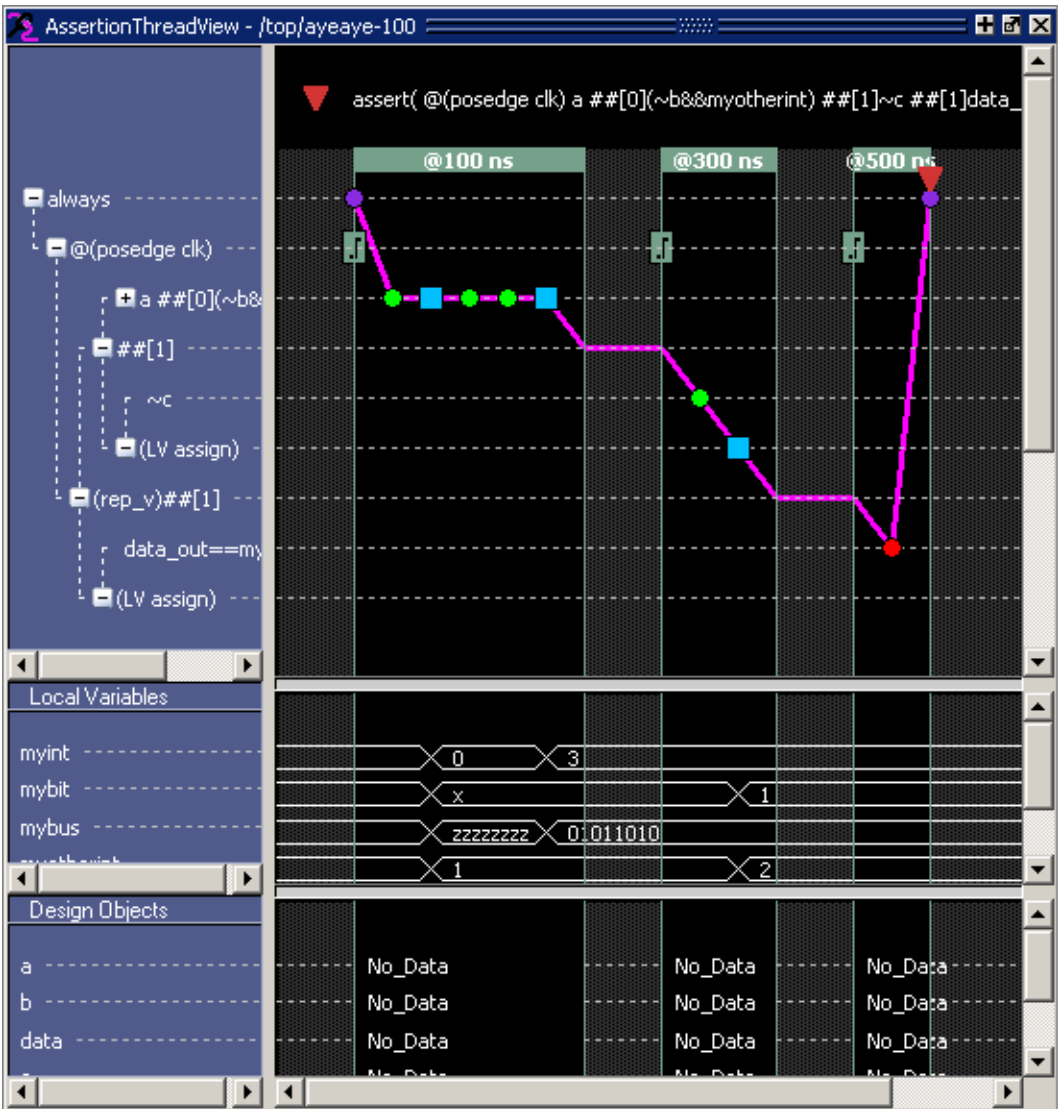
Objects

Table 59. Graphic Symbols for Current Directive State

Graphic Symbol	Status Information
■	State: Start
▶	State: Active
▶	State: Antecedent
▲	State: Passed

Table 59. Graphic Symbols for Current Directive State (continued)

Graphic Symbol	Status Information
▼	State: Failed

Table 60. Graphic Symbols for Clock, Thread, and Directive Status

Graphic Symbol	Status Information
▲	Directive Passed
▼	Directive Failed
●	Root thread started/completed
●	New evaluation thread forked
●	Boolean passed
●	Boolean failed
⏲	clock time & clock name
■	Local variable and value
▼	Thread failed but is redundant since other threads are still running
✖	Thread killed because directive was aborted or disabled
✖	Thread killed because other thread caused unilateral pass/fail
✪	Thread passed but directive waiting on other running threads

Table 61. ATV Window Popup Menu

Popup Menu Item	Description
View Source	Open the source file and highlight the assertion.
View Grid	Toggles the appearance of grid lines
Ascending Expressions	Toggles the display of the assertion into ascending or descending mode.
Annotate Local Vars	Displays local variable information in the Thread Viewer pane.
Show Local Vars	Toggles the display of the Local Variables pane.
Show Design Objects	Toggles the display of the Design Objects pane.

Table 61. ATV Window Popup Menu (continued)

Popup Menu Item	Description
View Full Object Names	Toggles the display of the object names in the Design Objects pane.
Add …	Adds the assertion to the selected window.
Zoom …[1]	Controls the zoom level of the Thread Viewer pane.

1. If you choose **Wave > Mouse Mode > Zoom Mode**, you do not need to press the Ctrl key.

Usage Notes

For more information on ATV window tasks, refer to Actions in the ATV Window in the User's Manual.

Call Stack Window

To access:

- **View > Call Stack**

The Call Stack window displays the current call stack when you single step the simulation, the simulation has encountered a breakpoint, or when you select any process in either the Structure or Processes windows.

Description

When debugging your design you can use the call stack data to analyze the depth of function calls that led up to the current point of the simulation, which include:

- Verilog functions and tasks

- VHDL functions and procedures

- SystemC methods and threads

- C/C++ functions

The Call Stack window also supports C Debug mode. (Refer to C Debug in the User's Manual.)

Figure 65. Call Stack Window
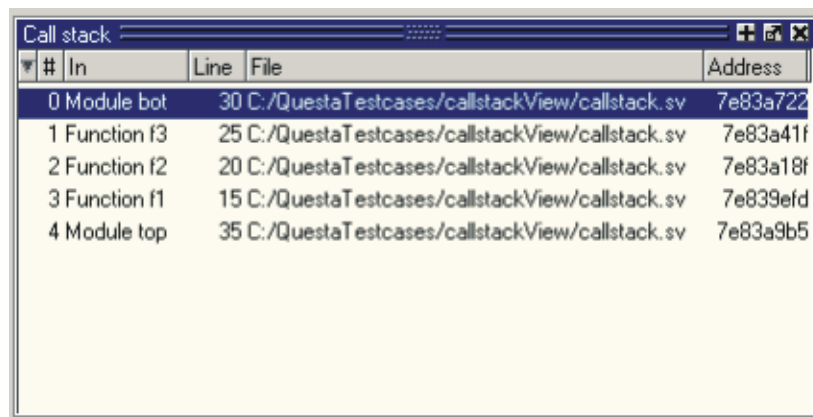
Objects

Table 62. Call Stack Window Columns

Column Title	Description
#	indicates the depth of the function call, with the most recent at the top.

Table 62. Call Stack Window Columns (continued)

Column Title	Description
In	indicates the function. If you see "unknown" in this column, you have most likely optimized the design such that the information is not available during the simulation.
Line	indicates the line number containing the function call.
File	indicates the location of the file containing the function call.
Address	indicates the address of the execution in a foreign subprogram, such as C.

Usage Notes

Call Stack Window Tasks

This window allows you to perform the following actions:

- Double-click the line of any function call:

  ◦ Displays the local variables at that level in the Locals Window.

  ◦ Displays the corresponding source code in the Source Window.

Related Commands of the Call Stack Window

Table 63. Commands Related to the Call Stack Window

Command Name	Description
stack down	This command moves down the call stack.
stack frame	This command selects the specified call frame.
stack level	This command reports the current call frame number.
stack tb	This command is an alias for the tb command.
stack up	This command moves up the call stack.

Capacity-Object and Capacity-Line Windows

To access:

- **View > Object-based > Capacity** or **View > Line-based Capacity**

- view capacity command, or view capacity line command

The Capacity-Object window shows memory use based on objects, and the Capacity-Line window shows memory usage based on point of allocation. You can have both windows open at the same time to debug memory issues.

Description

When your design contains Verilog design units you will see the following entries in the Type/Object column

- Always

- Always blocks

- **Assertions** — When using fine-grained analysis (vsim -capacity) this entry expands and provides information for each assertion.

- Blocks

- Classes — When using fine-grained analysis (vsim -capacity) this entry expands and provides information for each class.

- Continuous assignment

- Covergroups

- Function instances

- Initial

- Initial blocks

- Module instances

- Nets

- Parameters

- **QDAs** — When using fine-grained analysis (vsim -capacity) this entry expands and provides information for Queues, Dynamic Arrays, and Associative Arrays.
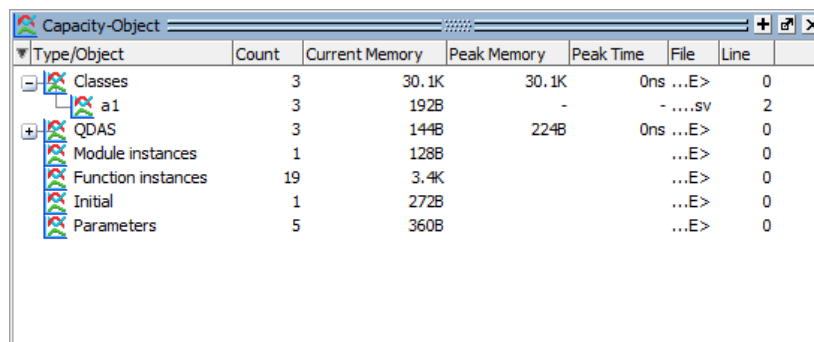
- Registers

- Solver

- System task instances

- Task instances

- Verilog Memories

- Verilog Ports

When your design contains VHDL design units you will see the following entries in the Type/Object column

- Instances

- Ports

- Signals

- Processes
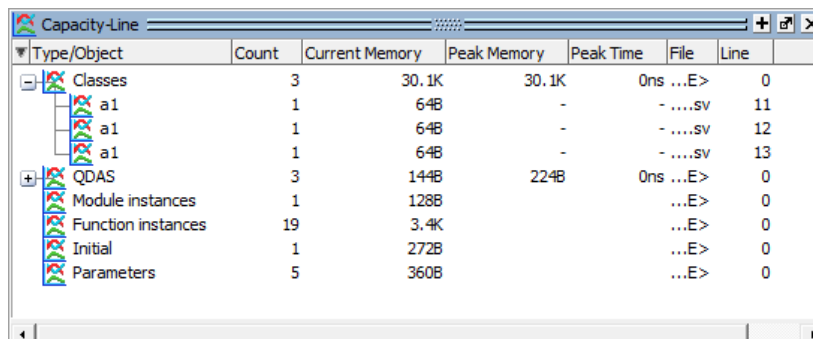
Refer to Capacity Analysis in the User's Manual for more information.

Figure 66. Capacity-Object Window

Type/Object	Count	Current Memory	Peak Memory	Peak Time	File	Line	
Classes	3	30.1K	30.1K	0ns	...E>	0	
a1	3	192B	-	-	....sv	2	
QDAS	3	144B	224B	0ns	...E>	0	
Module instances	1	128B			...E>	0	
Function instances	19	3.4K			...E>	0	
Initial	1	272B			...E>	0	
Parameters	5	360B			...E>	0	

Figure 67. Capacity-Line Window

Objects

Table 64. Capacity Window Columns

Column Title	Description
Type/Object	Refer to the Descriptions above.
Count	Quantity of design objects analyzed
Current Memory	Current amount of memory allocated, in bytes
Peak Memory	Peak amount of memory allocated, in bytes
Peak Time	The time, in ns, at which the peak memory was reached

Class Graph Window

To access:

- **View > Class Browser > Class Graph**

- view classgraph command

The Class Graph window provides a graphical view of your SystemVerilog classes, including any extensions of other classes and related methods and properties.

Description

Figure 68. Class Graph Window
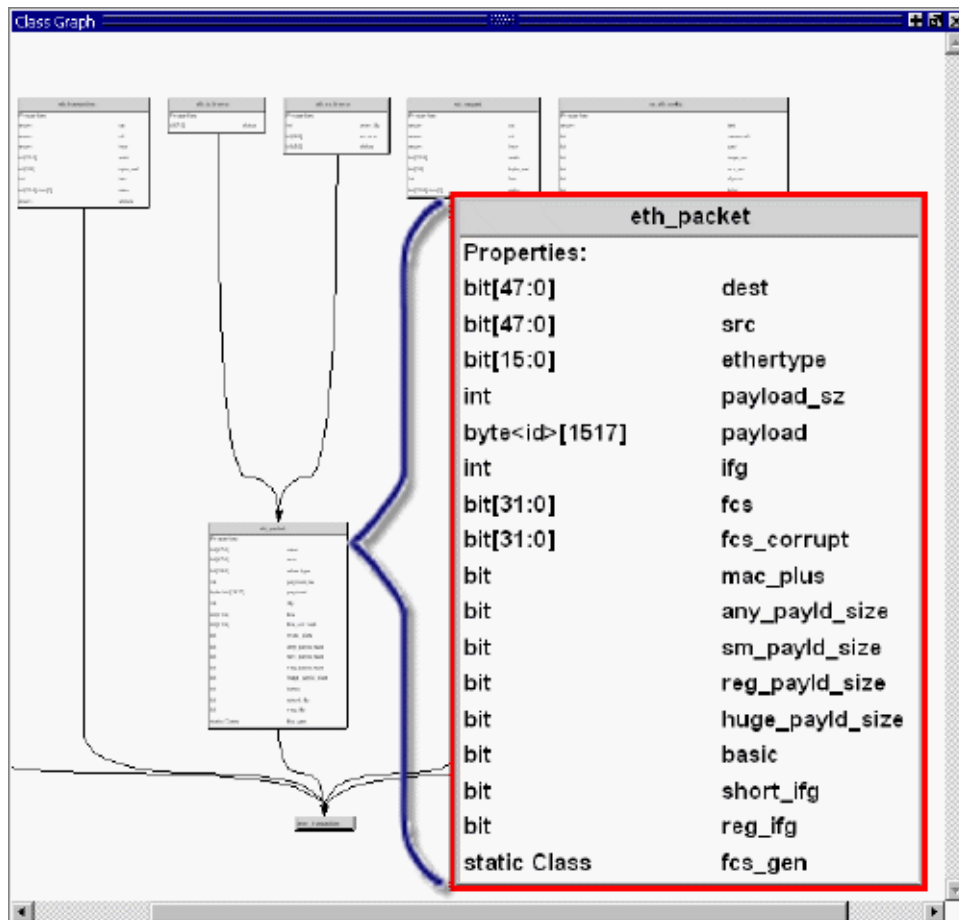
Objects

Table 65. Class Graph Window Popup Menu

Popup Menu Item	Description
Filter	Controls the display of methods and properties from the class boxes.
Zoom Full	Zooms the window to view all information
View Entire Design	Reloads the view to show the class hierarchy of the complete design.
Print to Postscript	Opens the Print Postscript dialog box for you to save the information to a .ps file.
Organize by Base/Extended Class	Reorganizes the window so that the base or extended (default) classes are at the top of the hierarchy.

Usage Notes

Navigating in the Class Graph Window

You can change the view of the Class Graph window with your mouse or the arrow keys on your keyboard.

- **Left click-drag** — Allows you to move the contents around in the window.

- **Middle Mouse scroll** — Zooms in and out.

- Middle mouse button strokes:

  - **Upper left** — Zoom full

  - **Upper right** — Zoom out. The length of the stroke changes the zoom factor.

  - **Lower right** — Zoom area.

- **Arrow Keys** — Scrolls the window in the specified direction.

  - **Unmodified** — Scrolls by a small amount.

  - **Ctrl+<arrow key>** — Scrolls by a larger amount

  - **Shift+<arrow key>** — Shifts the view to the edge of the display

Class Instances Window

To access:

- **View > Class Browser > Class Instances**

- view classinstances command

The Class Instances window shows the list of class instances and their values for a particular selected class type. You may add the class items directly to the wave or list windows, log them, or get other information about them.

Description

Figure 69. Class Instances Window
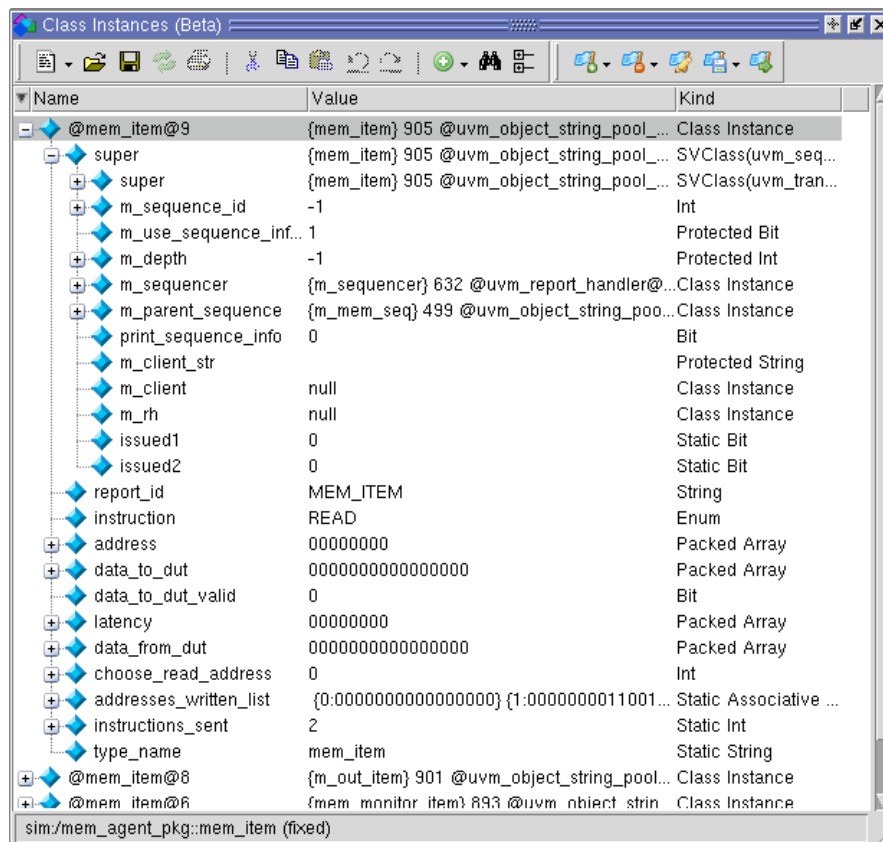
Objects

Table 66. Class Instances Window Popup Menu

Popup Menu Item	Description
View Declaration	Highlights the line of code where the type of the instance is declared, opening the source file if necessary.

Table 66. Class Instances Window Popup Menu (continued)

Popup Menu Item	Description
Add Wave	Adds the selected class instance to the Wave window.
Add to	Allows you to log the selected class instance, or add it to the Wave or List windows.

Usage Notes

Viewing Class Instances

The Class Instances window is dynamically populated by selecting SVClasses in the Structure (sim) window. All currently active instances of the selected class are displayed in the Class Instances window. Class instances that have not yet come into existence or have been destroyed are not displayed.

Once you have chosen the class type you want to observe, you can fix that instance in the window while you debug by choosing **File > Environment > Fix to Current Context**

Class Naming Format

Class instance names are formatted as follows: @<class_type>@<nnn> where @<class_type>@ is the name of the class type and <n> is the reference identifier for a particular instance of the class type. For example, @uvm_queue_3@14 is the 14th instance of the class uvm_queue_3.

Class Tree Window

To access:

- **View > Class Browser > Class Tree**

- view classtree command

The Class Tree window provides a hierarchical view of your SystemVerilog classes, including any extensions of other classes, related methods and properties, as well as any covergroups.

Description

Figure 70. Class Tree Window
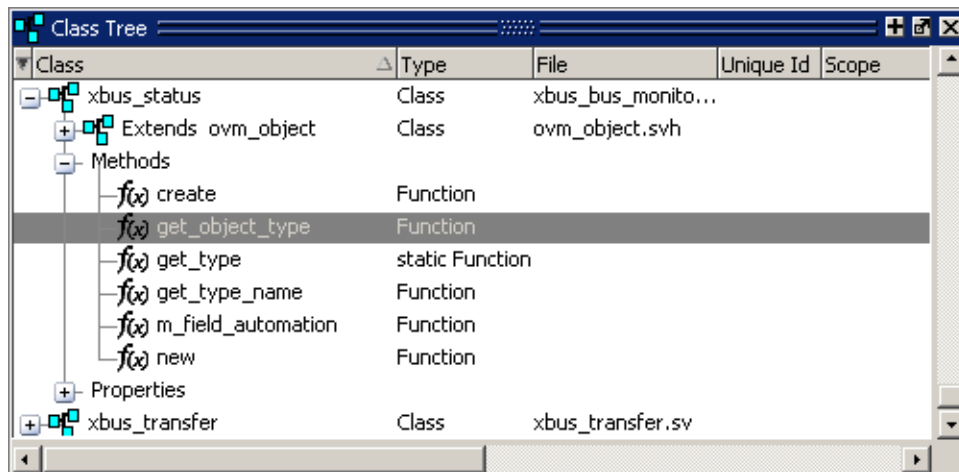
Objects

Table 67. Class Tree Window Icons

Icon	Description
	Class
	Parameterized Class
	Function
	Task
	Variable
	Virtual Interface

Table 67. Class Tree Window Icons (continued)

Icon	Description
	Covergroup
	Structure

Table 68. Class Tree Window Columns

Column	Description
Class	The name of the item
Type	The type of item
File	The source location of the item
Descriptive Name	The internal name of the parameterized class (only available with parameterized classes)
Scope	The scope of the covergroup (only available with covergroups

Table 69. Class Tree Window Popup Menu

Popup Menu Item	Description
View Declaration	Highlights the line of code where the item is declared, opening the source file if necessary.
View as Graph	Displays the class and any dependent classes in the Class Graph window (only available for classes).
Filter	allows you to filter out methods and or properties
Organize by Base/Extended Class	reorganizes the window so that the base or extended (default) classes are at the top of the hierarchy.

Code Coverage Analysis Window

To access:

- **View > Coverage > Code Coverage Analysis** select the desired analysis type from the Code Coverage Analysis window's title bar, detailed in Table 70.

- view analysis command

Use this window to view covered (executed), uncovered (missed), and/or excluded statements, branches, conditions, expressions, FSM states and transitions, as well as signals that have and have not toggled.

Description

This window is specific to the collection of coverage metrics, therefore you must have run your simulation with coverage collection enabled. Refer to Code Coverage in the User's Manual for more information.

Figure 71. Code Coverage Analysis
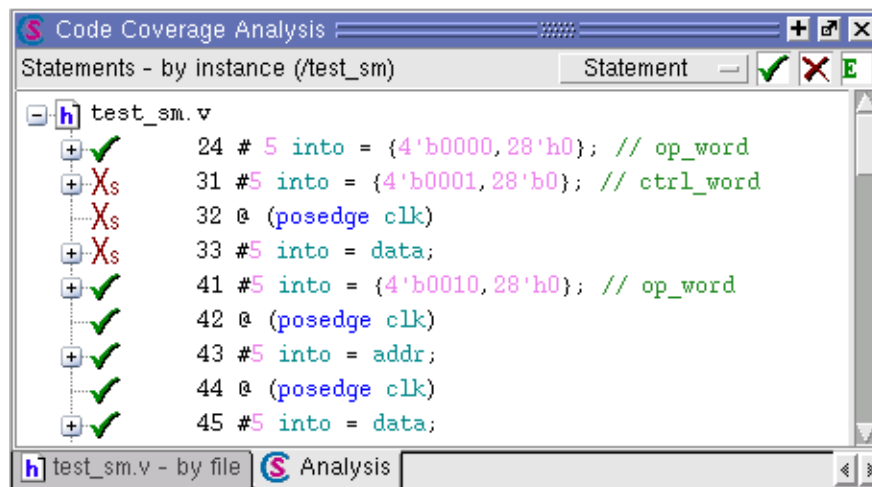
Code Coverage Analysis Window Toolbar

This section describes the GUI elements specific to the Code Coverage Analysis window. Primary of these are found in the toolbar. By default, the buttons listed in Table 70 are selected (active).

Objects

Table 70. Contents of Code Coverage Analysis Title Bar
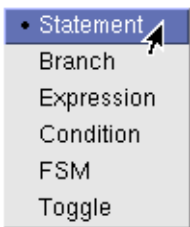
Selection/Button	Action
• Statement Branch Expression Condition FSM Toggle	**Analysis Type** button: Specifies the type of coverage analysis currently selected in the sub-window inside the Code Coverage Analysis window. Six selections are available when you click the type, as shown in the image to the left.
✓	Covered items button: When selected (default, as shown in image), all covered (hit) items are displayed in the window. When not selected, all items are filtered from view.
✗	Missed items button: When selected (default), all missed items (not executed) are displayed in the window. When not selected, all missed items are filtered from view.
E	Excluded items button: When selected (default), all excluded items are displayed in window. When not selected, excluded items are filtered from view.

Table 71. Code Coverage Analysis Window Popup Menu

Popup Menu Item	Description
Exclude Selection	excludes the selected lines of code
Exclude with Comment...	excludes selected lines of code with a comment.
Edit Comment...	allows you to edit the comment
Show Details	Adds Coverage Details information to the tree structure.
Show Structure	Adds Structure window information to the tree structure.
Test Analysis	Executes coverage analyze on the selected item(s) and prints information to a Test Analysis window. • Find Least Coverage • Find Most Coverage • Find Zero Coverage • Find Non Zero Coverage • Summary
Copy	copies selected item(s) to clipboard
Expand All	expands all items displayed in window

Table 71. Code Coverage Analysis Window Popup Menu (continued)

Popup Menu Item	Description
Collapse All	collapses all items displayed in window

Viewing Code Coverage Data and Current Exclusions

You can view executed or missed statements, branches, conditions, expressions, or FSMs, as well as items excluded from coverage.

Procedure

1. Select a file in the Files window, or an instance or design unit in the Structure window whose coverage you wish to analyze.

2. With the Code Coverage Analysis window active. select the type of coverage to view (Branch Analysis, Condition analysis, and so on) from the pulldown menu in the Analysis toolbar (Figure 71).

Figure 72. Missed Coverage in Code Coverage Analysis Windows
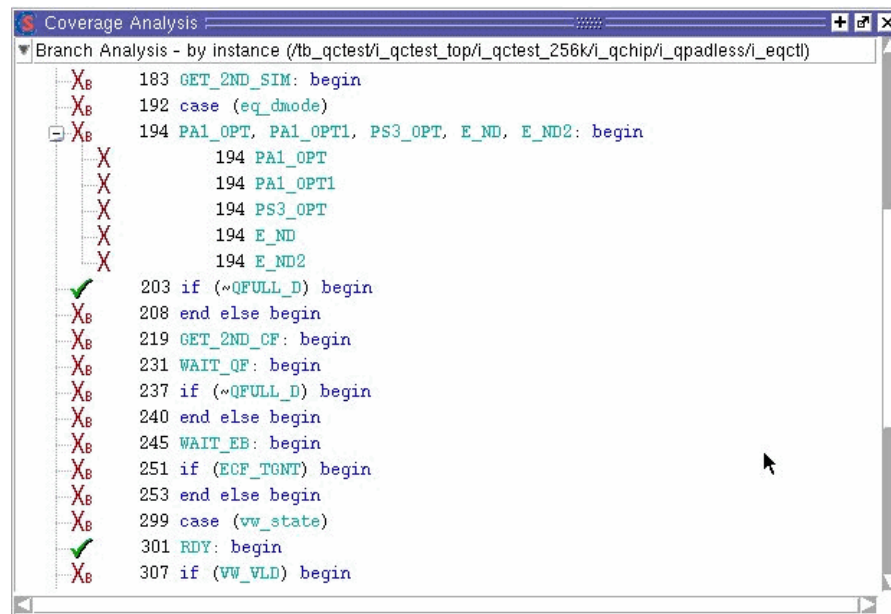
3. Each coverage type window includes a column for the line number and a column for statement, branch, condition, expression, or toggle on that line. An icon indicates whether the object was executed (green check mark), not executed (red X), or excluded (green E). See Table 110 for a complete list of icons.

Results

In the banner for all Coverage Analysis window types, the following information appears:

- Name of the window

- Whether the coverage is **by file** or **by instance** (depending on whether a file was selected in the **Files** tab or an instance or du from the **sim** tab)

- Scope of the coverage item (in parentheses) being displayed

- Analysis Type button, Covered Items button, Missed Coverage buttons, Excluded Items button (see Table 70)

You can change the scope displayed (for all Code Coverage Analysis windows) by selecting a new scope in the Structure or Files windows.

When you select (left-click) any item in the Statement, Branch, Condition, Expression, FSM or Toggle Analysis windows, the Coverage Details Window populates with related details (coverage statistic details, truth tables, exclusions and so on) about that object. In the case of a multi-line statement, branch, condition or expression, select the object on the last line of the item.

The Branch Analysis window includes a column for branch code (conditional "if/then/else" and "case" statements). "XT" indicates that the true condition of the branch was not executed. "XF" indicates that the false condition of the branch was not executed. Fractional numbers indicate how many case statement labels were executed.

When you right-click any item in the window, a pop-up menu appears with options to control the addition or removal of coverage exclusions. The options and their function is identical to that of the Source Window. Refer to Methods for Excluding Objects in the User's Manual for a description of adding comments with exclusions.

Note:
Multi-line objects are rooted in the last line, and exclusions must be applied on that line # in order to take effect.

Cover Directives Window

To access:

- **View > Coverage > Cover Directives**

- **view coverdirectives** command

The Cover Directives window displays a list of SystemVerilog and PSL, embedded and external, cover directives that were successfully compiled and simulated during the current simulation.

Description

Figure 73. Cover Directives Window
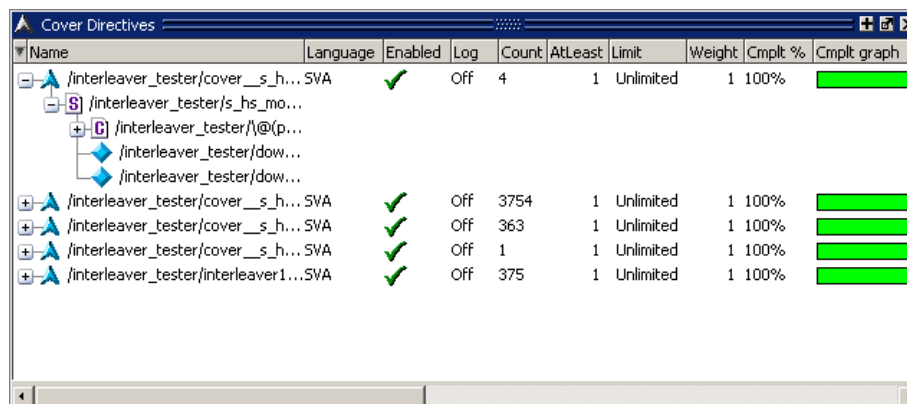
Objects

Table 72. Cover Directives Window Columns

Column	Description
AtLeast	number of times a directive has to fire to be considered 100% covered.
Cmplt %	coverage percentage for a directive. The percentage is the lesser of 100% or Count divided by AtLeast.
Cmplt graph	graphical bar chart of the completion percentage. Directives with 100% coverage are displayed in green.
Count	number of times a directive has "fired" during the current simulation.
Design Unit	design unit to which the directive is bound.
Design Unit Type	HDL type of the design unit. Not displayed by default.
Enabled	displays a green checkmark when a directive is enabled or a red X when a directive is disabled.

Table 72. Cover Directives Window Columns (continued)

Column	Description
Included	indicates whether the directive is included in aggregate statistics and reports.
Language	identifies the HDL used to write the assertion.
Limit	number of times the directive will execute before being disabled by the simulator. Default is Unlimited.
Log	indicates whether data for the directive is currently being added to the functional coverage database.
Memory	tracks the current memory used by the cover directive.
Name	lists directive names and design units. Also, any signals referenced in a directive are included in the hierarchy. Refer to Using Assert Directive Names in the User's Manual for more information.
Peak Memory	tracks the peak memory used by the cover directive.
Peak Memory Time	indicates the simulation run time at which the peak memory usage occurred.
Type	shows the cover directive type (Immediate or Concurrent). Not displayed by default.
Weight	shows the weighting factor that has been applied to the directive.

Usage Notes

Changing the Cover Directives Window Display Options

You can set the window to display cover directives in a **Recursive Mode** or in a **Show All Contexts** mode.

- The **Recursive Mode —** Displays all cover directives at and below the selected hierarchy instance, the selection being taken from a Structure window. (that is, the **sim** tab). Otherwise only items actually in that particular scope are shown.

- The **Show All Contexts** — Selection displays all instances in the design. It does not following the current context selection in a structure pane. The Show All Context display mode implies the recursive display mode as well, so the **Recursive Mode** selection is automatically grayed out.

You can choose between these two display modes by right-clicking in the Cover Directives window and selecting the option from the **Display Options** sub-menu.

Coverage Details Window

To access:

- **View > Coverage > Details**

- view details command

You can populate this window by selecting an item in one of the panes of the Code Coverage Analysis window: either Statement, Branch, Expression, Condition, FSM or Toggle. Use this window to view detailed results about coverage metrics from your simulation.

> **Note:**
> This window is specific to the collection of coverage metrics, therefore you must have run your simulation with coverage collection enabled. Refer to Code Coverage in the User's Manual for more information.

Description

Figure 74. Coverage Details Window Showing Expression Truth Table
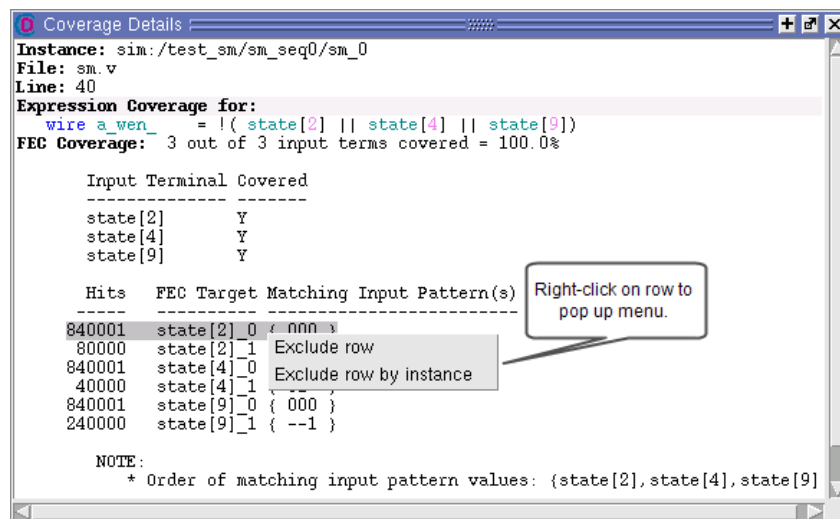
If a line number contains multiple statements, the coverage details window contains the metrics for each statement.

Toggle details are displayed as follows:

Figure 75. Coverage Details Window Showing Toggle Details
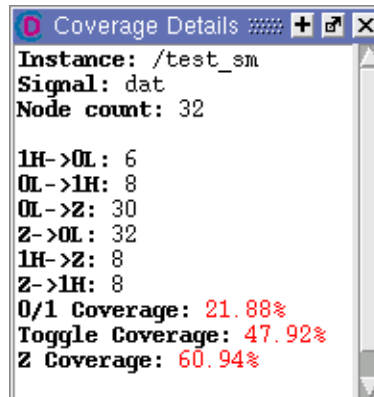
FSM details are displayed as shown in Figure 76:

Figure 76. Coverage Details Window Showing FSM Details
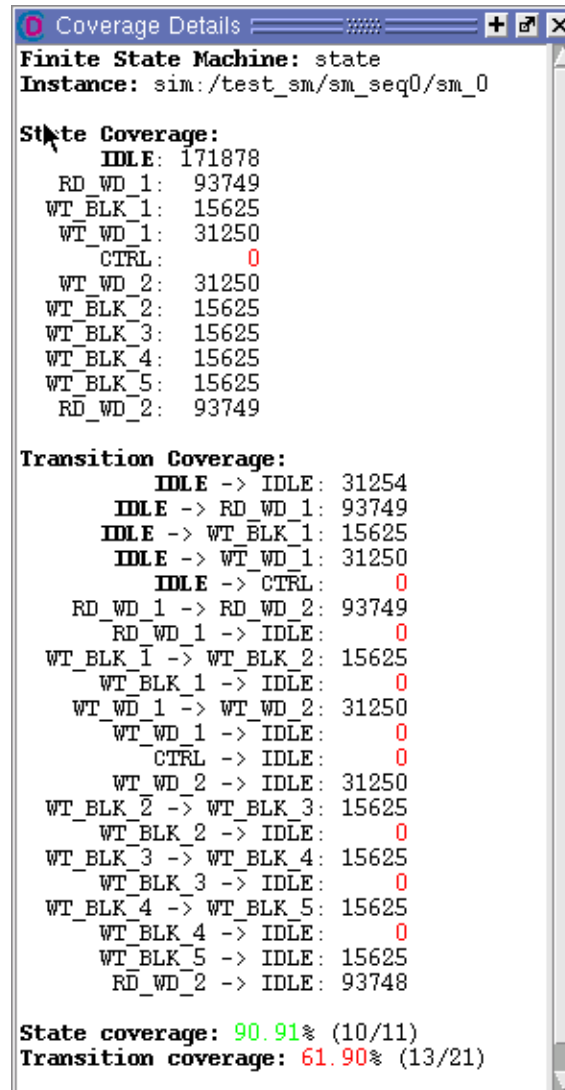
Objects

Coverage Details of Statement Coverage Fields

Column	Description
Instance	The dataset name followed by the hierarchical location of the statement. Only appears when you are analyzing coverage metrics by instance.
File	The name of the file containing the statement.
Line	The line number of the statement. In the case of a multi-line statement, this is the last line of the statement.
Statement Coverage for	Name of the statement itself.

Column	Description
Hits	The number of times the statement was hit during the simulation.

Coverage Details of Branch Coverage Fields

Column	Description
Instance	The dataset name followed by the hierarchical location of the branch. Only appears when you are analyzing coverage metrics by instance.
File	The name of the file containing the statement.
Line	The line number of the statement. In the case of a multi-line branch statement, this is the last line of the statement.
Branch Coverage for	The statement itself. • **Active** —The number of times the branch has been executed. • **True Hits** — The number of times the branch resolved to True.

Coverage Details of Condition and Expression Coverage Fields

Column	Description
Instance	The dataset name followed by the hierarchical location of the condition. Only appears when you are analyzing coverage metrics by instance.
File	The filename containing the condition.
Line	The line number of the filename containing the condition or expression. In the case of a multi-line condition statement, this is the last line of the statement.
Condition/Expression Coverage for	The syntax of the condition.
FEC Coverage	A tabular representation of the focused expression coverage metrics to satisfy the condition. Refer to FEC Report Examples in the User's Manual for more information about FEC condition/expression coverage. You can exclude rows or rows by instance through a popup menu accessible by right-clicking on a row in the table (see Figure 74).

Coverage Details of Toggle Coverage Fields

Column	Description
Instance	The dataset name followed by the hierarchical location of the signal. Only appears when you are analyzing coverage metrics by instance.
Signal	The name of the signal (data[6]) or (data).

Column	Description
Node Count	The size of the signal.
Toggle List	The list of toggles analyzed during simulation. This list will differ depending on whether you specified extended toggle coverage. Refer to Standard and Extended Toggle Coverage in the User's Manual for more information. • Toggle coverage shows toggle metrics between 0 and 1 • Extended toggle coverage shows toggle metrics between 0, 1 and Z.
Toggle Coverage	The percentage of nodes that were covered.
0/1 Coverage	The percentage of standard toggles that were covered.
Full Coverage	The percentage of extended toggles that were covered.
Z Coverage	The percentage of toggles involving Z that were covered.

Coverage Details of FSM Coverage Fields

Fields	Description
Finite State Machine	The name of the finite state machine
Instance	The dataset name followed by the hierarchical location of the FSM. Only appears when you are analyzing coverage metrics by instance.
State Coverage	A list of all the states, followed by the number of hits.
Transition Coverage	A list of all the transitions, followed by the number hits.
State Coverage	The coverage percentage for the states.
Transition Coverage	The coverage percentage for the transitions.

Covergroups Window

To access:

- **View > Coverage > Covergroups**

- view covergroups command

The Covergroups window displays SystemVerilog covergroups (cvg), coverpoints (cvp), crosses (cross) and bins (bin) in the current region (which is selected via the Structure window).

Note:
Refer to Functional Coverage Statistics in the GUI in the User's Manual for more information.

Description

Item names in black indicate items that contribute to coverage statistics. Item names in gray indicate non-contributing items — which are either excluded, empty, or have zero weight and do not contribute any coverage statistics to their parent items.

Figure 77. Covergroups Window

Objects

Table 73. Covergroups Window Columns

Column	Description
% Hit	the percentage of the total covergroup bins which have been hit (Total Hits divided by Covered Bins)
% of Goal	the percentage of the coverage goal that has been reached.
% over Goal	the percentage over the coverage goal that has been reached
Auto_bin_max	the maximum number of automatically created bins when no bins are explicitly defined for coverpoints.
BinRHS	the RHS value of a bin, specifically a string that describes the sampled values that could cause the particular bin to increment.
Class Type	lists the parameterized class type name of the corresponding classes in the window.
Comment	displays comments that appear with the instance of a covergroup, or a coverpoint or cross of the covergroup instance.
Cover Test	in post processing mode (Coverage View) it displays the name of the test which covered the bin. During live simulation, the '<current_test>' is displayed.
Cover Time	displays the time when the bin was covered.
Coverage	weighted average of the coverage of the constituent coverpoints and crosses.
Covered Bins (Hits)	lists the # of covered bins (hits) for covergroup, coverpoint, cross, and covergroup instance objects.
Cross_num_print_missing	the number of missing (not covered) cross product bins that must be saved to the coverage database and printed in the coverage report.
Detect_overlap	shows when a warning has been issued for an overlap between the range list (or transition list) of two bins of a coverpoint.
Get_inst_coverage	the value of the get_inst_coverage covergroup option. • When true, it enables the tracking of per instance coverage with the get_inst_coverage built-in method. • When false, the value returned by get_inst_coverage shall equal the value returned by get_coverage.
Goal	the desired coverage total as an integer percent.
Included	indicates whether the Covergroup is included in (check mark) or excluded from (X mark) aggregate statistics and reports.

Table 73. Covergroups Window Columns (continued)

Column	Description
Merge_instances	the value of the merge_instances covergroup type option. • When true, cumulative (or type) coverage is computed by merging instances together as the union of coverage of all instances. • When false, type coverage is computed as the weighted average of instances. • When "auto(1)" or "auto(0)" is displayed, this indicates a default value chosen by tool with the effective value inside the parenthesis.
Missing Bins	the number of covergroup bins missing coverage
Name	the name of the covergroup and its components.
Peak Transient Memory	the peak transient memory used by the covergroup.
Peak Transient Memory Time	the simulation run time at which the peak transient memory usage occurred.
Persistent Memory	the persistent memory used by the covergroup.
Real_interval	for analog coverpoints, which you can set to specify the number of unique value ranges in a real range. Mandatory for real coverpoint declarations. For example, if real_interval is 0.2, then the range [1.0:1.6] will have three unique value ranges: [1.0:1.2], [1.2:1.4], [1.4:1.6].
Samples	the sample count of covergroups TYPEs and covergroup instances.
Status	graphical representation of the Coverage column.
Strobe	the value of the strobe coverage group type option. If set to 1, all samples happen at the end of the time slot, like the $strobe system task.
Total Bins	displays the total # of bins for covergroup, coverpoint, cross, and covergroup instance objects, not including illegal, ignore, or default bins.
Transient Memory	the transient memory used by the covergroup.
Weight	the weighting of a covergroup instance for computing the overall instance coverage simulation. For coverpoints or crosses, it shows the weighting of a coverpoint or cross for computing the instance coverage of the enclosing covergroup.
Weighted Missing Bins	the number of weighted bins missing coverage.

Table 74. Covergroup Window Popup Menu

Popup Menu Item	Description
View Source	opens the selected file in a Source window
Report	creates a functional coverage report (in text or XML) of selected items
Hide Covergroup Instances	hides from the display: • All covergroup instances • only covergroups with per_instance set to 0
Use CrossPrintMissing	uses the value of option.cross_num_print_missing while displaying bins of covergroup cross scope
Show Zero Weight Objects	displays all objects with zero weighting
Test Analysis	executes coverage analyze on the selected instance and prints information to the Test Analysis window. • Find Least Coverage • Find Most Coverage • Find Zero Coverage • Find Non Zero Coverage • Summary
XML Import Hint	displays the XML Import Hint dialog box containing information about the Link Type and Link Name
Filter	setup — Opens the Filter Setup dialog Apply — Applies filter to selected item(s)
Expand	Allows you to expand or collapse information trees.
Display Options	displays covergroups recursively, or in all contexts
Exclude Selected	excludes selected item(s) from coverage statistics collection and reports
Exclude with Comment	excludes selected item(s), with a comment, from coverage statistics collection and reports
Clear Exclusion	clears coverage exclusion for selected item(s)

Usage Notes

Changing the Covergroups Window Display Options

> **Note:**
> Covergroups are created dynamically during simulation. This means they will not display in the GUI until you run the simulation.

You can set the window to display covergroups in a Recursive Mode or in a Show All Contexts mode.

- **Recursive Mode** — Displays all covergroups at and below the selected hierarchy instance, the selection being taken from a Structure window. (that is, the **sim** tab). Otherwise only items actually in that particular scope are shown.

- **Show All Contexts** — Selection displays all instances in the design. It does not follow the current context selection in the Structure window. The Show All Context display mode implies the recursive display mode as well, so the **Recursive Mode** selection is automatically grayed out.

You can choose between these two display modes by right-clicking in the Covergroups window and selecting the option from the **Display Options** sub-menu.

Dataflow Window

To access:

-   **View > Dataflow**

-   view dataflow command

Use this window to explore the "physical" connectivity of your design. You can also use it to trace events that propagate through the design; and to identify the cause of unexpected outputs.

Description

The Dataflow window displays:

-   processes

-   signals, nets, and registers

-   interconnects

The window has built-in mappings for all Verilog primitive gates (such as AND, OR, PMOS, NMOS). For components other than Verilog primitives, you can define a mapping between processes and built-in symbols. Refer to Symbol Mapping in the User's Manual for details.

Note:
You cannot view SystemC objects in the Dataflow window.

Figure 78. Dataflow Window
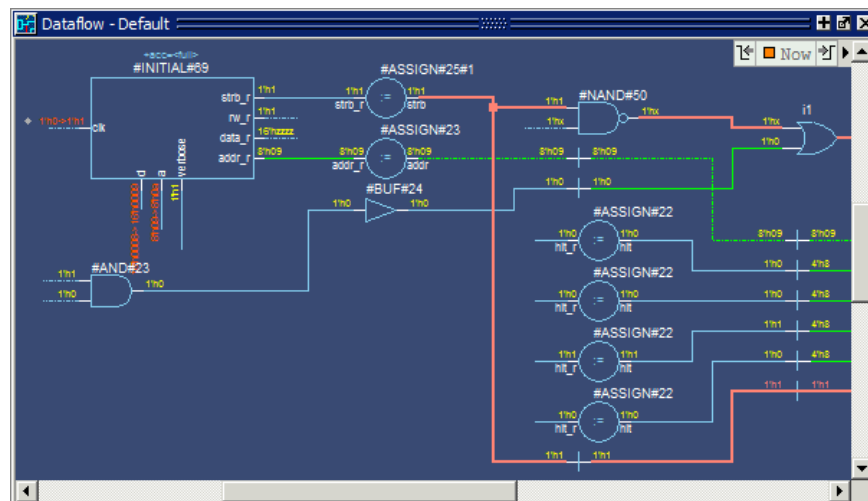
Objects

-   Interactive elements.

- ◦ Dataflow view

Dataflow Window Tasks

This section describes tasks for using the Dataflow window.

- **Select Mode** — your left mouse button is used for selecting objects and your middle mouse button is used for zooming the window. This is the default mode.

- **Zoom Mode** — your left mouse button is used for zooming the window and your middle mouse button is used for panning the window.

- **Pan Mode** — your left mouse button is used for panning the window and your middle mouse button is used for zooming the window.

Selecting Objects in the Dataflow Window
Zooming the View of the Dataflow Window
Panning the View of the Dataflow Window
Displaying the Wave Viewer Pane

Selecting Objects in the Dataflow Window

When you select an object, or objects, it will be highlighted an orange color.

Procedure

Perform the following selection tasks as needed:

If you want to...	Do the following:
Select a single object	Single click
Select multiple objects	Shift-click on all objects you want to select or click and drag around all objects in a defined area. Only available in Select Mode.

Zooming the View of the Dataflow Window

Several zoom controls are available for changing the view of the Dataflow window, including mouse strokes, toolbar icons and a mouse scroll wheel.

Procedure

You can zoom the view of the Dataflow window using one of the following methods:

If you want to...	Do the following...
Zoom Full — Fills the Dataflow window with all visible data	

If you want to...	Do the following...
Mouse stroke	Up/Left. Middle mouse button in Select and Pan mode, Left mouse button in Zoom mode.
Menu	**Dataflow > Zoom Full**
Zoom Toolbar	Zoom Full
Zoom Out	
Mouse stroke	Up/Right. Middle mouse button in Select and Pan mode, Left mouse button in Zoom mode.
Menu	**Dataflow > Zoom Out**
Zoom Toolbar	Zoom Out
Mouse Scroll	Push forward on the scroll wheel.
Zoom In	
Menu	**DataFlow > Zoom In**
Zoom Toolbar	Zoom In
Mouse Scroll	Pull back on the scroll wheel.
Zoom Area — Fills the Dataflow window with the data within the bounding box.	
Mouse stroke	Down/Right
Mouse stroke	Down/Left

Panning the View of the Dataflow Window

You can pan the view of the Dataflow window with the mouse or keyboard.

Procedure

Pan the view of the Dataflow window using one of the following methods:

If you want to...	Do the following...
Pan with the Mouse	In Zoom mode, pan with the middle mouse button. In Pan mode, pan with the left mouse button. In Select mode, pan with the Ctrl key and the middle mouse button.
Pan with the Keyboard	Use the arrow keys to pan the view. Shift+<arrow key> pans to the far edge of the view. Ctrl+<arrow key> pans by a moderate amount.

Displaying the Wave Viewer Pane

You can embed a miniature wave viewer in the Dataflow window.

Procedure

1. Choose the **Dataflow > Show Wave** menu item.

2. Select a process in the Dataflow pane to populate the Wave pane with signal information.

3. Refer to Exploring Designs with the Embedded Wave Viewer in the User's Manual for more information.

Figure 79. Dataflow Window and Panes
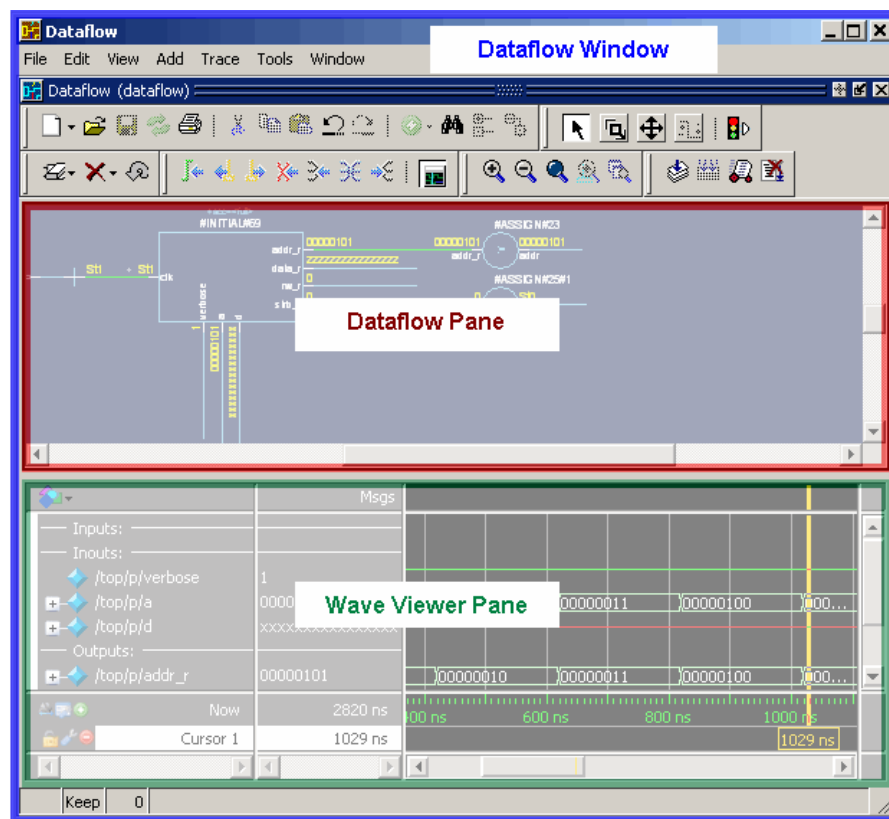
Files Window

To access:

- **View > Files**

- view files command

Use this window to display the source files and their locations for the loaded simulation.

📄 **Note:**
You must have executed the vsim command before this window will contain any information about your simulation environment.

Description

Figure 80. Files Window
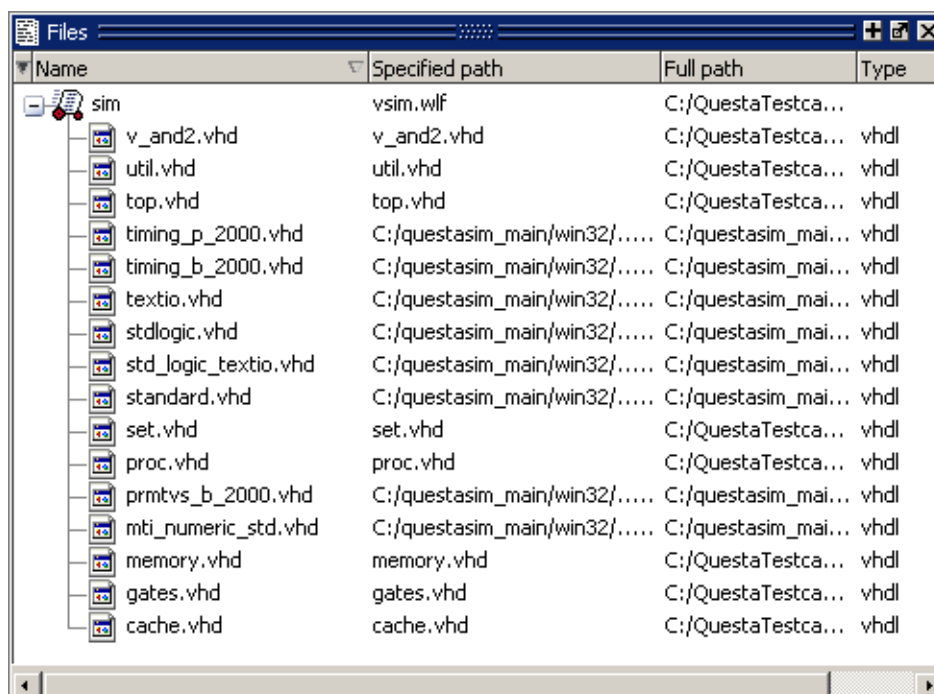
The Files menu becomes available in the **Main** menu when the Files window is active.

Objects

Table 75. Files Window Columns

Column Title	Description
Name	The name of the file

Table 75. Files Window Columns (continued)

Column Title	Description
Specified Path	The location of the file as specified in the design files.
Full Path	The full-path location of the design files.
Type	The file type.
Branch info	A series of columns reporting branch coverage
Condition info	A series of columns reporting condition coverage
Expression info	A series of columns reporting expression coverage
FEC condition info	A series of columns reporting condition coverage based on Focused Expression Coverage
FEC expression info	A series of columns reporting expression coverage based on Focused Expression Coverage
States info	A series of columns reporting finite state machine coverage
Statement info	A series of columns reporting statement coverage
Toggles info	A series of columns reporting toggle coverage
Transition info	A series of columns reporting transition coverage

Table 76. Files Window Popup Menu

Menu Item	Description
View Source	Opens the selected file in a Source window
Open in external editor	Opens the selected file in an external editor. Only available if you have set the Editor preference: • set PrefMain(Editor) {<path_to_executable>} • **Tools > Edit Preferences**; **by Name** tab, **Main** group.
Code Coverage >	These menu items are only available if you ran the simulation with the -coverage switch. • Code Coverage Reports — Opens the Coverage Text Report dialog box, allowing you to create a coverage report for the selected file. • Exclude Selected File — Executes the coverage exclude command for the selected file(s). • Clear Code Coverage Data — Clears all code coverage information collected during simulation

Table 76. Files Window Popup Menu (continued)

Menu Item	Description
Properties	Displays the File Properties dialog box, containing information about the selected file.

Table 77. Files Menu

Files Menu Item	Description
View Source	Opens the selected file in a Source window
Open in external editor	Opens the selected file in an external editor. Only available if you have set the Editor preference: • set PrefMain(Editor) {<path_to_executable>} • **Tools > Edit Preferences**; **by Name** tab, **Main** group.
Save Files	Saves a text file containing a sorted list of unique files, one per line. The default name is *summary.txt*.

FSM List Window

To access:

- **View > FSMList**

- view fsmlist command

Use this window to view a list of finite state machines in your design.

Description

This window is populated when you specify any of the following switches during compilation (vcom/vlog) or optimization (vopt).

- +cover or +cover=f

- +acc or +acc=f

Figure 81. FSM List Window

The FSM List menu becomes available in the **Main** menu when the FSM List window is active.

Objects

Table 78. FSM List Window Columns

Column Title	Description
Instance	Lists the FSM instances. You can reduce the number of path elements in this column by selecting the **FSM List > Options** menu item and altering the Number of Path Elements selection box.
States	The number of states in the FSM.
Transitions	The number of transitions in the FSM.

Table 79. FSM List Window Popup Menu

Popup Menu Item	Description
View FSM	Opens the FSM in the FSM Viewer window.
View Declaration	Opens the source file for the FSM instance.
Set Context	Changes the context to the FSM instance.
Add to <window>	Adds FSM information to the specified window.
Properties	Displays the FSM Properties dialog box containing detailed information about the FSM.

Table 80. FSM List Menu

Popup Menu Item	Description
View FSM	Opens the FSM in the FSM Viewer window.
View Declaration	Opens the source file for the FSM instance.
Set Context	Changes the context to the FSM instance.
Add to <window>	Adds FSM information to the specified window.
Options	Displays the FSM Display Options dialog box, which allows you to control: • How FSM information is added to the Wave Window. • How much information is shown in the Instance Column.

FSM Viewer Window

To access:

- From the **FSM List window**, double-click the FSM you want to analyze.

- From the **Objects**, **Locals**, **Wave**, or **Code Coverage Analyze FSM Analysis** windows, click the **FSM button** ⬚ for the FSM you want to analyze.

Use this window to graphically analyze finite state machines in your design.

Description

Figure 82. FSM Viewer Window
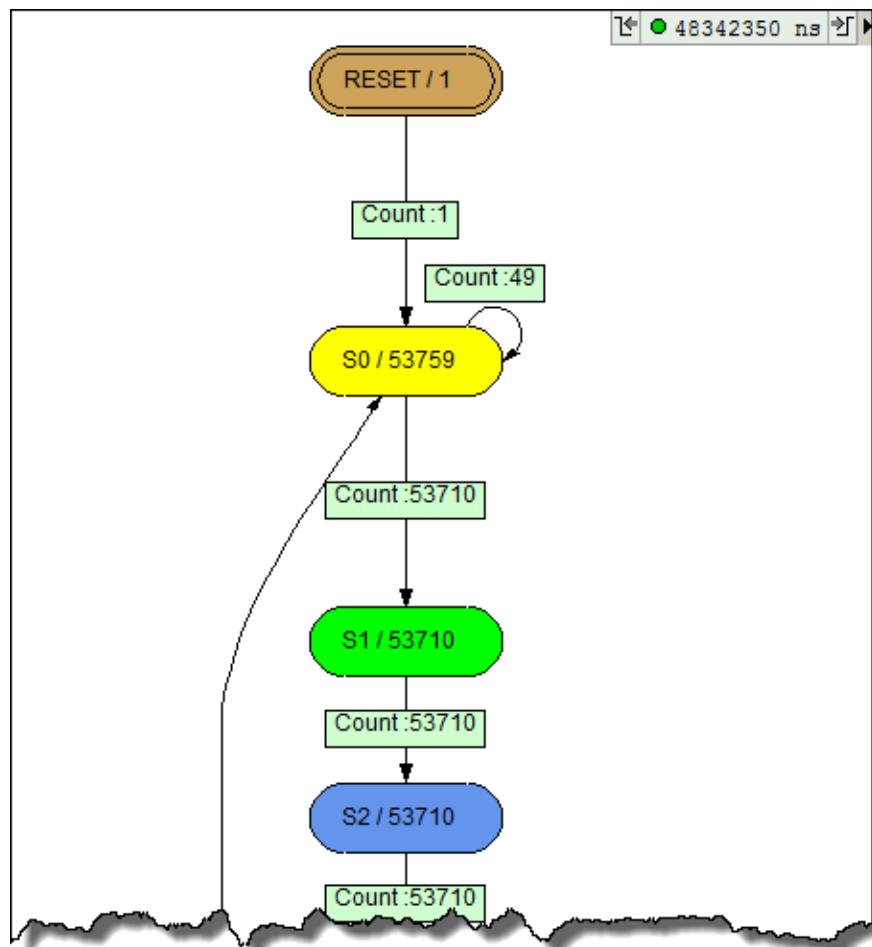
- Analyze FSMs and their coverage data — You must specify +cover, or explicitly +cover=f, during compilation and -coverage on the vsim command line to fully analyze FSMs with coverage data.

- Analyze FSMs without coverage data —You must specify +acc, or explicitly +acc=f, during compilation (vcom/vlog) or optimization (vopt) to analyze FSMs with the FSM Viewer window.

The **FSM View** menu becomes available in the **Main** menu when the FSM View window is active.

Objects

Table 81. FSM Viewer Window — Graphical Elements
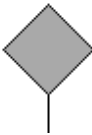
Graphical Element	Description	Definition
CTRL	Blue state bubble	Default appearance for non-reset states.
RD_WD_2	Green state bubble	Indicates the FSMs current state, or the state of the wave cursor location (when tracking the wave cursor).
RD_WD_1	Yellow state bubble	Indicates the FSMs previous state, or the state of the wave cursor location (when tracking the wave cursor).
IDLE	Tan state bubble with double outline.	Indicates a reset state.
(gray diamond)	Gray diamond	Indicates there are several transitions to reset with the same expression. This is a placeholder to reduce the number of objects drawn in the window. You can view all common expressions by choosing: **FSm View > Transitions to "reset" > Show All**
(transition box)	Transition box	Contains information about the transition, • Cond: specifies the transition condition • Count: specifies the coverage count
(black arrow)	Black transition line.	Indicates a transition.

Table 81. FSM Viewer Window — Graphical Elements (continued)

Graphical Element	Description	Definition
➡	Red transition line.	Indicates a transition that has zero (0) coverage.

1. The condition format is based on the GUI_expression_format operators, as described in Syntax and Conventions in the Command Reference Manual.

Table 82. FSM View Window Popup Menu

Popup Menu Item	Description
Transition	Only available when right-clicking on a transition. • **Goto Source** — Opens the source file containing the state machine and highlights the transition code. • **View Full Text** — Opens the View Transition dialog box, which contains the full text of the condition.
View Declaration	Opens the source file and bookmarks the file line containing the declaration of the state machine
Zoom Full	Displays the FSM completely within the window.
Set Context	Executes the env command to change the context to that of the state machine.
View Layout	Allows you to choose the orientation of the window: horizontal or vertical.
Hide Selected	Allows you to hide the selected state or transition
Unhide All	Resets the view to show all hidden states or transitions.
Add to …	Adds information about the state machine to the specific window.
Properties	Displays the FSM Properties dialog box containing detailed information about the FSM.

Table 83. FSM View Menu

FSM View Menu Item	Description
Show State Counts	Displays the coverage counts for each state in the state bubble.
Show Transition Counts	Displays the coverage counts for each transition.
Show Transition Conditions	Displays the condition for each transition. The condition format is based on the GUI_expression_format operators. (Refer to Syntax and Conventions in the

Table 83. FSM View Menu (continued)

FSM View Menu Item	Description
	Command Reference Manual for a description of the GUI_expression_format operators.)
Enable Info Mode Popups	Displays popup information when you hover over a state or transition.
Transitions to "Reset"	Controls the display of transitions to a reset state: • Show All • Show None — Will also add a "hide all" note to the lower-right hand corner. • Hide Asynchronous Only • Combine Common Transitions — (default) Creates a single transition for any transitions to reset that use the same condition. The transition is shown from a gray diamond that acts as a placeholder.
Options	Displays the FSM Display Options dialog box, which allows you to control: • How FSM information is added to the Wave Window. • How much information is shown in the Instance Column • Whether or not the Current Time information is shown in the upper-right corner of the window.

FSM Viewer Window Tasks

This section describes tasks for using the FSM Viewer window.

Using the Mouse in the FSM Viewer

Multiple mouse operations are defined for the FSM Viewer.

- The mouse wheel performs zoom & center operations on the diagram.

    ◦ **Mouse wheel up** — zoom out.

    ◦ **Mouse wheel down** — zoom in.

    Whether zooming in or out, the view will re-center towards the mouse location.

- **Left mouse button** — click and drag to move the view of the FSM.

- **Middle mouse button** — click and drag to perform the following stroke actions:

    ◦ **Up and left** — Zoom Full.

    ◦ **Up and right** — Zoom Out. The amount is determined by the distance dragged.

    ◦ **Down and right** — Zoom In on the area of the bounding box.

Using the Keyboard in the FSM Viewer

Certain keyboard operations are defined for the FSM Viewer.

- Arrow Keys — scrolls the window in the specified direction.

    ◦ Unmodified — scrolls by a small amount.

    ◦ **Ctrl+<arrow key>** — scrolls by a larger amount.

    ◦ **Shift+<arrow key>** — shifts the view to the edge of the display.

Exporting the FSM Viewer Window as an Image

Save the FSM view as an image for use in other applications.

Procedure

1. Select the FSM Viewer window.

2. Export to one of the following formats:

   - Postscript — **File > Print Postscript**

   - Bitmap (*.bmp*) — **File > Export > Image**

     ◦ JPEG (*.jpg*)

     ◦ PNG (*.png*)

     ◦ GIF (*.gif*)

Instance Coverage Window

To access:

- **View > Coverage > Instance Coverage**

- view instance command
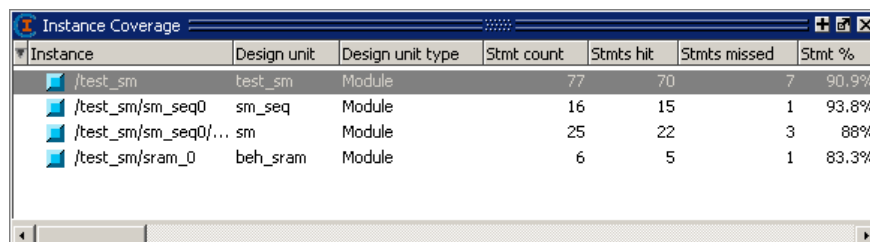
Use this window to analyze coverage statistics for each instance in a flat, non-hierarchical view. You can sort data columns to be more meaningful, and not be confused by hierarchy. This window contains the same code coverage statistics columns as in the Files and Structure windows.

Description

> **Note:**
> This window is specific to the collection of coverage metrics, therefore you must have run your simulation with coverage collection enabled. Refer to Code Coverage in the User's Manual for more information.

Figure 83. Instance Coverage Window

Objects

Table 84. Columns in the Instance Coverage Window

Column name	Description
Assertion %	the number of hits from the total number of assertions, as a percentage
Assertion graph	a bar chart displaying the Assertion %; if the percentage is below 90%, the bar is red; 90% or more, the bar is green; you can change this threshold percentage by editing the PrefCoverage(cutoff) preference variable
Assertion hits	Assertion hits shows different counts based on whether the -assertcounts or -assertdebug argument is used with the vsim command: • with -assertcounts or -assertdebug: number of assertions whose pass count is greater than 0 and whose fail count is equal to 0 • without -assertcounts or -assertdebug: number of assertions whose pass count is equal to 1 and whose fail count is equal to 0

Table 84. Columns in the Instance Coverage Window (continued)

Column name	Description
Assertion misses	the number of assertions whose fail count is greater than 0 or whose fail and pass counts both are equal to 0 (vacuous assertions)
Branch %	the current ratio of Branch hits to Branch count
Branch count	Files window — the number of executable branches in each file Structure window — the number of executable branches in each level and all levels under that level
Branch graph	a bar chart displaying the Branch %; if the percentage is below 90%, the bar is red; 90% or more, the bar is green; you can change this threshold percentage by editing the PrefCoverage(cutoff) preference variable
Branches hit	the number of executable branches that have been executed in the current simulation
Branches missed	the number of executable branches that were not executed in the current simulation
Cover %	the number of hits from the total number of cover directives, as a percentage
Cover graph	a bar chart displaying the Cover directive %; if the percentage is below 90%, the bar is red; 90% or more, the bar is green; you can change this threshold percentage by editing the PrefCoverage(cutoff) preference variable
Cover hits	the number of cover directives whose count values are greater than or equal to the at_least value.
Cover missed	the number of cover directives whose count values are less than the at_least value
Covergroup %	the number of hits from the total number of covergroups, as a percentage
Covergroup bins	the number of covergroup bins
Covergroup graph	a bar chart displaying the Covergroup %; if the percentage is below 90%, the bar is red; 90% or more, the bar is green
Covergroup hit bins	the number of hit covergroup bins
Covergroup missed bins	the number of missed covergroup bins
Covergroup weighted missed bins	the number of weighted covergroup bins that were missed
Design Unit	the name of the design unit
Design Unit Type	the type of design unit
FEC Condition %	the current ratio of FEC Condition hits to FEC Condition rows

Table 84. Columns in the Instance Coverage Window (continued)

Column name	Description
FEC Condition graph	a bar chart displaying the FEC Condition %; if the percentage is below 90%, the bar is red; 90% or more, the bar is green; you can change this threshold percentage by editing the PrefCoverage(cutoff) preference variable
FEC Condition rows	the number of FEC conditions in each instance
FEC Conditions hit	the number of times the FEC conditions in an instance that have been executed
FEC Conditions missed	the number of FEC conditions in an instance that were not executed
FEC Expression %	the current ratio of FEC Expression hits to FEC Expression rows
FEC Expression graph	a bar chart displaying the FEC Expression %; if the percentage is below 90%, the bar is red; 90% or more, the bar is green; you can change this threshold percentage by editing the PrefCoverage(cutoff) preference variable
FEC Expression rows	the number of executable expressions in each instance
FEC Expressions hit	the number of times expressions in an instance have been executed
FEC Expressions missed	the number of executable expressions in an instance that were not executed
State %	the current ratio of State hits to State rows
State graph	a bar chart displaying the State %; if the percentage is below 90%, the bar is red; 90% or more, the bar is green; you can change this threshold percentage by editing the PrefCoverage(cutoff) preference variable
States	the number of states encountered in each instance
States hit	the number of times the states were hit
States missed	the number of states in an instance that were not hit
Stmt %	the current ratio of Stmt hits to Stmt count
Stmt graph	a bar chart displaying the Stmt %; if the percentage is below 90%, the bar is red; 90% or more, the bar is green; you can change this threshold percentage by editing the PrefCoverage(cutoff) preference variable
Stmt count	the number of executable statements in each level and all levels under that level
Stmts hit	the number of executable statements that were executed in each level and all levels under that level
Stmts missed	the number of executable statements that were not executed in each level and all levels under that level
Toggle %	the current ratio of Toggle hits to Toggle nodes

Table 84. Columns in the Instance Coverage Window (continued)

Column name	Description
Toggle graph	a bar chart displaying the Toggle %; if the percentage is below 90%, the bar is red; 90% or more, the bar is green; you can change this threshold percentage by editing the PrefCoverage(cutoff) preference variable
Toggle nodes	the number of points in each instance where the logic will transition from one state to another
Toggles hit	the number of nodes in each instance that have transitioned at least once
Toggles missed	the number of nodes in each instance that have not transitioned at least once
Total Coverage	The weighted average of all the coverage types (functional coverage and code coverage) is recursive. Deselect **Code Coverage > Enable Recursive Coverage Sums** to view results for the local instance. Refer to Calculation of Total Coverage in the User's Manual for coverage statistics details.
Transition %	the current ratio of Transition hits to Transition rows
Transition graph	a bar chart displaying the State %; if the percentage is below 90%, the bar is red; 90% or more, the bar is green; you can change this threshold percentage by editing the PrefCoverage(cutoff) preference variable
Transitions	the number of transitions encountered in each instance
Transitions hit	the number of times the transitions were hit
Transitions missed	the number of transitions in an instance that were not hit

Table 85. Instance Coverage Popup Menu

Popup Menu Item	Description
Code coverage reports	Displays the Coverage Text Report dialog box, which allows you to create reports based on your code coverage metrics.
Set Filter	Displays the Figure 84
Clear code coverage data	Clears all of the code coverage data from the GUI.
Test Analysis	Executes coverage analyze on the selected instance and prints information to the Test Analysis window. • Find Least Coverage • Find Most Coverage • Find Zero Coverage • Find Non Zero Coverage • Summary
XML Import Hint	Displays the XML Import Hint dialog box with information about the Link Type and Name.

Usage Notes

Setting a Coverage Threshold

You can specify a percentage above or below which you do not want to see coverage statistics. For example, you might set a threshold of 85% such that only objects with coverage below that percentage are displayed. Anything above that percentage is filtered.

1. Right-click any object in the Instance Coverage window.

2. Choose **Set filter**. The "Filter instance list" dialog appears.

Figure 84. Filter Instance List Dialog Box
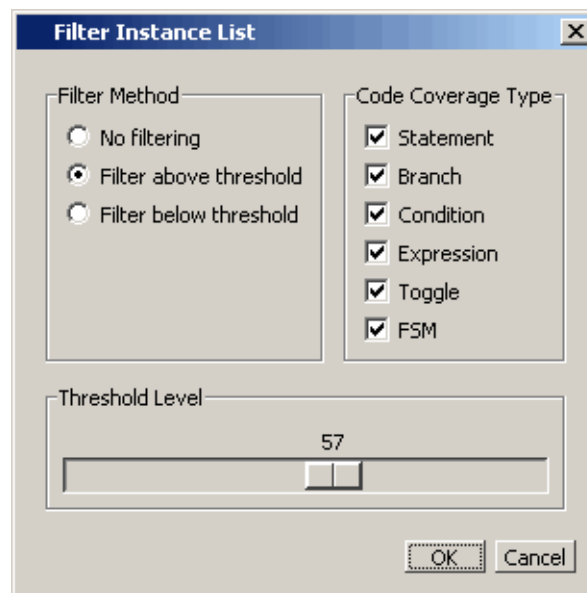
Library Window

To access:

- **View > Library**

- view library command

Use this window to view design libraries and compiled design units.

Description

Figure 85. Library Window
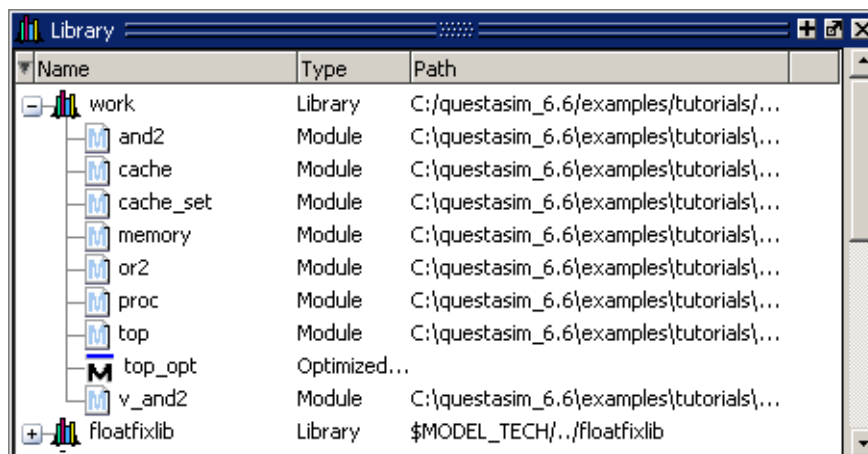
Objects

Table 86. Library Window Columns

Column Title	Description
Name	Name of the library or design unit
Path	Full pathname to the file
Type	Type of file

Table 87. Library Window Popup Menu

Popup Menu Item	Description
Simulate	Loads a simulation of the selected design unit.
Simulate with full Optimization	Loads a simulation of the selected design unit, implicitly calling vopt with no visibility

Table 87. Library Window Popup Menu (continued)

Popup Menu Item	Description
Simulate with Coverage	Loads a simulation of the selected design unity, enabling coverage (-coverage)
Edit	Opens the selected file in your editor window.
Refresh	Reloads the contents of the window
Recompile	Compiles the selected file.
Optimize	Runs vopt on the selected file.
Update	
Create Wave	Runs the wave create command for any ports in the selected design unit.
Delete	Removes a design unit from the library or runs the vdel command on a selected library.
Copy	Copies the directory location of libraries or the library location of design units within the library.
New	Allows you to create a new library with the Create a New Library dialog box.
Properties	Displays information about the selected library or design unit.

List Window

To access:

- **View > List**

- view list command

The List window displays simulation results in tabular format. Use this window to display a textual representation of waveforms, which you can configure to show events and delta events for the signals or objects you have added to the window.

Description

Common List window tasks include:

- Using gating expressions and trigger settings to focus in on particular signals or events. See Configuring New Line Triggering.

- Debugging delta delay issues. Refer to Delta Delays in the User's Manual for more information.

The window is divided into two adjustable panes, which allows you to scroll horizontally through the listing on the right, while keeping time and delta visible on the left.

Figure 86. Tabular Format of the List Window
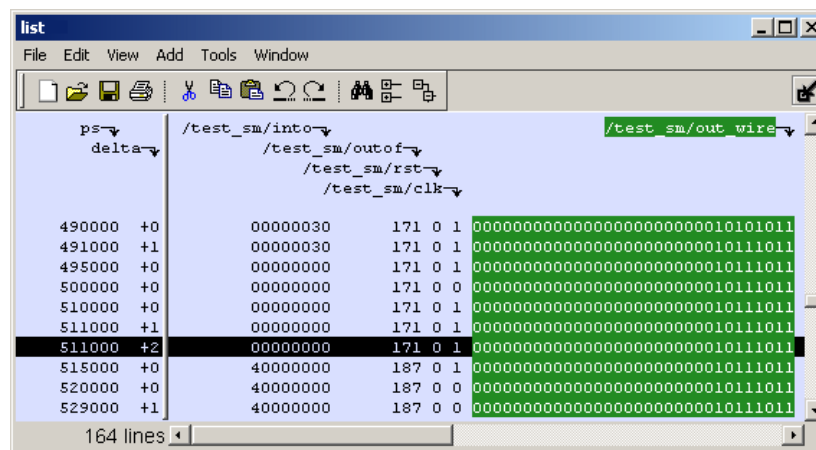
You can view the following object types in the List window:

- VHDL — Signals, aliases, process variables, and shared variables

- Verilog — Nets, registers, and variables

- SystemC — Primitive channels, ports, and transactions

- Comparisons — Comparison objects; Refer to Waveform Compare in the User's Manual for more information

- Virtuals — Virtual signals and functions

- SystemVerilog and PSL assertions — Assert directives

- SystemVerilog and PSL cover directives — Cover directives

- Questa Verification IP objects (Refer to Questa Verification IP Objects in the GUI in the User's Manual.)

- SystemVerilog — Transactions

Figure 87. List Window
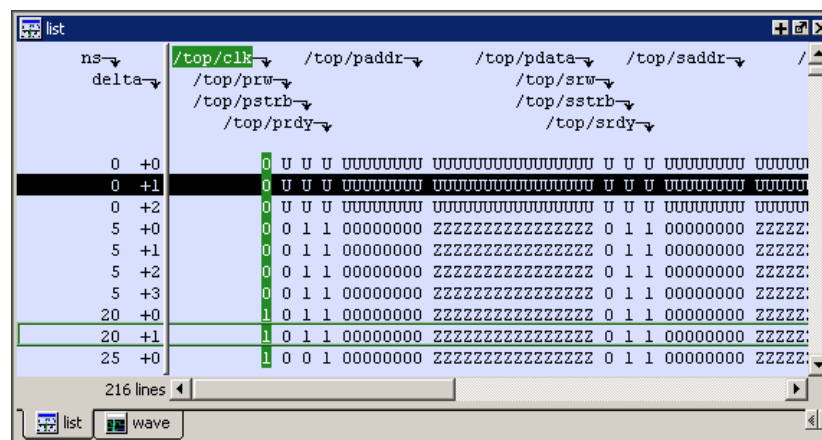
This section describes the GUI elements specific to the List window.

The List window is divided into two adjustable panes, which allow you to scroll horizontally through the listing on the right, while keeping time and delta visible on the left.

- The left pane shows the time and any deltas that exist for a given time.

- The right pane contains the data for the signals and objects you have added for each time shown in the left pane. The top portion of the window contains the names of the signals. The bottom portion shows the signal values for the related time.

Note:
The display of time values in the left column is limited to 10 characters. Any time value is more than 10 characters the characters are replaced with the following:

```
too narrow
```

The markers in the List window are analogous to cursors in the Wave window. You can add, delete and move markers in the List window similarly to the Wave window. You will notice two different types of markers:

- Active Marker — The most recently selected marker shows as a black highlight.

- Non-active Marker — Any markers you have added that are not active are shown with a green border.

You can manipulate the markers in the following ways:

- **Setting a marker** — When you click in the right-hand portion of the List window, you will highlight a given time (black horizontal highlight) and a given signal or object (green vertical highlight).

- Moving the active marker — List window markers behave the same as Wave window cursors. There is one active marker which is where you click along with inactive markers generated by the Add Marker command. Markers move based on where you click. The closest marker (either active or inactive) will become the active marker, and the others remain inactive.

- Adding a marker — You can add an additional marker to the **List** window by right-clicking at a location in the right-hand side and selecting **Add Marker**.

- Deleting a marker — You can delete a marker by right-clicking in the List window and selecting **Delete Marker**. The marker closest to where you clicked is the marker that will be deleted.

Objects

Table 88. List Window Popup Menu

Popup Menu Item	Description
Examine	Displays the value of the signal over which you used the right mouse button, at the time selected with the Active Marker
Annotate Diff	Allows you to annotate a waveform comparison difference with additional information. For more information refer to the compare annotate command. Available only during a Waveform Comparison
Ignore Diff	Flags the waveform compare difference as "ignored". For more information refer to the compare annotate command. Available only during a Waveform Comparison
Add Marker	Adds a marker at the location of the Active Marker
Delete Marker	Deletes the closest marker to your mouse location

Usage Notes

Other List Window Tasks

List > List Preferences — Allows you to specify the preferences of the List window.

File > Export > Tabular List — Exports the information in the List window to a file in tabular format. Equivalent to the command:

```
write list <filename>
```

File > Export > Event List — Exports the information in the List window to a file in print-on-change format. Equivalent to the command:

```
write list -event <filename>
```

File > Export > TSSI List — Exports the information in the List window to a file in TSSI. Equivalent to the command:

```
write tssi -event <filename>
```

Edit > Signal Search — Allows you to search the List window for activity on the selected signal.

List Window Tasks

You can perform multiple tasks using the List window.

Adding Data to the List Window

You can add objects to the List window from the menu, the command line, or the button bar.

Procedure

To add data to the List window, perform one of the following methods:

If you want to …	Do the following …
Use menu to add data	Right-click signals and objects in the Objects window or the Structure window and select **Add > to List**.
Use the command line to add data	Use the add list command.
Use the button bar to add data	Use the Add Selected to Window Button".

Selecting Multiple Signals

You can create a larger group of signals and assign a new name to this group.

Procedure

1. Select a group of signals

   • Shift-click on signal columns to select a range of signals.

   • Control-click on signal columns to select a group of specific signals.

2. Choose **List > Combine Signals**

3. Complete the Combine Selected Signals dialog box:

- **Name** — Specify the name you want to appear as the name of the new signal.

- **Order of Indexes** — Specify the order of the new signal as ascending or descending.

- **Remove selected signals after combining** — Specify whether the grouped signals should remain in the List window.

4. This process creates virtual signals. For more information, refer to Virtual Signals in the User's Manual.

Setting Time Markers in the List Window

Time markers in the List window are similar to cursors in the Wave window. Time markers tag lines in the data table so you can quickly jump back to that time. Markers are indicated by a thin box surrounding the marked line.

Figure 88. Time Markers in the List Window
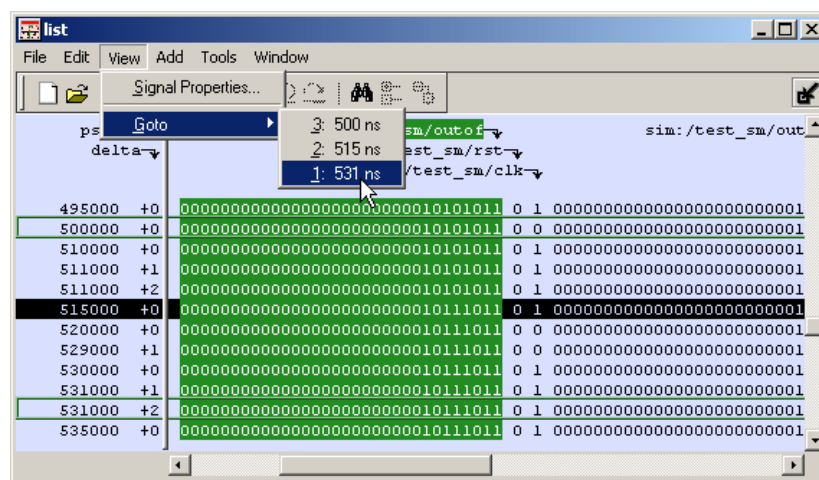
Working with Markers

The table below summarizes actions you can take with markers.

Table 89. Actions for Time Markers

Action	Method
Add marker	Select a line and then choose **List > Add Marker**
Delete marker	Select a tagged line and then choose **List > Delete Marker**
Goto marker	Choose **View > Goto <*time*>** (only available when undocked)

Expanded Time Viewing in the List Window

Event time may be shown in the List window in the same manner as delta time by using the -delta events option with the configure list command.

When the List window displays event times, the event time is relative to events on other signals also displayed in the List window. This may be misleading, as it may not correspond to event times displayed in the Wave window for the same events if different signals are added to the Wave and List windows.

The write list command (when used after the configure list -delta events command) writes a list file in tabular format with a line for every event. Please note that this is different from the write list -events command, which writes a non-tabular file using a print-on-change format.

The following examples illustrate the appearance of the List window and the corresponding text file written with the write list command after various options for the configure list -delta command are used.

Figure 89 shows the appearance of the List window after the configure list -delta none command is used. It corresponds to the file resulting from the write list command. No column is shown for deltas or events.

Figure 89. List Window After configure list -delta none Option is Used
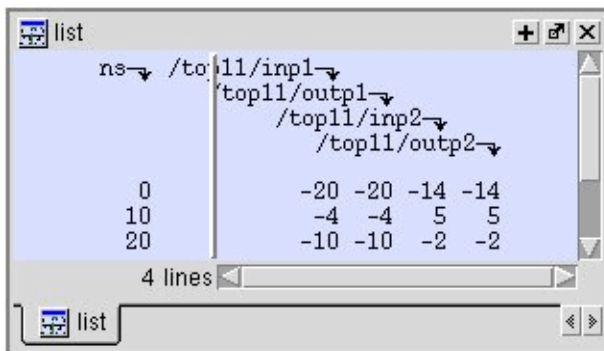
Figure 90 shows the appearance of the List window after the configure list -delta collapse command is used. It corresponds to the file resulting from the write list command. There is a column for delta time and only the final delta value and the final value for each signal for each simulation time step (at which any events have occurred) is shown.

Figure 90. List Window After configure list -delta collapse Option is Used
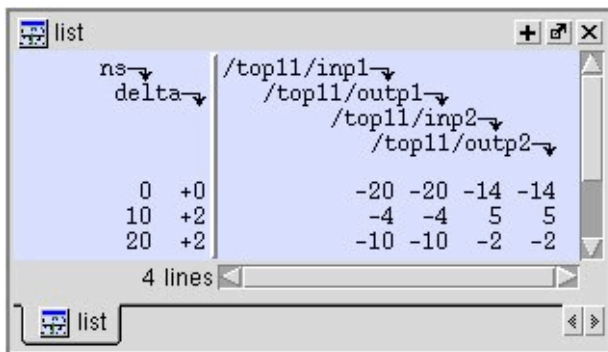
Figure 91 shows the appearance of the List window after the configure list -delta all option is used. It corresponds to the file resulting from the write list command. There is a column for delta time, and each delta time step value is shown on a separate line along with the final value for each signal for that delta time step.

Figure 91. List Window After write list -delta all Option is Used
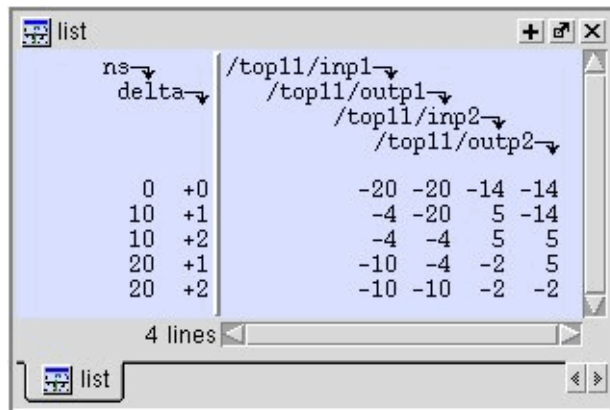
Figure 92 shows the appearance of the List window after the configure list -delta events command is used. It corresponds to the file resulting from the write list command. There is a column for event time, and each event time step value is shown on a separate line along with the final value for each signal for that event time step. Since each event corresponds to a new event time step, only one signal will change values between two consecutive lines.

Figure 92. List Window After write list -event Option is Used
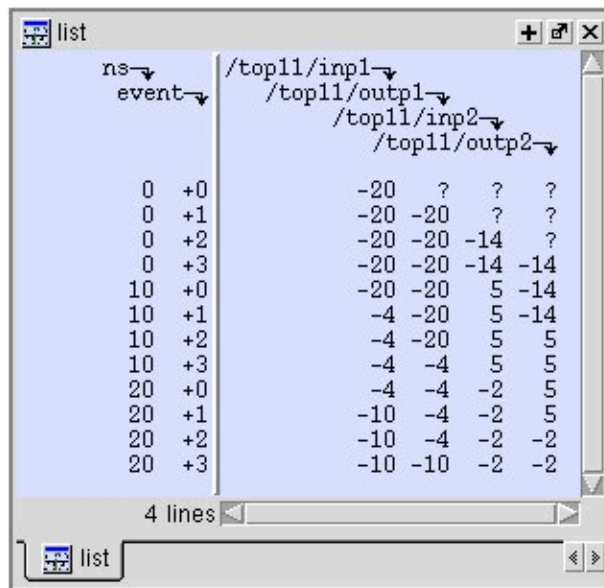
Searching in the List Window

Use the List window to locate objects.

Procedure

Search the List window using one of the following methods:

If you want to...	Do the following...
Find signal names	• Choose **Edit > Find** • Click the **Find** toolbar button (binoculars icon). • Use the find command. The first two of these options will open a Find mode toolbar at the bottom of the List window. By default, the Search For option is set to Name. For more information, see Find and Filter Functions.
Search for values or transitions	• Choose **Edit > Signal Search** • Click the **Find** toolbar button (binoculars icon) and select **Search For > Value** from the Find toolbar that appears at the bottom of the List window. • Use the search command.

Searching for Values or Transitions

The search command lets you search for values of selected signals.

When you choose **Edit > Signal Search**, the List Signal Search dialog box appears.

Figure 93. Wave Signal Search Dialog Box
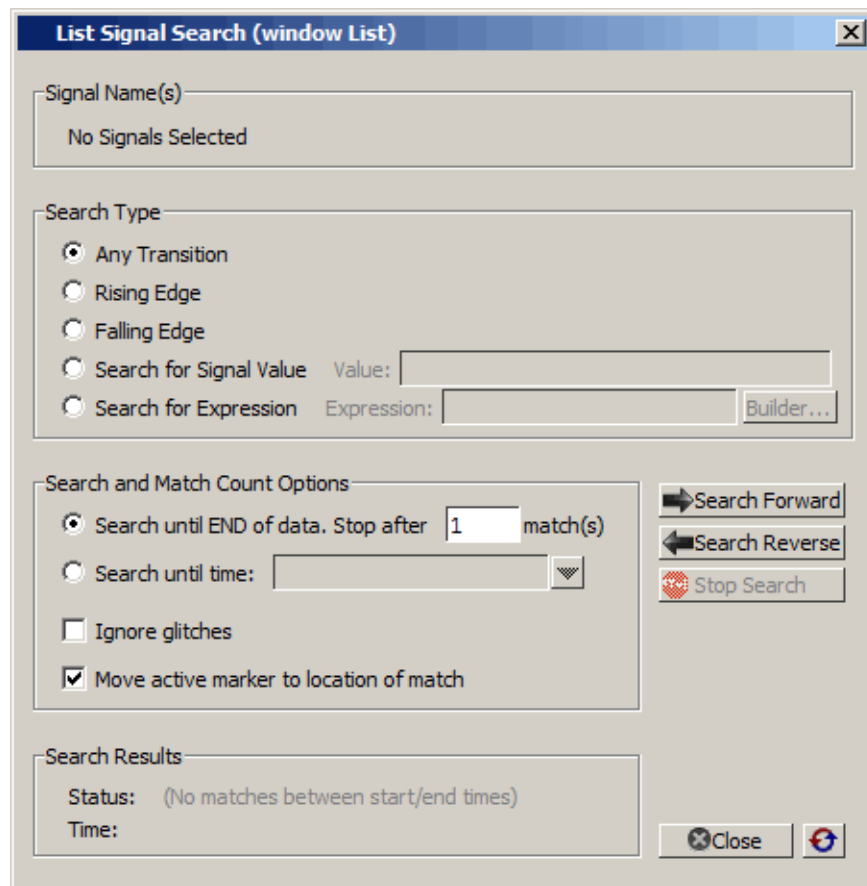
One option of note is **Search for Expression**. The expression can involve more than one signal but is limited to signals currently in the window. Expressions can include constants, variables, and DO files. Refer to Expression Syntax in the Command Reference Manual for more information.

Any search terms or settings you enter are saved from one search to the next in the current simulation. To clear the search settings during debugging click the Reset To Initial Settings button. The search terms and settings are cleared when you close Questa SIM.

Note:
If your signal values are displayed in binary radix, refer to Searching for Binary Signal Values in the GUI in the Syntax and Conventions section of the Command Reference Manual for details on how signal values are mapped between a binary radix and std_logic.

Using the Expression Builder for Expression Searches

The Expression Builder is a feature of the List Signal Search dialog box and the List trigger properties dialog box. You can use it to create a search expression that follows the GUI_expression_format.

Procedure

1. Choose **Edit > Signal Search ...**from the main menu. This displays the Wave Signal Search dialog box.

2. Select **Search for Expression**.

3. Click the **Builder** button. This displays the Expression Builder dialog box shown in Figure 94.

Figure 94. Expression Builder Dialog Box
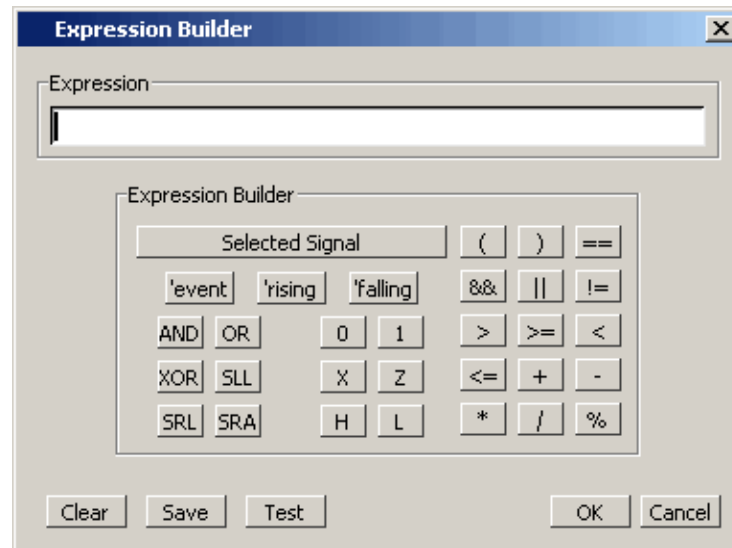
4. You click the buttons in the Expression Builder dialog box to create a GUI expression. Each button generates a corresponding element of expression syntax, and is displayed in the Expression field. (Refer to GUI_expression_format Expression Syntax in the Command Reference Manual). In addition, you can use the **Selected Signal** button to create an expression from signals you select from the List window.

5. For example, instead of typing in a signal name, you can select signals in a List window and then click **Selected Signal** in the Expression Builder. This displays the Select Signal for Expression dialog box shown in Figure 95.

Figure 95. Selecting Signals for Expression Builder
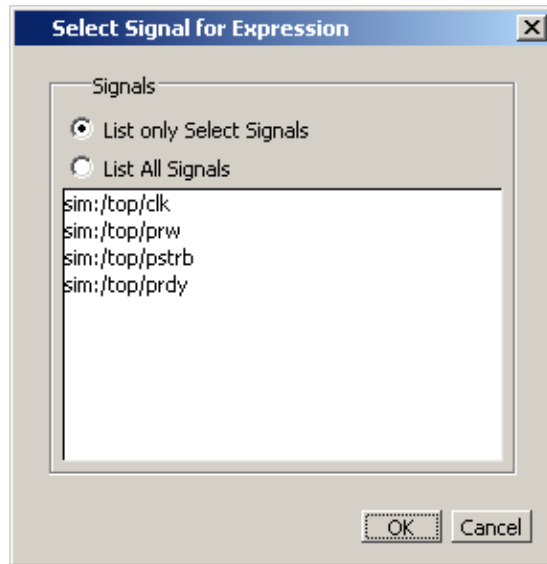
6. Note that the buttons in this dialog box allow you to determine the display of signals you want to put into an expression:

   List only Select Signals — list only those signals that are currently selected in the parent window.

   List All Signals — list all signals currently available in the parent window.

7. Once you have selected the signals you want displayed in the Expression Builder, click **OK**.

Saving an Expression to a Tcl Variable

Clicking the **Save** button will save the expression to a Tcl variable. Once saved this variable can be used in place of the expression. For example, say you save an expression to the variable *<foo>*

Here are some operations you could do with the saved variable:

- Read the value of *<foo>* with the set command:

     set foo

- Put $foo in the Expression: entry box for the Search for Expression selection.

- Issue a searchlog command using foo:

     searchlog -expr $foo 0

Formatting the List Window

Modify the display properties and objects of the List window to create a view that is easiest for you to use.

Setting List Window Display Properties

Before you add objects to the List window, you can set the window's display properties. To change when and how a signal is displayed in the List window, choose **Tools > List Preferences** from the List window menu bar (when the window is undocked).

Figure 96. Modifying List Window Display Properties
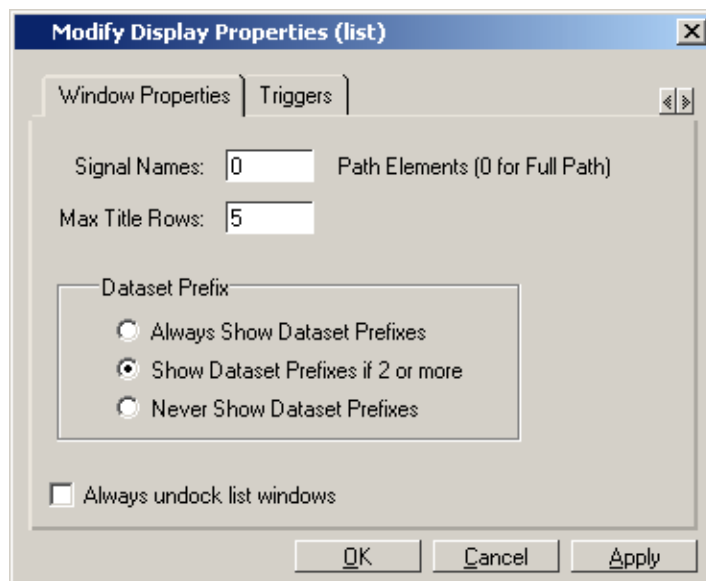
Formatting Objects in the List Window

You can adjust various properties of objects to create the view you find most useful. Select one or more objects and then choose **View > Signal Properties** from the List window menu bar (when the window is undocked).

Changing Radix (base) for the List Window

One common adjustment you can make to the List window display is to change the radix (base) of an object. To do this, choose **View > Properties** from the main menu, which displays the List Signal Properties dialog box. Figure 97 shows the list of radix types you can select in this dialog box.

Figure 97. List Signal Properties Dialog
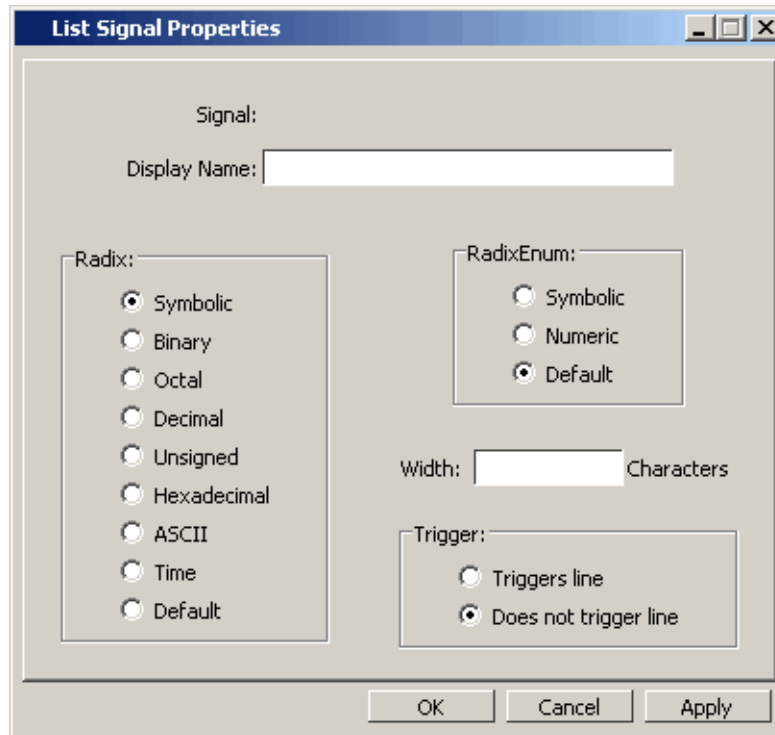
The default radix type is symbolic, which means that for an enumerated type, the window lists the actual values of the enumerated type of that object. For the other radix types (binary, octal, decimal, unsigned, hexadecimal, ASCII, time), the object value is converted to an appropriate representation in that radix.

Changing the radix can make it easier to view information in the List window. Compare the image below (with decimal values) with the image in the section "List Window" on page 158 (with symbolic values).

Figure 98. Changing the Radix in the List Window
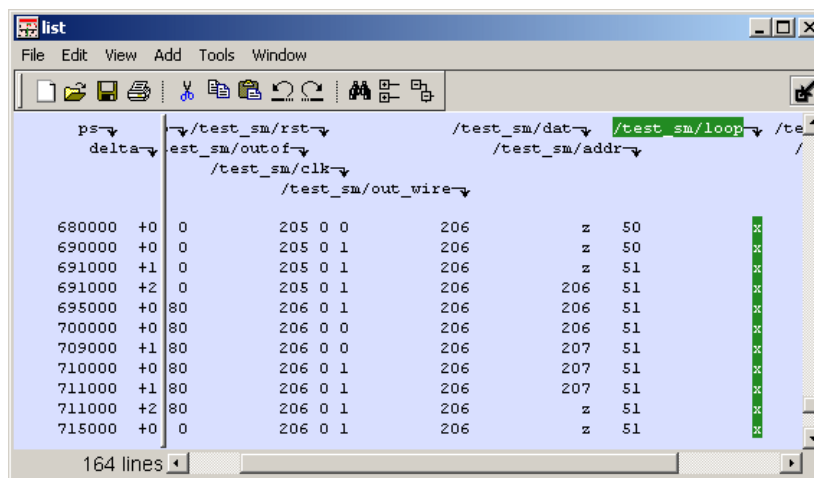
In addition to the List Signal Properties dialog box, you can also change the radix:

- Change the default radix for the current simulation using **Simulate > Runtime Options** (Main window)

- Change the default radix for the current simulation using the radix command.

- Change the default radix permanently by editing the DefaultRadix variable in the *modelsim.ini* file. (Refer to DefaultRadix in the User's Manual.)

Saving the Window Format

By default, all List window information is lost once you close the window. If you want to restore the windows to a previously configured layout, you must save a window format file as follows.

Procedure

1. Add the objects you want to the List window.

2. Edit and format the objects to create the view you want.

3. Save the format to a file by choosing **File > Save Format**. This opens the Save Format dialog box where you can save List window formats in a *.do* file.

    To use the format file, start with a blank List window and run the DO file in one of two ways:

    - Invoke the do command from the command line:

        VSIM> do <my_format_file>

    - Choose **File > Load**.

        ___
        Note:
        Window format files are design-specific. Use them only with the design you were simulating when they were created.
        ___

    In addition, you can use the write format restart command to create a single *.do* file that will recreate all debug windows and breakpoints (refer to Saving and Restoring Breakpoints in the User's Manual) when invoked with the do command in subsequent simulation runs. The syntax is:

        write format restart <filename>

    If the *modelsim.ini* variable is set to this *.do* filename, it will call the write format restart command upon exit. (Refer to ShutdownFile in the User's Manual.)

Combining Signals into Buses

You can combine signals in the List window into buses. A bus is a collection of signals concatenated in a specific order to create a new virtual signal with a specific value. A virtual compare signal (the result of a comparison simulation) is not supported for combination with any other signal.

Procedure

To combine signals into a bus, use one of the following methods:

If you want to …	Do the following …
Use the GUI	Select two or more signals in the Wave or List window and then choose **List > Combine Signals** from the menu bar. A virtual signal that is the result of a comparison simulation is not supported for combining with any other signal.
Use the Command Line	Use the virtual signal command at the Main window command prompt.

Configuring New Line Triggering

New line triggering refers to what events cause a new line of data to be added to the List window. By default Questa SIM adds a new line for any signal change including deltas within a single unit of time resolution.

You can set new line triggering on a signal-by-signal basis or for the whole simulation. To set for a single signal, choose **View > Signal Properties** from the List window menu bar (when the window is undocked) and select the **Triggers line** setting. Individual signal settings override global settings.

Figure 99. Line Triggering in the List Window
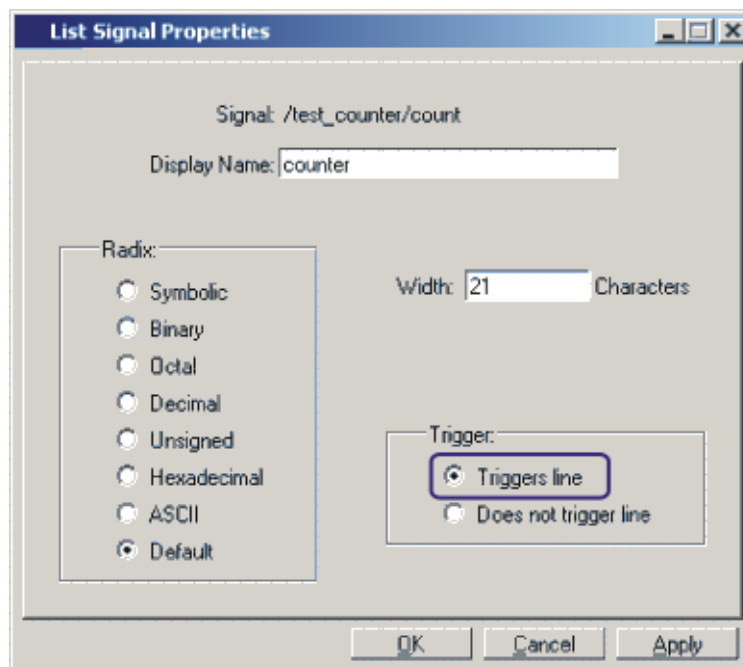
To modify new line triggering for the whole simulation, choose **Tools > List Preferences** from the List window menu bar (when the window is undocked), or use the configure command. When you choose **Tools > List Preferences**, the Modify Display Properties dialog appears:

Figure 100. Setting Trigger Properties
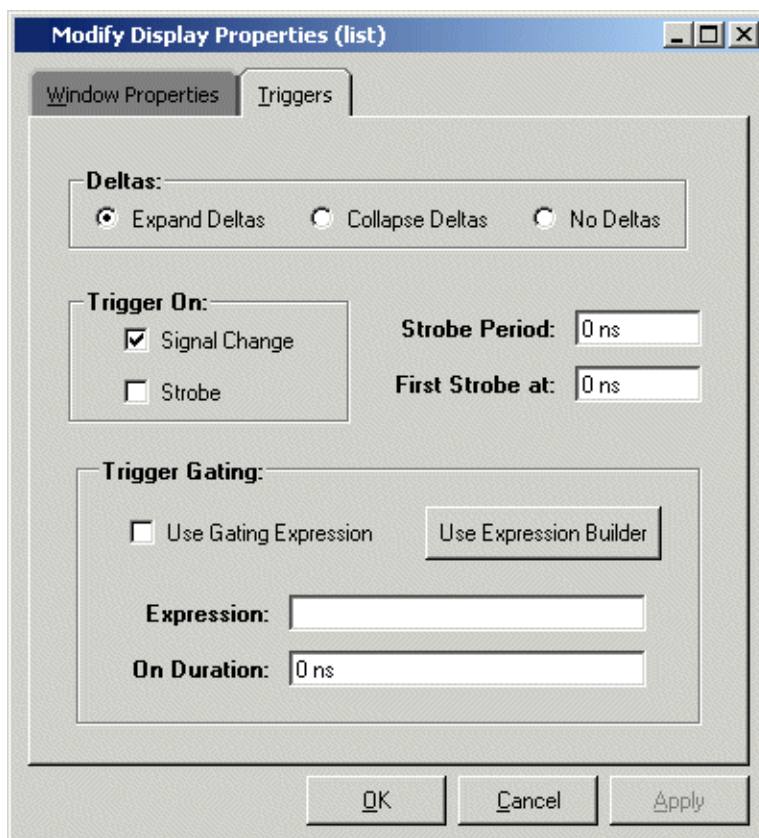
The following table summaries the triggering options:

Table 90. Triggering Options

Option	Description
Deltas	Choose between displaying all deltas (Expand Deltas), displaying the value at the final delta (Collapse Delta). You can also hide the delta column all together (No Delta), however this will display the value at the final delta.
Strobe trigger	Specify an interval at which you want to trigger data display
Trigger gating	Use a gating expression to control triggering; refer to Using Gating Expressions to Control Triggering for more details.

Using Gating Expressions to Control Triggering

Trigger gating controls the display of data based on an expression. Triggering is enabled once the gating expression evaluates to true. This setup behaves much like a hardware signal analyzer that starts recording data on a specified setup of address bits and clock edges.

Here are some points about gating expressions:

- Gating expressions affect the display of data but not acquisition of the data.

- The expression is evaluated when the List window would normally have displayed a row of data (given the other trigger settings).

- The duration determines for how long triggering stays enabled after the gating expression returns to false (0). The default of 0 duration will enable triggering only while the expression is true (1). The duration is expressed in x number of default timescale units.

- Gating is level-sensitive rather than edge-triggered.

Trigger Gating Example Using the Expression Builder

This example shows how to create a gating expression with the Questa SIM Expression Builder. Here is the procedure:

1. Select the signal in the List window by clicking on its name in the header area of the List window.

2. Undock the List window.

3. Choose Tools > List Preferences from the List window menu bar and select the Triggers tab.

4. Click the Use Expression Builder button.

Figure 101. Trigger Gating Using Expression Builder
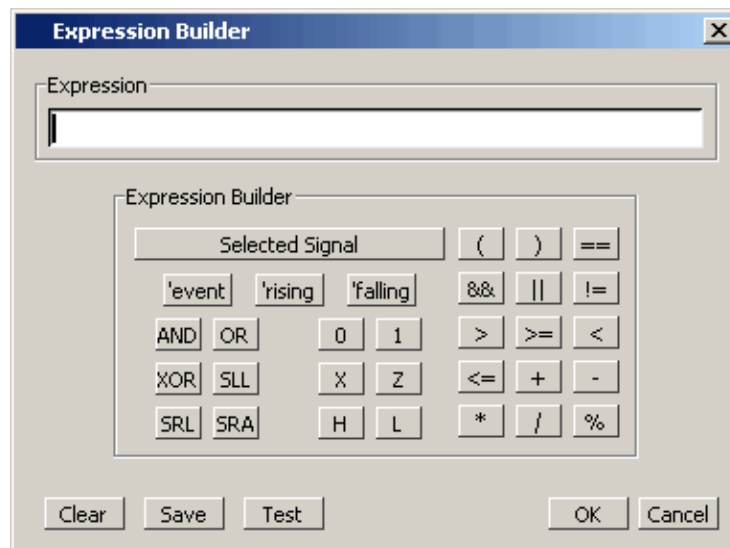
5. Click the **Selected Signal** button to open the "Select Signal for Expression" dialog box.

Figure 102. Select Signal for Expression Dialog Box
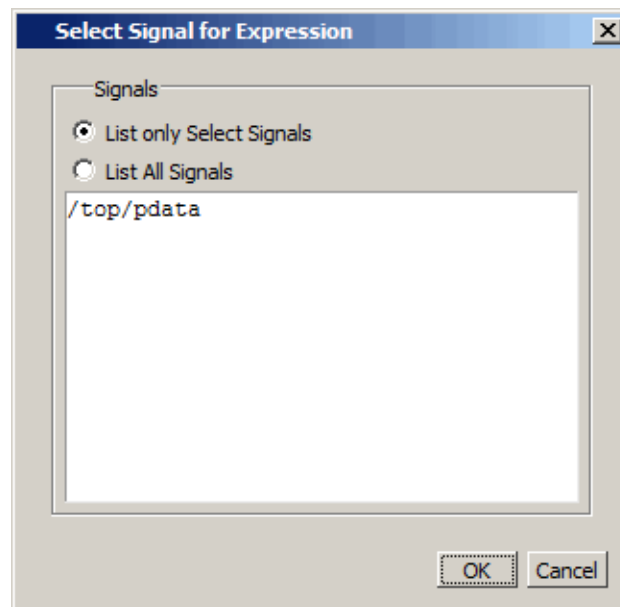
6. Click the "List only Select Signals" radio button.

7. Click the desired signal to highlight it.

8. Click the **OK** button to close the Select Signal for Expression dialog box and enter the name of the selected signal into the Expression field of the Expression Builder.

9. In the Expression Builder, click the 'rising button.

10. Click **OK** to close the Expression Builder.

    You should see the name of the signal plus "rising" added to the Expression entry box of the "Modify Display Properties" dialog box.

11. Click OK to close the dialog box.

If you already have simulation data in the List window, the display should immediately switch to showing only those cycles for which the gating signal is rising. If that is not quite what you want, you can go back to the expression builder and adjust it until you get it the way you want it.

If you want the enable signal to work like a "One-Shot" that would display all values for the next, say 10 ns, after the rising edge of enable, then set the On Duration value to 10 ns.

Trigger Gating Example Using Commands

The following commands show the gating portion of a trigger configuration statement:

configure list -usegating 1
configure list -gateduration 100
configure list -gateexpr {/test_delta/iom_dd'rising}

Refer to the configure command description in the Command Reference Manual for details.

Sampling Signals at a Clock Change

You easily can sample signals at a clock change using the add list command with the -notrigger argument. The -notrigger argument disables triggering the display on the specified signals. For example:

add list clk -notrigger a b c

When you run the simulation, List window entries for clk, a, b, and c appear only when clk changes.

If you want to display on rising edges only, you have two options:

1. Turn off the List window triggering on the clock signal, and then define a repeating strobe for the List window.

2. Define a "gating expression" for the List window that requires the clock to be in a specified state. See above.

Viewing Differences in the List Window

The Waveform Compare feature allows you to compare simulation runs and display differences in the List window. For a complete discussion of the Waveform Compare feature and how differences are displayed in both the Wave and List windows.

Locals Window

To access:

- **View > locals**

- view locals command

Use this window to display data objects declared in the current, or local, scope of the active process. These data objects are immediately visible from the statement that will be executed next, which is denoted by a blue arrow in a Source window. The contents of the window change from one statement to the next.

Description

When encountering a C breakpoint, the Locals window displays automatic local variables and their value in current C/C++ function scope.

Figure 103. Locals Window
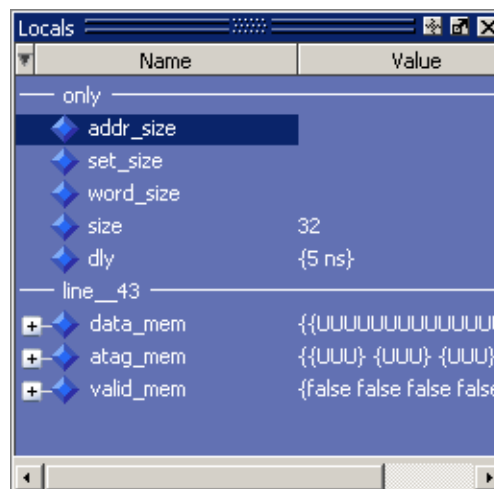
Change Selected Variable Dialog Box

This dialog box allows you to change the value of the object you selected. When you click Change, the tool executes the change command on the object.

Figure 104. Change Selected Variable Dialog Box
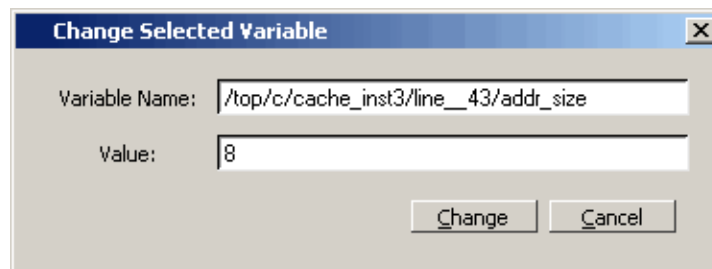
The Change Selected Variable dialog is prepopulated with the following information about the object you had selected in the Locals window:

- **Variable Name** — contains the complete name of the object.

- **Value** — contains the current value of the object.

When you change the value of the object, you can enter any value that is valid for the variable. An array value must be specified as a string (without surrounding quotation marks). To modify the values in a record, you need to change each field separately.

Objects

Column	Description
Name	lists the names of the immediately visible data objects. This column also includes design object icons for the objects, refer to the section Design Object Icons and Their Meanings for more information
Value	lists the current value(s) associated with each name
State Count	Not shown by default. This column, State Hits, and State % are all specific to coverage analysis
State Hits	Not shown by default
State %	Not shown by default

Table 91. Locals Window Columns

Popup Menu Item	Description
View Declaration	Displays, in the Source window, the declaration of the object
Add	Adds the selected object(s) to the specified window
Copy	Copies selected item to clipboard
Find	Opens the Find toolbar at the bottom of the window
Expand/Collapse	Expands or collapses data in the window
Global Signal Radix	Sets radix for selected signal(s) in all windows
Change	Displays the Change Selected Variable dialog box, which allows you to alter the value of the object

Usage Notes

This section describes tasks for using the Locals window.

Viewing Data in the Locals Window

You cannot actively place information in the Locals window, it is updated as you go through your simulation. However, there are several ways you can trigger the Locals window to be updated.

- Run your simulation while debugging.

- Select a Process from the Processes Window.

- Select a Verilog function or task or VHDL function or procedure from the Call Stack Window.

Memory Data Window

To access:

- Double-click a memory in the **Memory List** window.

Use this window to view the contents of a memory.

Description

Figure 105. Memory Data Window
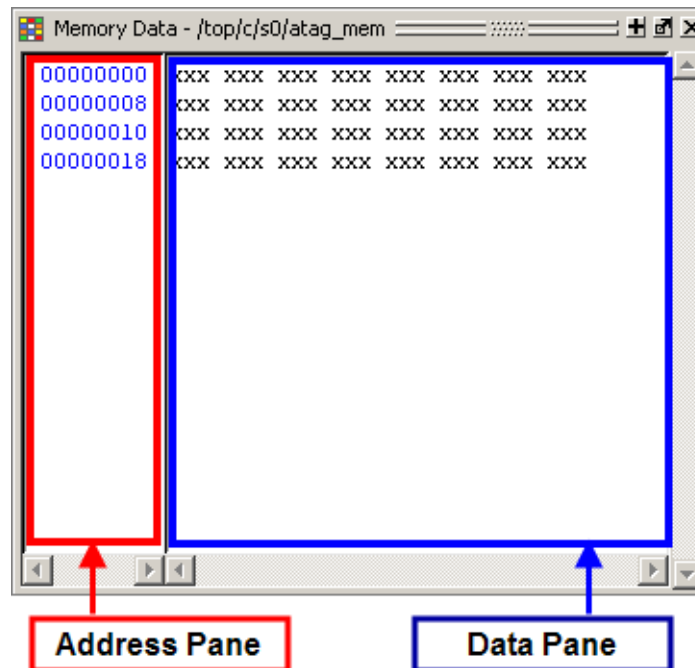
Objects

Table 92. Memory Data Popup Menu — Address Pane

Popup Menu Item	Description
Goto	Allows you to go to a specific address
Split Screen	Splits the Memory Data window to allow you to view different parts of the memory simultaneously.
Properties	Allows you to set various properties for the Memory Data window.
Close Instance	Closes the active Memory Data window.
Close All	Closes all Memory Data windows.

Table 93. Memory Data Popup Menu — Data Pane

Popup Menu Item	Description
Edit	Allows you to edit the value of the selected data.
Change	Allows you to change data within the memory through the use of the Change Memory dialog box.
Import Data Patterns	Allows you to import data patterns into the selected memory through the Import Memory dialog box.
Export Data Patterns	Allows you to export data patterns from the selected memory through the Export Memory dialog box.
Split Screen	Refer to items in the Table 92
Properties	
Close Instance	
Close All	

Table 94. Memory Data Menu

Popup Menu Item	Description
Memory Declaration	Opens a Source window to the file and line number where the memory is declared.
Compare Contents	Allows you to compare the selected memory against another memory in the design or an external file.
Import Data Patterns	Refer to items in the Table 93
Export Data Patterns	
Expand Packed Memories	Toggle the expansion of packed memories.
Identify Memories Within Cells	Toggle the identification of memories within Verilog cells.
Show VHDL String as Memory	Toggle the identification of VHDL strings as memories.
Split Screen	Refer to items in the Table 92

Usage Notes

Direct Address Navigation

You can navigate to any address location directly by editing the address in the address column. Double-click any address, type in the desired address, and hit **Enter**. The address display scrolls to the specified location.

Splitting the Memory Contents Window

To split a memory contents window into two screens displaying the contents of a single memory instance choose **Memory Data > Split Screen**.

This allows you to view different address locations within the same memory instance simultaneously.

Figure 106. Split Screen View of Memory Contents
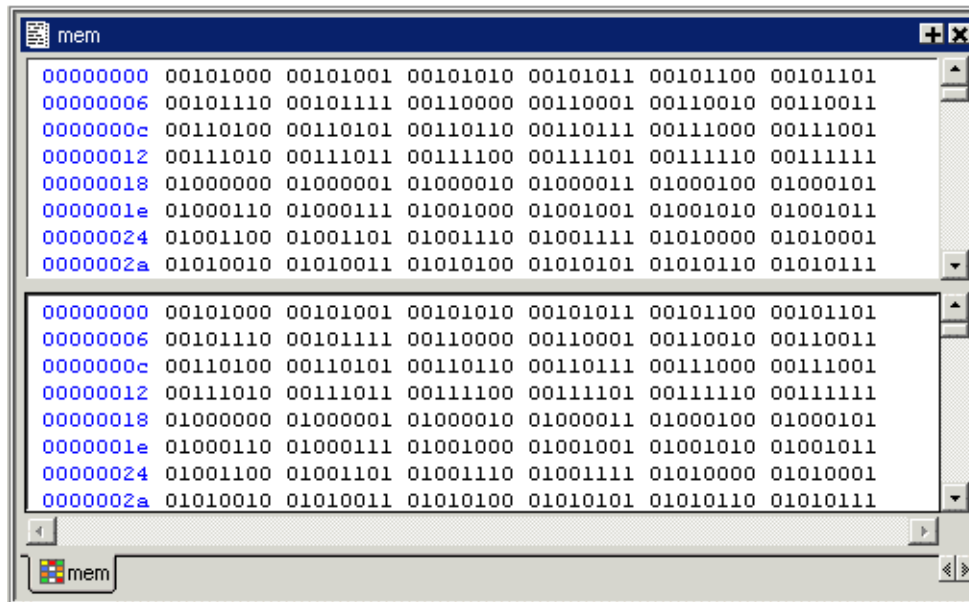
Memory List Window

To access:

- **View > Memory List**

- view memory list command

Use this window to view a list of all memories in your design.

Description

Single dimensional arrays of integers are interpreted as 2D memory arrays. In these cases, the word width listed in the Memory window is equal to the integer size, and the depth is the size of the array itself.

Memories with three or more dimensions display with a plus sign '+' next to their names in the Memory window. Click the '+' to show the array indices under that level. When you finally expand down to the 2D level, you can double-click the index, and the data for the selected 2D slice of the memory will appear in a memory contents window.

The simulator identifies certain kinds of arrays in various scopes as memories. Memory identification depends on the array element kind as well as the overall array kind (that is, associative array, unpacked array, and so forth.).

Table 95. Memory Identification

	VHDL	**Verilog/SystemVerilog**	**SystemC**
Element Kind[1]	• enum[2] • bit_vector • floating point type • std_logic_vector • std_ulogic_vector • integer type	any integral type (that is, integer_type): • shortint • int • longint • byte • bit (2 state) • logic • reg • integer • time (4 state) • packed_struct/packed_union (2 state) • packed_struct/packed_union (4 state)	• unsigned char • unsigned short • unsigned int • unsigned long • unsigned long long • char • short • int • float • double • enum • sc_bigint • sc_biguint • sc_int
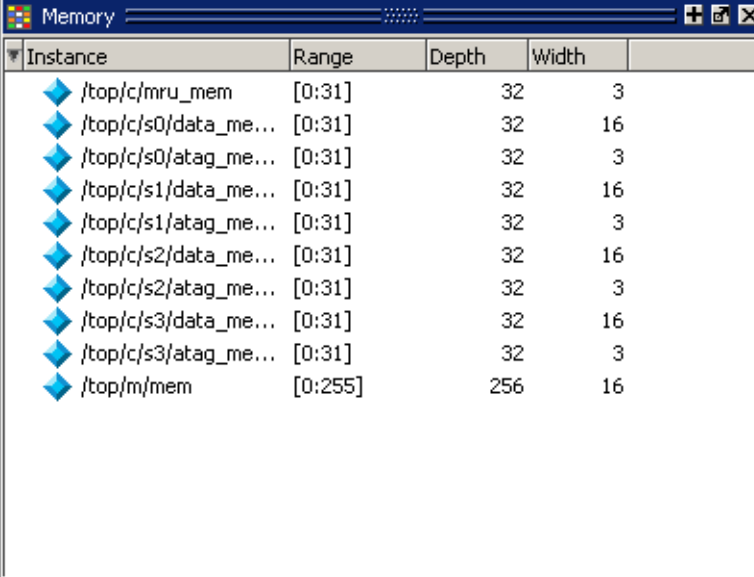
1. The element can be "bit" or "std_ulogic" if the array has dimensionality >= 2.
2. These enumerated types must have at least one enumeration literal that is not a character literal. The listed width is the number of entries in the enumerated type definition and the depth is the size of the array itself.

Table 95. Memory Identification (continued)

	VHDL	Verilog/SystemVerilog	SystemC
		• packed_array (single-Dim, multi-D, 2 state and 4 state) • enum • string	• sc_uint • sc_signed • sc_unsigned

Scope: Recognizable in	• architecture
	
	• process
	
	• record
		• module
		
		• interface
		
		• package
		
		• compilation unit
		
		• struct
		
		• static variables within a
		  • task
		  • function
		  • named block
		  • class
			• sc_module

Array Kind	• single-dimensional • multi-dimensional	• any combination of unpacked, dynamic and associative arrays[3] • real/shortreal • float	• single-dimensional • multi-dimensional

---

3. Any combination of unpacked, dynamic, and associative arrays is considered a memory, provided the leaf level of the data structure is a string or an integral type.

Figure 107. Memory List Window

Objects

Table 96. Memory List Window Columns

Column Title	Description
Instance	Hierarchical name of the memory
Range	Memory range
Depth	Memory depth
Width	Word width

Table 97. Memory List Popup Menu

Popup Menu Item	Description
View Contents	Opens a Memory Data window for the selected memory.
Memory Declaration	Opens a Source window to the file and line number where the memory is declared.
Compare Contents	Allows you to compare the selected memory against another memory in the design or an external file.
Import Data Patterns	Allows you to import data patterns into the selected memory through the Import Memory dialog box.

Table 97. Memory List Popup Menu (continued)

Popup Menu Item	Description
Export Data Patterns	Allows you to export data patterns from the selected memory through the Export Memory dialog box.

Table 98. Memories Menu

Popup Menu Item	Description
View Contents	Refer to items in the Table 97
Memory Declaration	
Compare Contents	
Import Data Patterns	
Export Data Patterns	
Expand Packed Memories	Toggle the expansion of packed memories.
Identify Memories Within Cells	Toggle the identification of memories within Verilog cells.
Show VHDL String as Memory	Toggle the identification of VHDL strings as memories.

Usage Notes

Viewing Packed Arrays

By default, packed dimensions are treated as single vectors in the Memory List window. To expand packed dimensions of packed arrays, choose **Memories > Expand Packed Memories**.

To change the permanent default, edit the PrefMemory(ExpandPackedMem) variable. This variable affects only packed arrays. If the variable is set to 1, the packed arrays are treated as unpacked arrays and are expanded along the packed dimensions such that they appear as a linearized bit vector. Refer to the section Setting GUI Preferences for details on setting preference variables.

Viewing Memory Contents

When you double-click an instance on the Memory List window, Questa SIM automatically displays a Memory Data window, where the name used on the tab is taken from the name of the instance, as seen in the Memory window. You can also enter the command add mem *<instance>* at the vsim command prompt.

Viewing Multiple Memory Instances

You can view multiple memory instances simultaneously. A Memory Data window appears for each instance you double-click in the Memory List window. When you open more than one window for the same memory, the name of the tab receives an numerical identifier after the name, such as "(2)".

Saving Memory Formats in a DO File

You can save all open memory instances and their formats (for example, address radix, data radix, and so forth) by creating a DO file.

Procedure

1. Select the Memory List window

2. Choose **File > Save Format**

   displays the Save Memory Format dialog box

3. Enter the file name in the "Save memory format" dialog box

   By default it is named *mem.do*. The file will contain all open memory instances and their formats.

   To load it at a later time, choose **File > Load**.

Saving Memories to the WLF File

By default, memories are not saved in the WLF file when you issue a "log -r /*" command. To get memories into the WLF file you will need to explicitly log them.

For example:

log /top/dut/i0/mem

If you want to use wildcards, then you will need to remove memories from the WildcardFilter list. To see what is currently in the WildcardFilter list, use the following command:

set WildcardFilter

If "Memories" is in the list, reissue the set WildcardFilter command with all items in the list except "Memories."

> **Note:**
> For post-process debug, you can add the memories into the Wave or List windows but the Memory List window is not available.

Questa SIM GUI Reference Manual

Message Viewer Window

To access:

- **View > Message Viewer** and select a loaded WLF dataset for viewing

- view msgviewer <dataset>.wlf command

Use this window to easily access, organize, and analyze any Note, Warning, Error or other elaboration and runtime messages written to the transcript during the simulation run.

Description

By default, the tool writes transcript messages during elaboration and runtime only to the transcript. To write messages to the WLF file (thus the Message Viewer window), use the -displaymsgmode and -msgmode options with the command to change the default behavior. By writing messages to the WLF file, the Message Viewer window is able to organize the messages for your analysis during the current simulation as well as during post simulation.

You can control what messages are available in the transcript, WLF file, or both with the following switches:

- displaymsgmode messages — User generated messages resulting from calls to Verilog Display System Tasks and PLI/FLI print function calls. By default, these messages are written only to the transcript, which means you cannot access them through the Message Viewer window. In many cases, these user generated messages are intended to be output as a group of transcript messages, thus the default of transcript only. The Message Viewer treats each message individually, therefore you could lose the context of these grouped messages by modifying the view or sort order of the Message Viewer.

  To change this default behavior you can use the -displaymsgmode argument with the vsim command. The syntax is:

  ```
  vsim -displaymsgmode {both | tran | wlf}
  ```

  You can also use the displaymsgmode variable in the *modelsim.ini* file.

  The message transcribing methods that are controlled by -displaymsgmode include:

  ◦ Verilog Display System Tasks — $write, $display, $monitor, and $strobe. The following also apply if they are sent to STDOUT: $fwrite, $fdisplay, $fmonitor, and $fstrobe.

  ◦ FLI Print Function Calls — mti_PrintFormatted and mti_PrintMessage.

  ◦ PLI Print Function Calls — io_printf and vpi_printf.

  ◦ SystemC Macros — SC_REPORT_INFO, SC_REPORT_WARNING, SC_REPORT_ERROR, and SC_REPORT_FATAL.

- msgmode messages — All elaboration and runtime messages not part of the displaymsgmode messages. By default, these messages are written only to the transcript. To change this default behavior you can use the -msgmode argument with the vsim command. The syntax is:

```
vsim -msgmode {both | tran | wlf}
```

To write messages to the WLF file and transcript, which provides access to the messages through the Message Viewer window, you can also use the msgmode variable (described in the User's Manual) in the *modelsim.ini* file.

Figure 108. Message Viewer Window
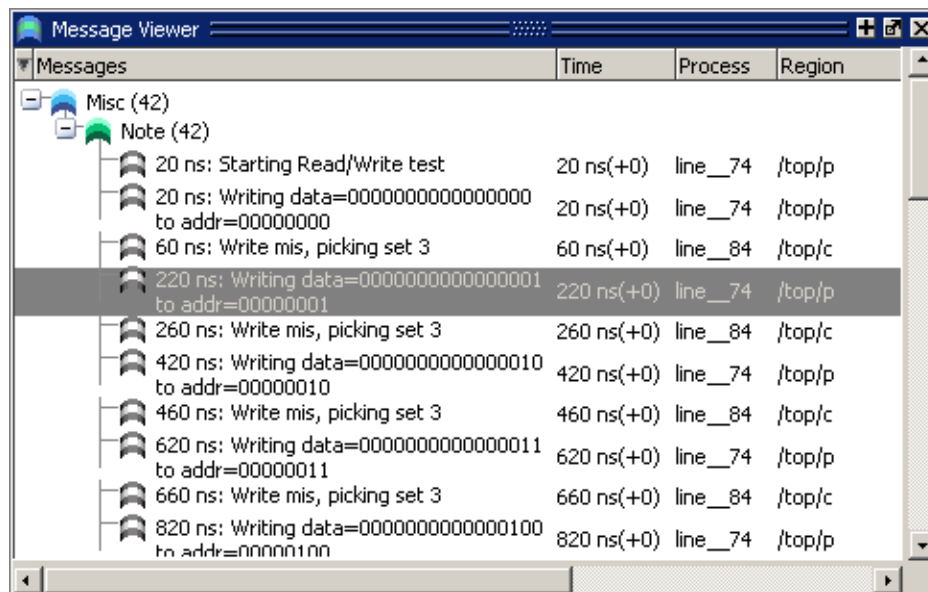
Note:
The Messages Bar in the Wave window provides indicators as to when a message occurred.

Message Viewer Window Tasks

Figure 109 and Table 99 provide an overview of the Message Viewer and several tasks you can perform.

Figure 109. Message Viewer Window — Tasks
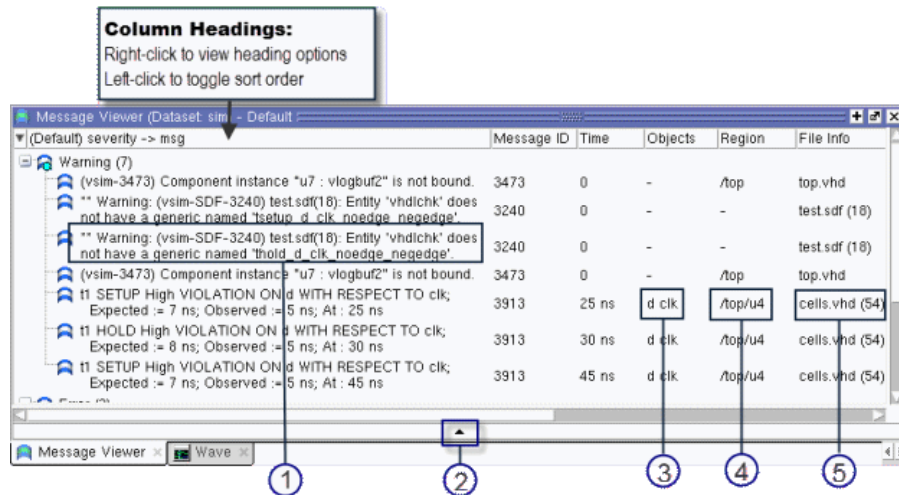
Table 99. Message Viewer Tasks

Icon	Task	Action
1	Display a detailed description of the message.	Right-click the message text then choose**View Verbose Message**.
1	Open message in Source window	Double-click the message in the Message Column.
1	Expand a hierarchical node	Double-click a non-leaf node in the Messages column.
2	Open the Configuration Options for Message Viewer dialog. Provides access to Analysis Questions, Column Layout, Filter Expression, Hierarchy Configuration and Sort Configuration.	Left-click the down arrow to toggle the "drawer" open and closed.
3	Open the source file and add a bookmark to the location of the object(s).	Double-click the object name(s).
3	Change the focus of the Structure and Objects windows.	Double-click the hierarchical reference.
4	Open the source file and set a marker at the line number.	Double-click the file name.

Objects

Table 100. Message Viewer Window Columns

Column	Description
Assertion Expression	Assertion expression associated with the message

Table 100. Message Viewer Window Columns (continued)

Column	Description
Assertion Filename	Name of the file where the assertion failure message originated
Assertion Line Number	Line number within the filename where the message originated
Assertion Name	Name of the assertion associated with the message
Assertion Start Time	Start time of the assertion associated with the message
Category	Keyword for the various categories of messages: • DISPLAY • FLI • PA • PLI • SDF • TCHK • VCD • VITAL • WLF • MISC • <user-defined>
Count	Date the test was run (in UCDB format)
File Info	Filename related to the cause of the message, and in some cases the line number in parentheses
Iteration/Delta	Iteration (delta) in which the message occurs
Message	Organized tree-structure of the sorted messages, as well as, when expanded, the text of the messages
Message ID	Message ID
Message ID Name	Message ID name
Objects	Object(s) related to the message, if any
Process	Process or leaf associated with the message
Region	Hierarchical region related to the message, if any
Severity	Message severity, such as Warning, Note or Error
Sim Time	Total simulation time

Table 100. Message Viewer Window Columns (continued)

Column	Description
sim Timeunits	Time unit used by the simulation
Source File Name	Name of the file where the message originated
Source File Number	Declaration number of the file associated with the message
Source Line Number	Line number within "filename" where the message originated
Time	Time of simulation when the message was issued
Timing Check Kind	Information about timing checks
Verbosity	Verbosity information from $messagelog system tasks. (Refer to $messagelog in the User's Manual for more information.)
VRM Context	Username under which the test was run
WLF Filename	Name of WLF file from which message was imported
WLF Raw Time	Simulation time (in ticks) associated with the message
WLF Time Unit	Simulation time unit

Table 101. Message Viewer Window Popup Menu

Popup Menu Item	Description
Reload Viewer Data	Opens a Source window for the file, and in some cases takes you to the associated line number.

View
• Verbose Message
• Message Source
• Log File
• Object Declarations
• Assertion Info
• ATV
• Change Environment
• Waveform:
  • Go to Time in Wave
  • Add Objs to Wave
	Opens selected item:
	Verbose Message dialog box with details about message
	Source code at line number where message is
	Log file, in a Source window
	Object window, to view declarations
	Assertion window, to view assertions
	ATV, to view assertion threads
	Change environment
	Waveform window, opens:
	at time of selected message
	adds objects associated with selected message

Analysis Questions	Opens Analysis Questions dialog box; used for saving and managing specific queries of the data.

Table 101. Message Viewer Window Popup Menu (continued)

Popup Menu Item	Description
Filter Expressions	Opens Filter Expressions dialog; used for saving and managing filters.
Hierarchy Configurations	Opens Hierarchy Configurations dialog box; used for saving and managing particular hierarchy configurations of the data.
Sort Configurations	Opens Sort Configurations dialog box, which allows you save and manage sorting for your configurations.
Column Layouts	Opens Configure Column Layout dialog; used for creating, editing and managing the configuration of columns.
Show Titles in Hier Column	Toggles on and off showing the titles within the hierarchy column
Global Options	Configures how/when Message Viewer opens.
Load/Save Setup File	Loads/Saves a particular setup to a name you specify.
Expand/Collapse Selected/All	Manipulates the expansion of the Messages column.

Usage Notes

Message Viewer Configuration of Data

The Message Viewer window contains a "drawer" of options for configuring the data, including analysis questions, column layouts, filter expressions, hierarchy of data, and sort configurations. The "drawer", where all these settings can be set in one convenient location, is opened with a small toggle button at the bottom of the window.

Figure 110. Configuration Options for Message Viewer
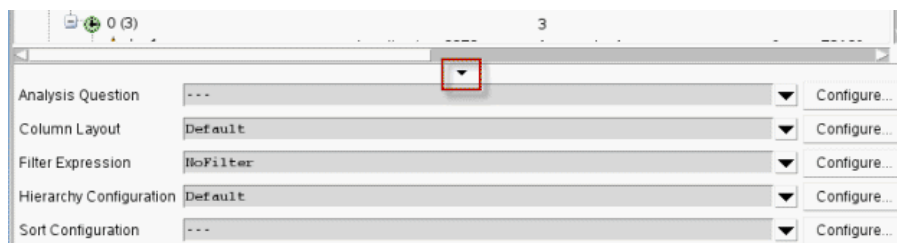
Custom Hierarchy Configurations

To save your own custom column layout and any filter settings to an external file (*<msgviewer>.do*), choose **File > Export > Hierarchy Configuration**while the window is active. You can reload these settings with the do command. This export does not retain changes to column width.

Filter Expressions Dialog Box

Filter Expressions Dialog Box

To access: With the Message Viewer active, select **Filter Expressions > [Configure Filters]**.

You can customize exactly which messages are shown in both the Message Viewer and the Results Analysis windows using the Filter Expressions dialog box.

Description

The Filter Expression dialog box in Figure 111 allows you to **Create** a new filter expression, **Edit** the selected filter expression, and **Remove**, **Copy**, **Rename**, or **Apply** a selected filter expression. The **Done** button closes the dialog box.

Figure 111. Filter Expressions Dialog Box
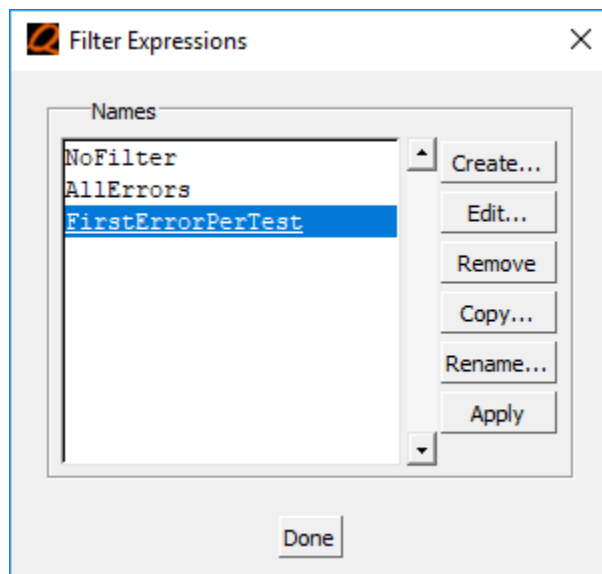
Click the **Create** button to open the **Create Filter Expression** dialog box shown in Figure 112.

Figure 112. Create a Filter Expression
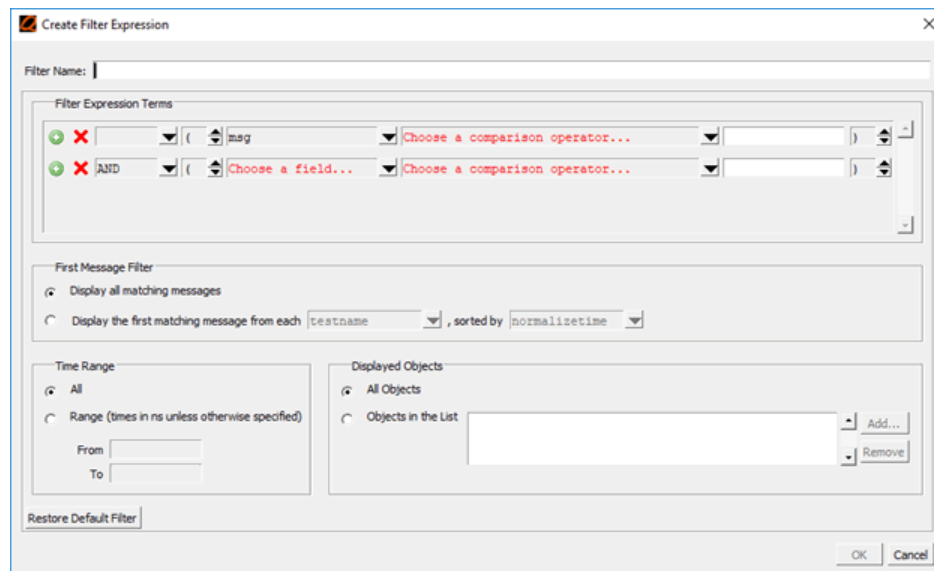
Objects

- Filter Name field

  You must enter a unique filter name before a new filter expression can be saved.

- Filter Expression Terms area

  Create filter rules that specify which messages are shown in the windows. From left to right, each filter rule is made up of the following:

  - **Add** and **Remove** buttons ⊕ ✗ — either add a rule filter row below the current row or remove that rule filter row.

  - **Logic field** — specifies a logical argument for combining adjacent rules. Your choices are: AND, OR, AND NOT, and OR NOT.

  - **Open Parenthesis field** — controls rule groupings by specifying, if necessary, any open parentheses. The up and down arrows increase or decrease the number of parentheses in the field.

  - **Column field** — specifies that your filter value applies to a specific column of the Message Viewer.

  - **Comparison operator field** — select from a drop-down list of comparison operators - includes Numeric, Text, and SQL comparison operators.

    - For numeric- and time-based filter values your choices are: ==, !=, <, <=, >, and >=.

    - For text-based filter values, your choices are: Contains, Doesn't Contain, or Exact.

    - For SQL filter values, your choices are: Like, Glob, Is, Is Not, In, and Not In.

◦ **Filter Value field** — specifies the filter value associated with your filter rule.

◦ **Closed Parenthesis field** — controls rule groupings by specifying, if necessary, any closed parentheses. The up and down arrows increase or decrease the number of parentheses in the field.

• First Message Filter area

Allows you to control the appearance of either all matching messages or just the first matching message (with further filtering options).

• Time Range area

Allows you to filter which messages appear according to simulation time. The default is to display messages for the complete simulation time.

• Displayed Objects area

Allows you to filter which messages appear according to the values in the Objects column. The default is to display all messages, regardless of the values in the Objects column. The Objects in the list text entry box allows you to specify filter strings, where each string must be on a new line.

Objects Window

To access:

- **View > Objects**

- view objects command

- Wave window > View Objects Window Button

Use this window to view the names and current values of declared data objects in the current region, as selected in the Structure window.

Description

This section describes GUI elements specific to this Window.

Current Time Label — Displays the Current Time or the Now (end of simulation) time. This is the time used to control state values annotated in the window. (For details, refer to Current Time Label.)

Viewable data objects include:

- signals

- nets

- registers

- constants and variables not declared in a process

- generics

- parameters

- transactions

- SystemC member data variables

- Questa Verification IP objects. Refer to Questa Verification IP Objects in the GUI in the User's Manual for more information.

Figure 113. Objects Window
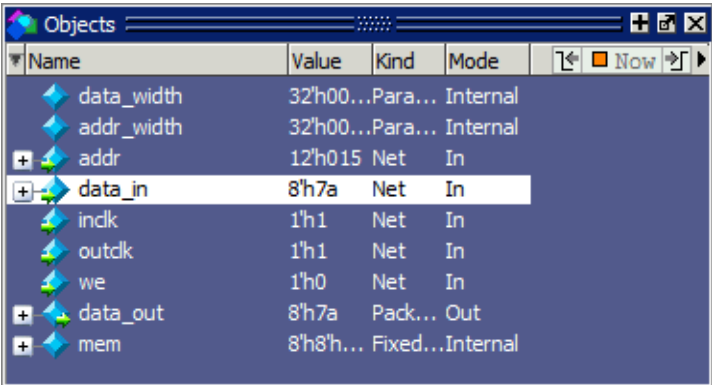
The Object filter toolbar provides filtering of design objects appearing in the Objects window.

Figure 114. Object Window Toolbar
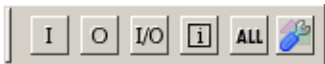
Objects

Table 102. Columns in the Objects Window

Column name	Description
Name	the name of each object in the current region
Value	the current value of each object
Kind	the object type
Mode	the object mode (internal, in, out, and so forth.)
PAInfo	When applicable, labels an object as related to isolation (ISO) or level-shifters (LS).
1H -> 0L	the number of times each object has transitioned from a 1 or a High state to a 0 or a Low state
0L -> 1H	the number of times each object has transitioned from a 0 or a Low state to 1 or a High state
0L -> Z	the number of times each object has transitioned from a 0 or a Low state to a high impedance (Z) state
Z -> 0L	the number of times each object has transitioned from a high impedance state to a 0 or a Low state
1H -> Z	the number of times each object has transitioned from a 1 or a High state to a high impedance state

Table 102. Columns in the Objects Window (continued)

Column name	Description
Z -> 1H	the number of times each object has transitioned from a high impedance state to 1 or a High state
State Count	the number of values a state machine variable can have
State Hits	the number of state machine variable values that have been hit
State %	the current ration of State Hits to State Count
# Nodes	the number of scalar bits in each object
# Toggled	the number of nodes that have transitioned at least once. A signal is considered toggled if and only if: • it has 0- >1 and 1->0 transitions and NO Z transitions, or • if there are ANY Z transitions, it must have ALL four of the Z transitions. Otherwise, the counts are place in % 01 or % Z columns. For more specifics on what is considered toggled, refer to Toggle Countsin the User's Manual.
% Toggled	the current ratio of the # Toggled to the # Nodes for each object
% 01	the percentage of 1H -> 0L and 0L -> 1H transitions that have occurred (transitions in the first two columns)
% Full	the percentage of all transitions that have occurred (all six columns)
% Z	the percentage of 0L -> Z, Z -> 0L, 1H -> Z, and Z -> 1H transitions that have occurred (last four columns)

Table 103. Objects Window Popup Menu

Popup Menu Item	Description
View Declaration	Opens a Source window to the declaration of the object
View Memory Contents	Opens a Memory Data window to display memory contents of selected object
Add Wave	Adds the selected object(s) to the Wave window
Add Wave New	Creates a new instance of the Wave window and adds the selected object(s) to that window.
Add Wave To	Opens a drop down list of Wave windows when multiple windows exist. Adds the selected object(s) to the selected Wave window.
Add Dataflow	Adds the selected object(s) to a Dataflow window

Table 103. Objects Window Popup Menu (continued)

Popup Menu Item	Description
Add to	Add the selected object(s) to any one of the following: Wave window, List window, Log file, Schematic window, Dataflow window. You may choose to add only the Selected Signals, all Signals in Region, all Signals in Design.
UPF	Allows you to view additional information about UPF objects: • View — Opens the source file for viewing UPF information regarding the object. • Add to wave — Adds UPF-based information related to the object to the Wave window.
Event Traceback	Enables Causality Traceback • Show Cause • Show Driver • Show Root Cause • Show 'X' Cause (ChaseX)
Copy	Copies information about the object to the clipboard
Find	Opens the Find box
Insert Breakpoint	Adds a breakpoint for the selected object
Toggle Coverage	Control toggle coverage of the selected object(s). Submenus allow the following options: • Add - add to toggle coverage • Extended - include as extended toggle coverage • Enable - enable toggle coverage • Disable - disable toggle coverage • Reset - reset toggle coverage
Modify	Modify the selected object(s) by selecting one of the following from the submenu: • Force - opens Force Selected Signal dialog • Remove Force - remove effect of force command • Change Value - change value of selected • Apply Clock - opens Define Clock dialog • Apply Wave - opens Create Pattern Wizard
Radix	Opens Signal Radix dialog, allowing you to set the radix of selected signal(s) in all windows

Table 103. Objects Window Popup Menu (continued)

Popup Menu Item	Description
Show	Shows list of port types and object kinds that are displayed. Includes a Change Filter selection that opens the Filter Objects dialog, which allows you to filter the display.

Table 104. Object Window Toolbar Buttons

Button	Name	Shortcuts	Description
I	View Inputs Only	None	Changes the view of the Objects window to show inputs.
O	View Outputs Only	None	Changes the view of the Objects Window to show outputs.
I/O	View Inouts Only	None	Changes the view of the Objects Window to show inouts.
i	Vies Internal Signals	None	Changes the view of the Objects Window to show Internal Signals.
ALL	Reset All Filters	None	Clears the filtering of Objects Window entries and displays all objects.
	Change Filter	None	Opens the Filter Objects dialog box.

Objects Window Tasks

This section describes tasks for using the Objects window.

Interacting with Other Windows

You can interact with other windows from the Objects window.

Procedure

1. Click an entry in the window to highlight that object in the Dataflow, Schematic, and Wave windows.

2. Double-click an entry to highlight that object in a Source window

Setting Signal Radix

You can set the signal radix for a selected signal or signals in the Objects window as follows.

Procedure

1. Click (LMB) a signal to select it or use Ctrl-Click Shift-Click to select a group of signals.

2. Choose **Objects > Radix** from the menu bar; or right-click the selected signal(s) and choose **Radix** from the popup menu.

This opens the Signal Radix dialog box (Figure 115), where you may select a radix. This sets the radix for the selected signal(s) in the Objects window and every other window where the signal appears.

Figure 115. Setting the Global Signal Radix from the Objects Window
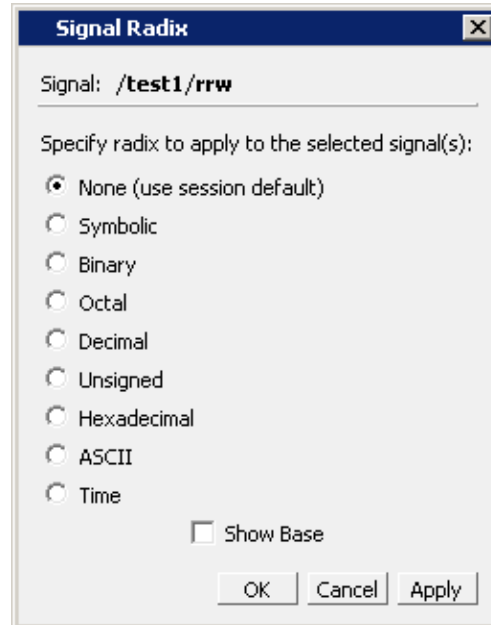
Finding Contents of the Objects Window

You can filter the contents of the Objects window by either the Name or Value columns.

Procedure

1. Ctrl-F to display the Find box at the bottom of the window.

2. Click the "Search For" button and select the column to filter on.

3. Enter a string in the Find text box

4. Enter

Filtering Contents of the Objects Window

You can filter the contents of the Objects window by the Name column.

Procedure

1. Ctrl-F to display the Find box at the bottom of the window.

2. Ctrl-M to change to "Contains" mode.

3. Enter a string in the Contains text box

The filtering will occur as you begin typing. You can disable this feature with Ctrl-T.

4. Filters are stored relative to the region selected in the Structure window. If you re-select a region that had a filter applied, that filter is restored. This allows you to apply different filters to different regions.
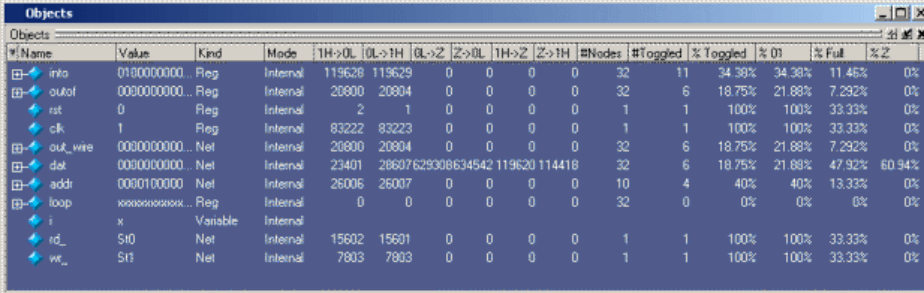
Filtering by Signal Type

The **View > Filter** menu selection allows you to specify which signal types to display in the Objects window. Multiple options can be selected. Choose **Change Filter** to open the Filter Objects dialog box, where you can select port modes and object types to be displayed.

Viewing Toggle Coverage in the Objects Window

Toggle coverage data can be displayed in the Objects window in multiple columns, as shown in the following figure. Right-click the column title bar and select Show All Columns to make sure all Toggle coverage columns are displayed. There is a column for each transition type.

Figure 116. Objects Window - Toggle Coverage

Pending Exclusions Window

To access:

- **View > Coverage > Pending Exclusions**

- view pending command

The Pending Exclusions window contains a list of exclusion commands that have not yet been executed.

Description

The Pending Exclusions window displays a list of all exclusion commands added while two-step exclusions are enabled, which are stored awaiting execution. You can:

- Save this list to a file.

- Add to the list by loading a file.

- Execute or remove selected exclusion commands with window buttons or the Pending Exclusions window popup menu.

Figure 117. Pending Exclusions Window
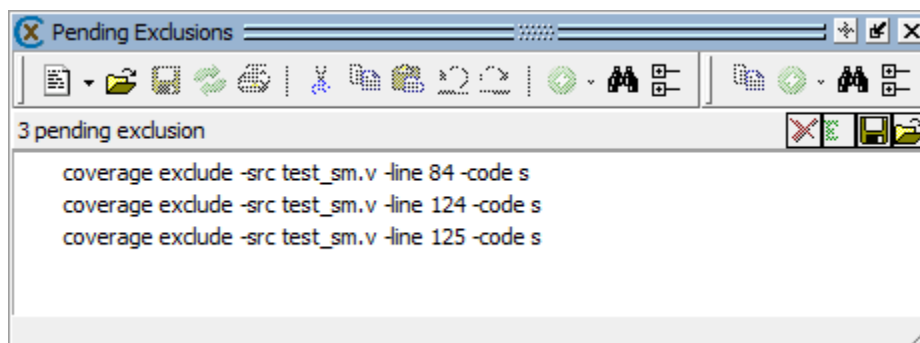
Objects

Table 105. Pending Exclusions Window Popup Menu

Popup Menu Item	Description
Load Exclusion File	Opens a dialog box for loading exclusion files.
Save Exclusion File	Opens a dialog box for saving an exclusion file.
Filter	Opens a dialog box for applying and creating filters.
Apply	Applies selected exclusion commands.

Table 105. Pending Exclusions Window Popup Menu (continued)

Popup Menu Item	Description
Apply All	Applies all exclusion commands within the window.
Clear	Clears selected exclusion commands from the window.
Clear All	Clears all exclusion commands from the window.

Usage Notes

The Pending Exclusions window remains visible while two-step exclusions are disabled, however it is not possible to remove or run exclusions from the pending list unless two-step exclusions are enabled.

Processes Window

To access:

- **View > Process**

- view process command

Use this window to view a list of HDL and SystemC processes in one of four viewing modes. In addition, the data in this window will change as you run your simulation and processes change states or become inactive.

Description

The four viewing modes are as follows:
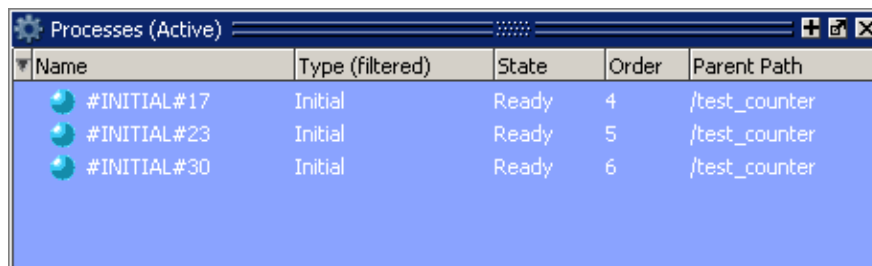
Active — (default) active processes in your simulation.

In Region — process in the selected region.

Design — intended for primary navigation of ESL (Electronic System Level) designs where processes are a foremost consideration.

Hierarchy — a tree view of any SystemVerilog nested fork-joins.

Figure 118. Processes Window

Objects

Table 106. Processes Window Column Descriptions

Column Title	Description
Name	Name of the process.
Class Info	SystemVerilog class object id or UVM component name.
Order	Displays the execution order of all processes in the Active and Ready states in the active kernel queue. Processes that are not in the Active or Ready states do not yet have any order, in which case the column displays a dash (-). The Process window updates the execution order automatically as simulation proceeds.
Parent Path	Hierarchical parent pathname of the process

Table 106. Processes Window Column Descriptions (continued)

Column Title	Description
State	Process state. • Idle — Indicates an inactive SystemC Method, or a process that has never been active. The Idle state will occur only for SystemC processes or methods. It will never occur for HDL processes. • Wait — Indicates the process is waiting for a wake up trigger (change in VHDL signal, Verilog net, SystemC signal, or a time period). • Ready — Indicates the process is scheduled to be executed in current simulation phase (or in active simulation queue) of current delta cycle. • Active – Indicates the process is currently active and being executed. • Queued — Indicates the process is scheduled to be executed in current delta cycle, but not in current simulation phase (or in active simulation queue). • Done — Indicates the process has been terminated, and will never restart during current simulation run. Processes in the Idle and Wait states are distinguished as follows. Idle processes(except for ScMethods) have never been executed before in the simulation, and therefore have never been suspended. Idle processes will become Active, Ready, or Queued when a trigger occurs. A process in the Wait state has been executed before but has been suspended, and is now waiting for a trigger. SystemC methods can have one of the four states: Active, Ready, Idle or Queued. When ScMethods are not being executed (Active), or scheduled (Ready or Queued), they are inactive (Idle). ScMethods execute in 0 time, whenever they get triggered. They are never suspended or terminated.
Type	Process type, according to the language, including the following types: • Always • Assign • Final • Fork-Join (dynamic process like fork-join, sc_spawn, and so forth.) • Initial • Implicit (internal processes created by simulator like Implicit wires, and so forth.) • Primitive (UDP, Gates, and so forth.) • ScMethod • ScThread (SC Thread and SC CThread processes) • VHDL Process

Profiling Windows

To access:

- **View > Profiling > Call Tree Profile**

- view calltree command, view du command, view ranked command, view structural command, view profiledetails command

Use the profiling windows to view performance or memory profiling information about your simulation.

Description

The following are the five profiling windows:

Calltree — Displays information in a hierarchical form that indicates the call order dependencies of functions or routines.

Design Units — Displays information aggregated for the different design units.

Ranked — Displays information for each function or instance.

Structural — Displays information aggregated for different instances.

Profile Details — Displays detailed profiling information based on selections in the other Profile Windows

---
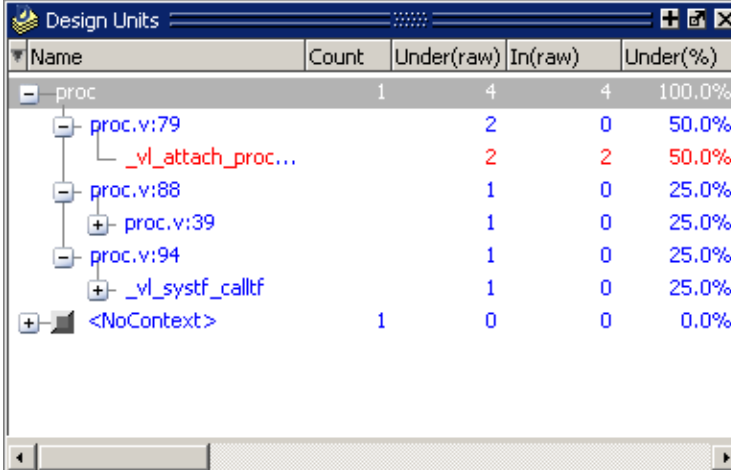
Note:
You must have enabled performance or memory profiling. Refer to Classic Performance Profiler in the User's Manual for more information.

Figure 119. Profile Calltree Window

Figure 120. Profile Design Unit Window
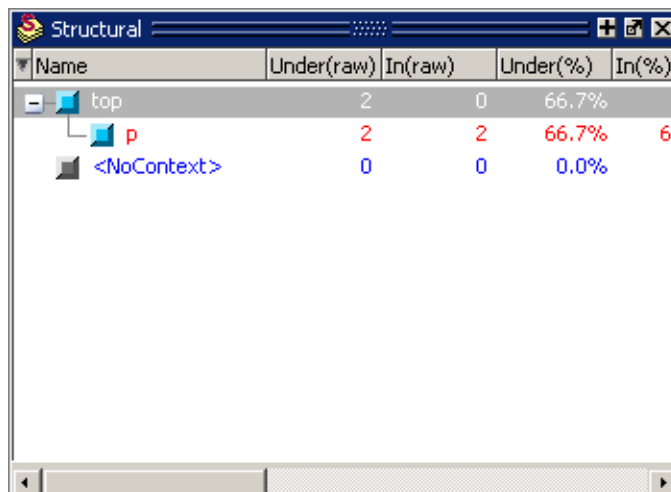
Name	Count	Under(raw)	In(raw)	Under(%)
⊟ proc	1	4	4	100.0%
⊟ proc.v:79		2	0	50.0%
└ _vl_attach_proc...		2	2	50.0%
⊟ proc.v:88		1	0	25.0%
⊞ proc.v:39		1	0	25.0%
⊟ proc.v:94		1	0	25.0%
⊞ _vl_systf_calltf		1	0	25.0%
⊞ <NoContext>	1	0	0	0.0%

Figure 121. Profile Ranked Window
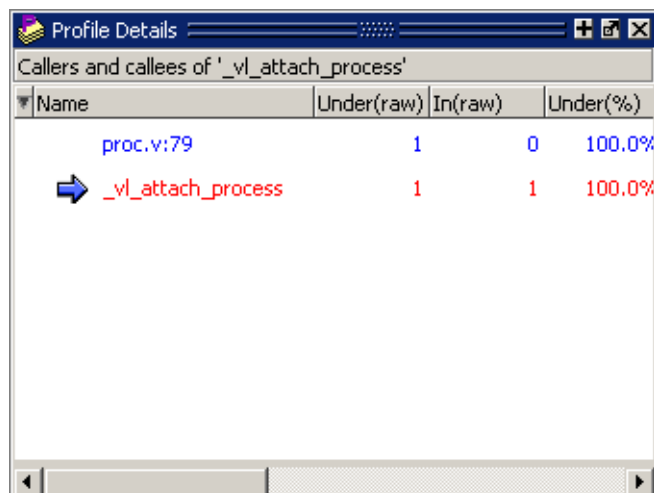
Name	Under(raw)	In(raw)	Under(%)	In(%)
_vl_attach_process	1	1	33.3%	33.3%
_vl_systf_calltf	1	1	33.3%	33.3%
proc.v:79	1	0	33.3%	0.0%
proc.v:94	1	0	33.3%	0.0%
<NoCallStack>	0	0	0.0%	0.0%

Figure 122. Profile Structural Window

Figure 123. Profile Details Window

Objects

Table 107. Profile Calltree Window Column Descriptions

Column Title	Description
%Parent	lists the ratio, as a percentage, of the samples collected during the execution of a function or instance to the samples collected in the parent function or instance. (Not available in the Profile Ranked window.)
Count	lists the number of instances of the design unit detected. (Only available in the Profile Design Unit window.)
In%	lists the ratio (as a percentage) of the total samples collected during a function or instance.

Table 107. Profile Calltree Window Column Descriptions (continued)

Column Title	Description
In(raw)	lists the raw number of Profiler samples collected during a function or instance.
MemIn	lists the amount of memory allocated to a function or instance.
MemIn(%)	lists the ratio (as a percentage) of the amount of memory allocated to a function or instance to the total memory available.
MemUnder	lists the amount of memory allocated to a function, including all support routines under that function; or, the amount of memory allocated to an instance, including all instances beneath it in the structural hierarchy.
MemUnder(%)	lists the ratio (as a percentage) of the amount of memory allocated to a function and all of its support routines to the total memory available; or, the ratio of the amount of memory allocated to an instance, including all instances beneath it in the structural hierarchy, to the total memory available.
Name	lists the parts of the design for which profiling information was captured.
sum(MemIn(%))	lists the ratio of the cumulative memory allocated. (Only available in the Profile Ranked and Profile Design Unit windows.)
sum(MemIn)	lists the cumulative memory allocated. (Only available in the Profile Ranked and Profile Design Unit windows.)
Under (raw)	lists the raw number of Profiler samples collected during the execution of a function, including all support routines under that function; or, the number of samples collected for an instance, including all instances beneath it in the structural hierarchy.
Under(%)	lists the ratio (as a percentage) of the samples collected during the execution of a function and all support routines under that function to the total number of samples collected; or, the ratio of the samples collected during an instance, including all instances beneath it in the structural hierarchy, to the total number of samples collected.

Schematic Window

To access:

- **View > Schematic** or **Add > To Schematic**

- view schematic command or add schematic [-incr | -full] <design_unit_name>

The Schematic window provides two views of the design — a Full View, which is a structural overview of the design; and an Incremental View, which uses Click-and-Sprout actions to incrementally add to the selected net's fanout. The Incremental view displays the logical gate equivalent of the RTL portion of the design, making it easier to understand the intent of the design.

Description

> **Note:**
> The **Add To > Schematic** menu options change dynamically based on which debug window is currently active.

> **Note:**
> If -incr or -full is not provided with the add schematic command, the design unit will be added to the currently active Schematic view mode.

Figure 124. Schematic Window
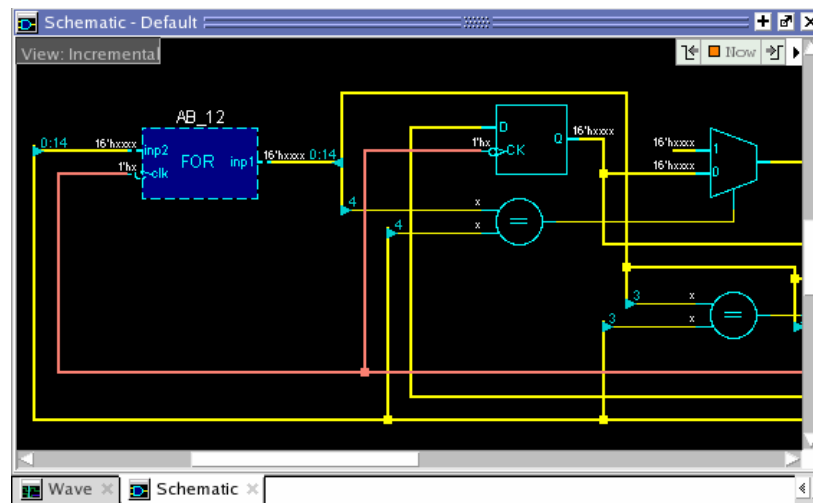
To create the necessary data to display the schematic you must:

- Use the +acc switch with the vopt command to provide accessibility into the design for debugging; and use the -debugdb switch with the vopt command to collect combinatorial and sequential logic data into the work library

> **Note:**
> The +acc option supports selective visibility into your design in order to reduce the size of the debugging database. For example, if your testbench has an instance called "instDut" of the design under test, you can use vopt -debugdb +acc+'/instDut' to generate a debug database for only that instance.

- Use the -debugdb switch with the vsim command to create the debug database

- Log the results

For example, if you have a Verilog design with a top level module named top.v you would do the following:

```
vlib work
vlog *.v
vopt +acc top -o top_opt -debugdb
vsim -debugdb top_opt
log -r /*
```

A View: indicator is displayed in the top left corner of the window (Figure 125). You can toggle back and forth between views by simply clicking this View: indicator.

Figure 125. Schematic View Indicator
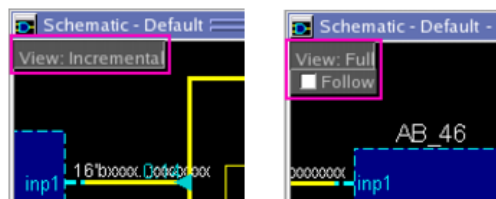
The Incremental View is ideal for design debugging. It allows you to explore design connectivity by tracing signal readers/drivers to determine where and why signals change values at various times.

The Full View is a more static view that can be dynamically linked to your selections in other windows with the "Follow" selection.

Objects

Table 108. Schematic Window Popup Menu

Popup Menu Item	Selection	Description
View Selection	Declaration	Opens a Source window to the line of code where the object is declared.
	Instantiation	Applies only to modules. Opens a Source window to the line of code where the module is instantiated.
	FSM Viewer	Displays the selected item in the FSM Viewer.
	Wave Pane	

Table 108. Schematic Window Popup Menu (continued)

Popup Menu Item	Selection	Description
	This Window	Displays selected signals in embedded Wave viewer
	New Window	Erase contents of current Incremental view and redraw only selected object
		Redraw selected object(s) in Incremental View of new Schematic window
Zoom	Zoom In	Zoom in 2x
	Zoom Out	Zoom out 2x
	Zoom Full	Zoom so all objects fits into display,
	Zoom Selected	Zoom into the selected object
	Zoom Highlight	Zoom into the highlighted object.
Fold/Unfold		Hides or shows the contents of selected instance or abstract block. Displays folded instance or abstract block as a solid blue box with dashed border.
Expand Net To	Drivers	Displays all drivers, readers, fanins, fanouts, design inputs and hierarchy inputs of the selected net.
	Readers	
	Drivers & Readers Fanins	
	Fanouts	
	Fanins & Fanouts	
	Design Inputs	
	Hierarchy Inputs	
Event Traceback	Show Driver	Traces to the immediate driving process of the selected signal at the current time
	Show Cause	Traces to the first sequential process that drives the selected signal at the current time
	Show Root Cause	
	Show 'X' Cause (ChaseX)	Traces to the root cause of the selected signal at the current time
	Show All Possible Drivers	Traces to the root of an unknown value (X)
	View Path Times	Highlights drivers in the Source window and lists possible drivers
	Trace to Driving Event	
	Start Trace from End Time	
	Start Trace from Begin Time	
Highlight	Add	Highlights any selected objects with color you select
	Remove	Removes highlight color from selected objects
	Remove All	Removes all highlight colors from display

Table 108. Schematic Window Popup Menu (continued)

Popup Menu Item	Selection	Description
Add	Add All Signals to Wave	Adds all signals in schematic to the Wave window
	Add to Wave	Adds selected signal to Wave window
	Add to Schematic	Adds selected signal to Current window or New window
	Add to Dataflow	Adds selected signal to Dataflow window
	Add to List	Adds selected signal to List window
	Add to Watch	Adds selected signal to Watch window
Edit	Global Signal Radix	Changes radix for all displayed signals in Schematic window
	Undo	Undo previous action
	Redo	Redo undone action
	Cut	Cut selected
	Copy	Copy selected
	Paste	Paste selected
	Delete	Delete selected
	Select Highlighted	Make highlighted objects the selected objects
	Select All	Select all objects in display
	Unselect All	Unselect all objects in display
	Regenerate Layout	Regenerates display to improve layout
	Delete All	Deletes everything from Schematic window
Find		Opens the Search Bar (at bottom of window) in the Find mode to make searching for objects easier, especially with large designs.
Save		Saves everything in Schematic, including sticky notes to a *.sch* file
Restore	Current Window	Restores saved *.sch* file to the current window or to a new window. If Current Window is selected, the saved *.sch* file overwrites existing information.
	New Window	
Sticky Note	Add	Adds sticky note (annotation) to selected net or component
	Remove	Removes sticky note from selected item
	Hide/Unhide	Hides or unhides sticky note for selecte item
	Hide/Unhide All	Hides or unhides all sticky notes
Show	Unconnected Pins	Shows or hides unconnected pins in folded instances
	Instance Names	Display instance names when checked
	Net Names	Display net names when checked
	Net Rip Index	Display bus ripper indices when checked

Table 108. Schematic Window Popup Menu (continued)

Popup Menu Item	Selection	Description
	Pin Names	Display pin names when checked
	Design Unit Names	Display design unit names when checked
	Signal Values	Display signal values when checked
	Show All	Display all of the above
	Hide All	Hide all of the above

Usage Notes

Adding Objects to the Incremental View

You can use any of the following methods to add objects to the Incremental View of the Schematic window:

- Drag and drop objects from other windows. Both nets and instances may be dragged and dropped. Dragging an instance will result in the addition of all nets connected to that instance. When an object is dragged and dropped into the Incremental view, the add schematic command will be reflected in the Transcript window.

- Use the **Add to > Schematic** menu options:

Selected Signals — Display selected signals

Signals in Region — Display all signals from the current region.

Signals in Design — Clear the window and display all signals from the entire design.

- Select the object(s) you want placed in the Schematic Window, then click-and-hold the Add Selected to Window Button in the Standard Toolbar and select **Add to Schematic**.

- Use the add schematic command.

Navigating in the Schematic Window

You can use the mouse to navigate and select items within the Schematic window. The following descriptions are based on the Select mouse mode (as set by the **Schematic > Mouse Mode** menu).

- Strokes with the **middle mouse button**:

Up — Move up in the hierarchy (does nothing when you are already at the top level)

Down — Move down in the hierarchy to the selected instance or the instance from which you began the stroke.

Up/Left — Zoom full.

Down/Left — Zoom in on any selected items.

Up/Right — Zoom out. The factor of the zoom changes depending on the length of the stroke.

Down/Right — Zoom area. The box indicates your desired zoom view.

- Zoom in and out with the **mouse scroll wheel**.

  The view will zoom and center on your mouse location.

- Selecting objects with the **left mouse button**:

  Your selection is also reflected in the Structure, Objects, and Wave windows.

  Single click — highlights selected objects.

  Double click — opens a Code Preview window with the source code of the selected item highlighted.
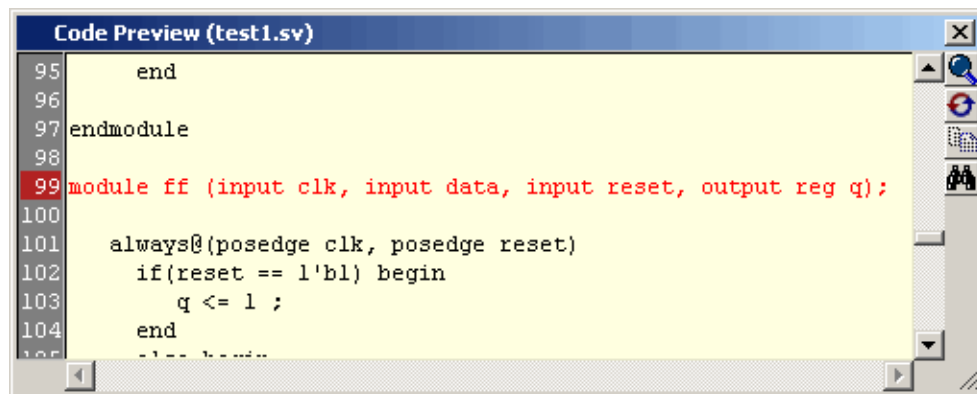
  Click and drag — selects all objects within the bounding box.

  Shift-click — highlights multiple selected objects.

The alternate mouse modes affect the above mouse controls as follows:

- **Schematic > Mouse Mode > Zoom Mode**

Left mouse button — Single click selects items, click and drag performs zoom actions.

Middle mouse button — click and drag pans the schematic view.

- **Schematic > Mouse Mode > Pad Mode**

Left mouse button — Single click selects items, click and drag pans the schematic view.

Middle mouse button — click and drag performs zoom actions.

Incremental Schematic Options Dialog Box

To access: When the Incremental view is active: **Schematic > Preferences**

The schematic window provides additional control over the amount and type of information displayed, depending on whether you are using the Incremental View or the Full View. Display options are available when the Schematic window is active.

Description

The dialog box is the same when the Full view is active and **Schematic > Preferences** is chosen, only the title is changed.

Figure 126. Incremental Schematic Options Dialog Box
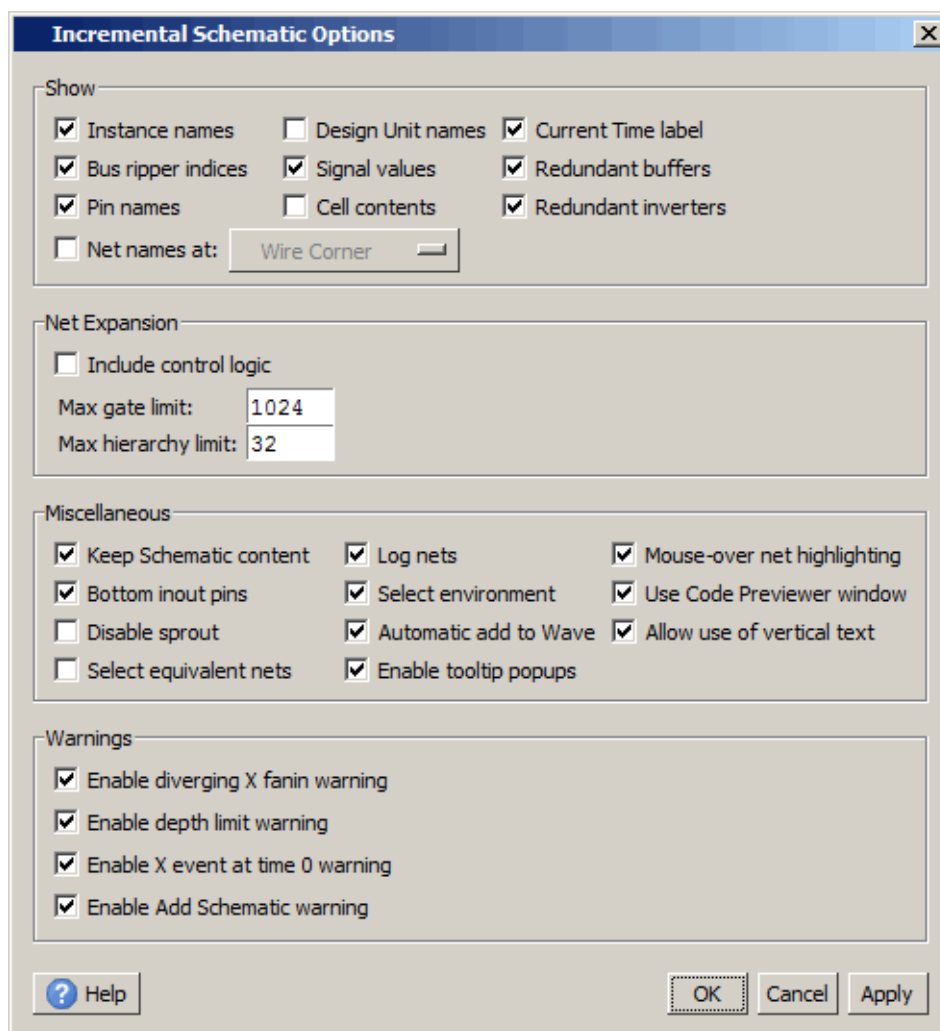
Note:
Changing the Hierarchy option causes the Schematic window to be erased.

Click the Information button at the bottom left corner to get option descriptions when you mouse over the

option.

Objects

- Dialog box elements.

  - The **Show** section of the dialog box:

    - **Instance names** — Show instance names for architectures, modules, processes, and so on.

    - **Bus ripper indices** — Show ripper indices for busses.

    - **Pin names** — Show pin names in the architectures, modules, processes, and so on.

    - **Design Unit names** — Show design unit names for architectures, modules, processes, and so on.

    - **Signal values** — Show signal values annotated onto the nets. Signal values are based on the "current time"– which is set by the active cursor in the Wave window or the Current Time Label in the Source or Schematic windows.

    - **Cell contents** — Show the internals of a library cell (`celldefine or VITAL).

    - **Current Time Label** — Displays the Current Time or the Now (end of simulation) time. This is the time used to control state values annotated in the window. (For details, see Current Time Label.)

Figure 127. Current Time Label in the Schematic Window
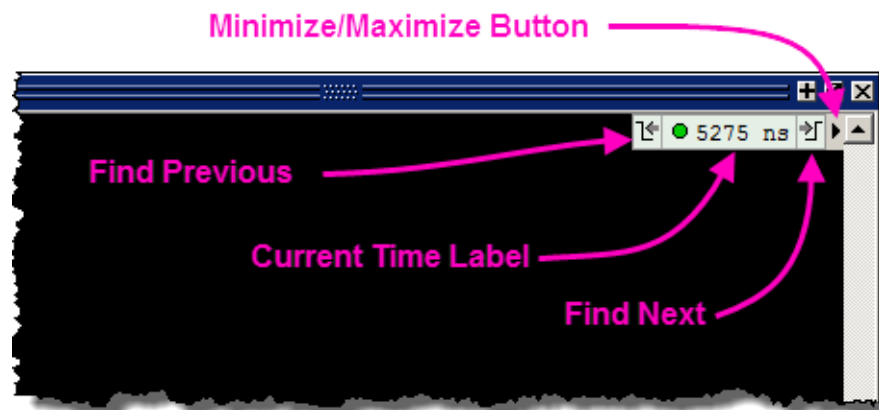
  - **Redundant buffers** — Display redundant buffers.

- ◦ **Redundant inverters** — Displays redundant inverters.

- ◦ **Net Names** — Show all net names:

  - • **Wire corner** — Displays net names at or near wire corners.

  - • **Window Edge**— Displays net names only if net extends past the edge of the window.

- ◦ The **Net Expansion** section of the dialog controls whether control logic is followed when doing an Expand To Fanin/Fanout:

  - • **Max gate limit** — Specifies the maximum number of gates the fanin/fanout should go through before stopping. The default value is 1024.

  - • **Max hierarchy limit** — specifies the maximum number of hierarchy levels the fanin/fanout should go through before stopping. The default value is 32.

Source Window

To access:

- **File > Open**

- Click the **Open** icon

- Double-click objects in other windows

- edit command

The Source window allows you to view and edit source files as well as set breakpoints, and step through design files.

Description

By default, the Source window displays your source code with line numbers. You may also see the different graphic elements.

Also, you will see various code coverage indicator icons (see Coverage Data in the Source Window for details).

Figure 128. Source Window
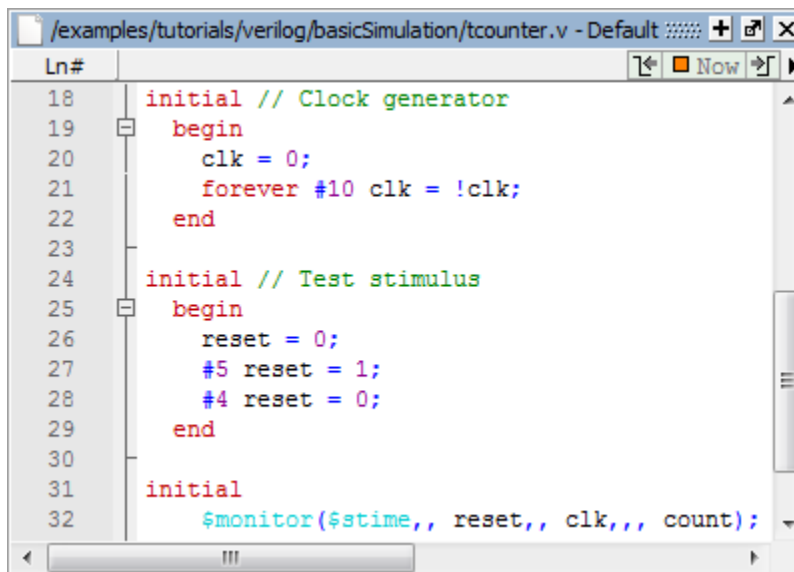
If you double-click an item in the Objects window or in the structure tab (**sim** tab), the underlying source file for the object will open in the Source window and scroll to the line where the object is defined.

By default, files you open from within the design (such as when you double-click an object in the Objects window) open in Read Only mode. To make the file editable, right-click in the Source window and select (uncheck) Read Only. To change this default behavior, set the PrefSource(ReadOnly) variable to 0. Refer to Setting GUI Preferences for details on setting preference variables.

Objects

- Graphical elements.

  - ◦ **Red line numbers** — denote executable lines, where you can set a breakpoint

  - ◦ **Blue arrow** — denotes the currently active line or a process that you have selected in the Processes Window

  - ◦ **Red ball in line number column** — denotes file-line breakpoints; gray ball denotes breakpoints that are currently disabled

  - ◦ **Blue flag in line number column** — denotes line bookmarks

  - ◦ **Current Time Label** — displays the Current Time or the Now (end of simulation) time. This is the time used to control state values annotated in the window.

  - ◦ **Code Folding Indicators** — denotes sections of code that can be folded or expanded

  - ◦ **Green check mark** — denotes statements, branches (true), or expressions in a particular line that have been covered.

  - ◦ **Red X with no subscripts** — denotes that multiple kinds of coverage on the line are not covered.

  - ◦ **Red X with subscripts** — denotes a statement, branch (false or true), condition or expression was not covered.

  - ◦ **Green E with no subscripts** — denotes a line of code to which active coverage exclusions have been applied. Every item on line is excluded; none are hit.

  - ◦ **Green E with subscripts** — denotes a line of codes with various degrees of exclusion.

Usage Notes

Disabling Automatic Opening of Source Files

By default, the Source window opens when the simulator hits a breakpoint, encounters a call to $finish(), or you are single stepping through your code. In each case, the simulator stops, the Source window opens and displays the last line of code that was executed. You can disable automatic opening by changing the preference variable settings:

- Breakpoints — Set the PrefSource(OpenOnBreak) variable to 0.

- $finish() call — Set the PrefSource(OpenOnFinish) variable to 0.

- Single Stepping — Set the PrefSource(OpenOnStep) variable to 0.

Displaying Multiple Source Files

By default each file you open or create is marked by a window tab, as shown in the graphic below.

Figure 129. Displaying Multiple Source Files
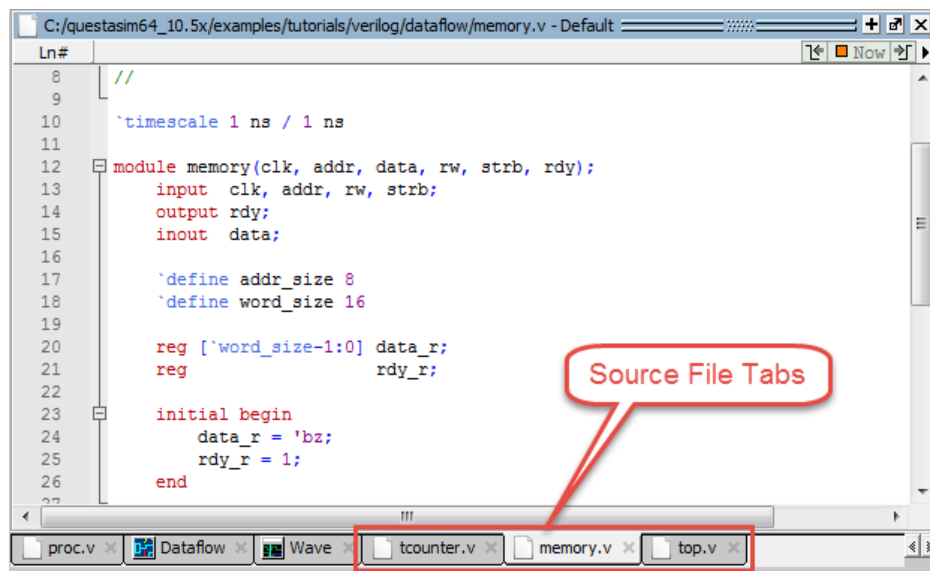
Using the Source Window

Use the Source window to view and analyze your data.

Dragging and Dropping Objects into the Wave and List Windows

Questa SIM allows you to drag and drop objects from the Source window to the Wave and List windows.

Procedure

1. In the Source window, double-click an object to highlight it.

2. Drag the object to the Wave or List window.

3. Optionally, to place a group of objects into the Wave and List windows, drag and drop any section of highlighted code.

4. When an object is dragged and dropped into the Wave window, the add wave command will be reflected in the Transcript window.

Adding Objects to Other Windows

You can add Source window objects to other dynamically linked windows with the right-click context menu.

Procedure

1. In the Source window, double-click an object to highlight it.

2. Right-click anywhere in the Source window to open the context menu.

3. Select **Add** to display the window options.

Figure 130. Source Window Add Options
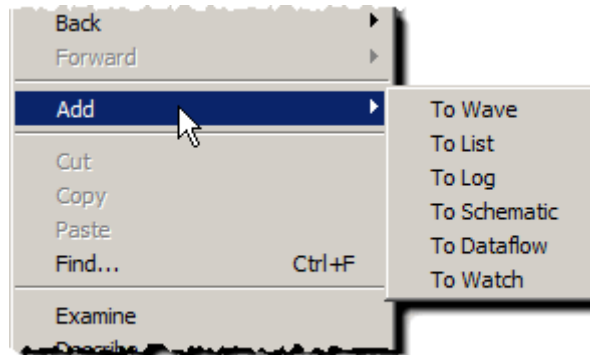
4. Select a window to add the selected object.

5. Optionally, to place a group of Source window objects into another window, highlight a section of code then right click to access the **Source > Add** menu.

Setting your Context by Navigating Source Files

When debugging your design from within the GUI, you can change your context while analyzing your source files.

Figure 131 shows the pop-up menu the tool displays after you select then right-click an instance name in a source file.

Figure 131. Setting Context from Source Files
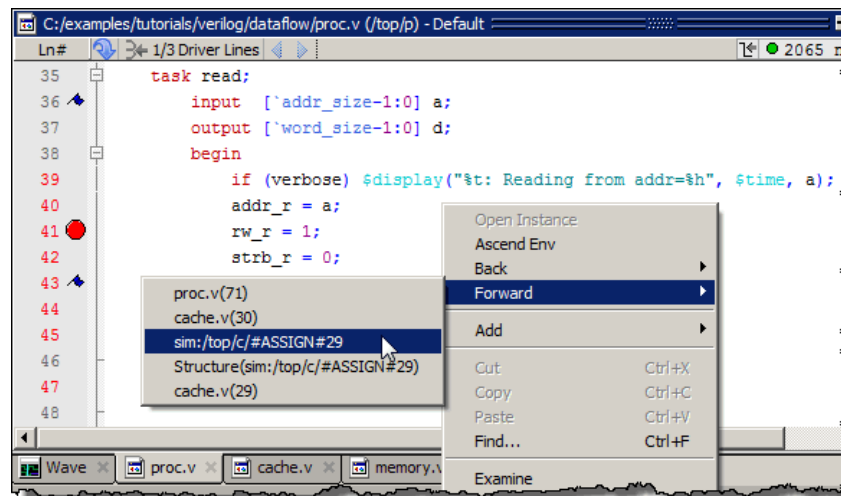
The title bar of the Source window displays your current context, parenthetically, after the file name and location. This changes as you alter your context, either through the pop-up menu or by changing your selection in the Structure window.

This functionality allows you to easily navigate your design for debugging purposes by remembering where you have been, similar to the functionality in most web browsers. The navigation options in the pop-up menu function as follows:

- **Open Instance** — changes your context to the instance you have selected within the source file. This is not available if you have not placed your cursor in, or highlighted the name of, an instance within your source file.

  If any ambiguities exists, most likely due to generate statements, this option opens a dialog box allowing you to choose from all available instances.

- **Ascend Env** — changes your context to the file and line number in the parent region where the current region is instantiated. This is not available if you are at the top-level of your design.

- **Forward/Back** — allows you to change to previously selected contexts. This is not available if you have not changed your context.

The Open Instance option is essentially executing an environment command to change your context, therefore any time you use this command manually at the command prompt, that information is also saved for use with the Forward/Back options.

Highlighted Text in a Source Window

The Source window can display text that is highlighted as a result of various conditions or operations, such as the following:

- Double-clicking an error message in the transcript shown during compilation

- Using **Evemt Traceback > Show Driver**

- Coverage-related operations

In these cases, the relevant text in the source code is shown with a persistent highlighting. To remove this highlighted display, choose **More > Clear Highlights** from the right-click popup menu of the Source window. If the Source window is docked, you can also perform this action by selecting **Source > More > Clear Highlights** from the Main menu. If the window is undocked, select **Edit > Advanced > Clear Highlights**.

> **Note:**
> Clear Highlights does not affect text that you have selected with the mouse cursor.

Example

To produce a compile error that displays highlighted text in the Source window, do the following:

1. Choose **Compile > Compile Options**

2. In the Compiler Options dialog box, click either the VHDL tab or the Verilog & SystemVerilog tab.

3. Enable Show source lines with errors and click OK.

4. Open a design file and create a known compile error (such as changing the word "entity" to "entry" or "module" to "nodule").

5. Choose **Compile > Compile** and then complete the Compile Source Files dialog box to finish compiling the file.

6. When the compile error appears in the Transcript window, double-click it.

7. The source window is opened (if needed), and the text containing the error is highlighted.

8. To remove the highlighting, choose **Source > More > Clear Highlights**.

Hyperlinked Text in a Source Window

The Source window supports hyperlinked navigation. To turn hyperlinked text on or off in the Source window, do the following:

1. Click anywhere in the Source window to make it the active window.

2. Select Source > Show Hyperlinks.

When you double-click hyperlinked text, the selection jumps from the usage of an object to its declaration. This provides the following operations:

- Jump from the usage of a signal, parameter, macro, or a variable to its declaration.

- Jump from a module declaration to its instantiation, and vice versa.

- Navigate back and forth between visited source files.

Coverage Data in the Source Window

The Source Window includes two columns for code coverage statistics when in Coverage View mode (vsim -coverage): Hits and BC. The coverage data presented in the Source window is calculated either "by file" or "by instance", as indicated just after the source file name. The value of these columns depends on the calculation method/mode used.

The meaning of the values in each of the columns are detailed in Table 109.

Table 109. Coverage Data Columns

Column	Value represents
Hits	In "byinstance" mode: how many times that line of code was executed
	In "byfile" mode: the sum of hits of that line from all the instances of the corresponding DU.
BC (branch coverage)	the branch executed
	Only "true" counts are displayed, with the exception of the AllFalse branch (if any). The AllFalse count is listed next to the first "if" condition in an if-else tree that does not contain a terminating catch-all "else" branch. You can determine the branch false count by subtracting counts in the BC column from the Hits column.

These columns provide an immediate visual indication about how your source code is executing. The code coverage indicator icons include check marks, 'X's and 'E's. A description of each code coverage indicator icon is provided in Table 110.

Figure 132. Coverage in Source Window
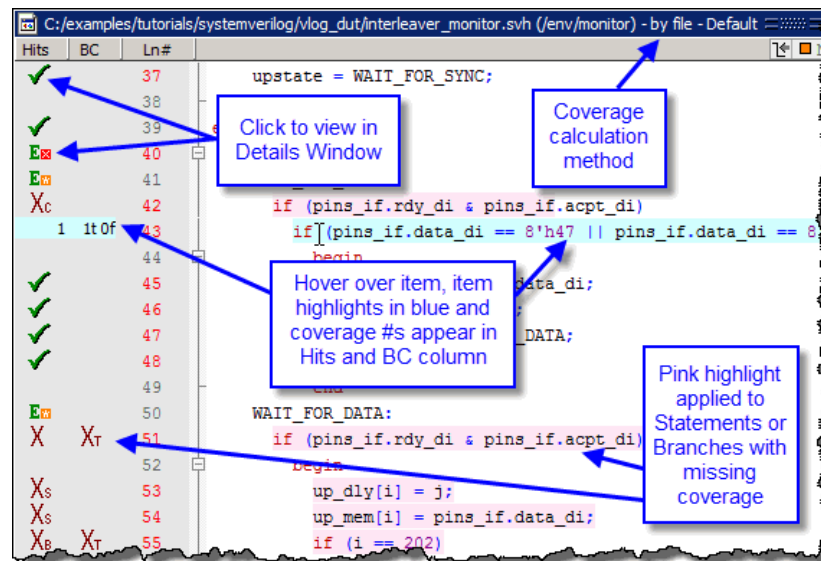
Detailed Coverage Information

To see more information about any coverage item, click the indicator icon, or in the Hits or BC column for the line of interest: In the case of a multiple-line item, this would be last line of the item. If the Coverage Details window is open, this action brings up detailed coverage information for that line. If the window is not open, a right click menu option is available to open it.

For example, when you select an expression in the Code Coverage Analysis' Expression Analysis window, and you click in the column of a line containing an expression, the associated truth tables appear in the Coverage Details window. Each line in the truth table is one of the possible combinations for the expression. The expression is considered to be covered (gets a green check mark) if the entire truth table is covered.

Coverage Numbers Mismatch Source

If coverage numbers are mismatched between Code Coverage Analysis windows and the Source window, check to make sure that both are being calculated the same — either "by file" or "by instance".

Hover Behavior

When you hover over statements, conditions or branches in the Source window, the Hits and BC columns display the coverage numbers for that line of code. For example, in Figure 132, the blue highlighted line shows that the expression (b=b'b1) was hit 1 time. The value in the Hits column shows the total coverage for all items in the truth table (as shown in the Coverage Details window when you click the specific line in the hits column).

Source Window Code Coverage Indicator Icons

Table 110. Source Window Code Coverage Indicators

Icon	Description/Indication
✔	All statements, branches (true), conditions, or expressions on a particular line have been executed

Table 110. Source Window Code Coverage Indicators (continued)

Icon	Description/Indication
X	Multiple kinds of coverage on the line were not executed
X$_T$	True branch not executed (BC column)
X$_F$	False branch not executed (BC column)
X$_C$	Condition not executed (Hits column)
X$_E$	Expression not executed (Hits column)
X$_B$	Branch not executed (Hits column)
X$_S$	Statement not executed (Hits column)
E	Indicates a line of code to which active coverage exclusions have been applied. Every item on the line is excluded; none are hit
E	Some excluded items are hit
E	Some items are excluded, and all items not excluded are hit
E	Some items are excluded, and some items not excluded have missing coverage
E	Auto exclusions have been applied to this line. Hover the cursor over the EA and a tool tip balloon appears with the reason for exclusion

To display only numbers in Hits and BC columns, choose **Tools > Code Coverage > Show Coverage Numbers**.

When the source window is active, you can skip to "missed lines" three ways:

- Select **Edit > Previous Coverage Miss** and **Edit > Next Coverage Miss**

- Click the **Previous zero hits** and **Next zero hits icons** on the toolbar.

- press Shift-Tab (previous miss) or Tab (next miss)

Controlling Data Displayed in a Source Window

The **Tools > Code Coverage** menu contains several commands for controlling coverage data display in a Source window.

Hide/Show coverage data — toggles the Hits column off and on.

Hide/Show branch coverage — toggles the BC column off and on.

Hide/Show coverage numbers — displays the number of executions in the Hits and BC columns rather than check marks and Xs. When multiple statements occur on a single line an ellipsis ("...") replaces the Hits number. In such cases, hover the cursor over each statement to highlight it and display the number of executions for that statement.

Show coverage By Instance — displays only the number of executions for the currently selected instance in the Main window workspace.

Debugging with Source Annotation

With source annotation you can interactively debug your design by analyzing your source files in addition to using the Wave and Signal windows. Source annotation displays simulation values, including transitions, for each signal in your source file.

Figure 133 shows an example of source annotation, where the red values are added below the signals.

Figure 133. Source Annotation Example
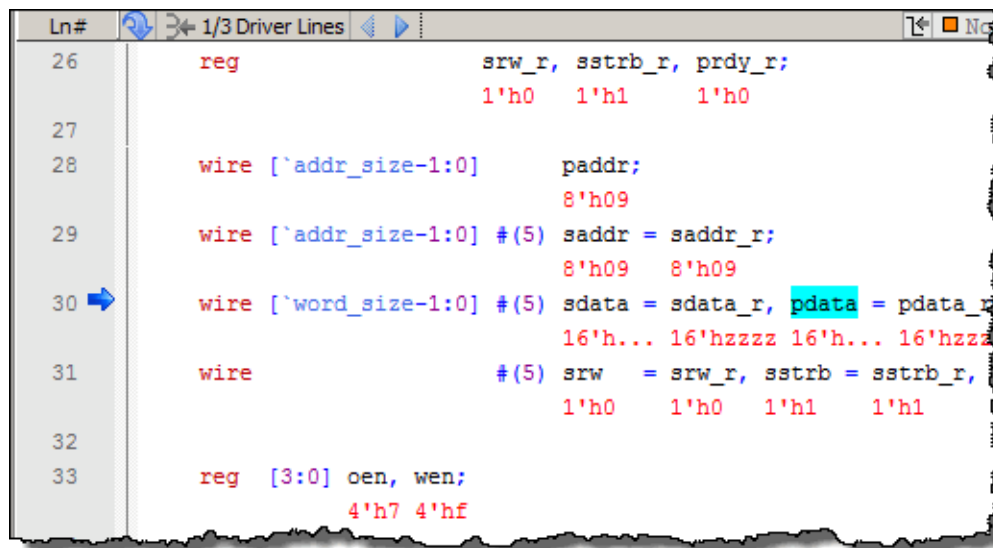
Turn on source annotation by selecting **Source > Show Source Annotation** or by right-clicking a source file and selecting **Show Source Annotation**. Note that transitions are displayed only for those signals that you have logged.

To analyze the values at a given time of the simulation you can either:

- Show the signal values at the current simulation time. This is the default behavior. The window automatically updates the values as you perform a run or a single-step action.

- Show the signal values at current cursor position in the Wave window.

You can switch between these two settings by performing the following actions:

- When Docked:

  ◦ **Source > Examine Now**

  ◦ **Source > Examine Current Cursor**

- When Undocked:

   ◦ **Tools > Options > Examine Now**

   ◦ **Tools > Options > Examine Current Cursor**

You can highlight a specific signal in the Wave window by double-clicking on an annotation value in the source file.

Accessing Textual Connectivity Information

The Source window contains textual connectivity information that allows you to explore the connectivity of your design though the source code. This feature is especially useful when used with source annotation turned on.

When you double-click an instance name in the Structure (sim) window, a Source window will open at the appropriate instance. You can then access textual connectivity information in the Source window by right-clicking any signal. This opens a popup menu that gives you the choices shown in Figure Figure 134.

Figure 134. Popup Menu Choices for Textual Dataflow Information
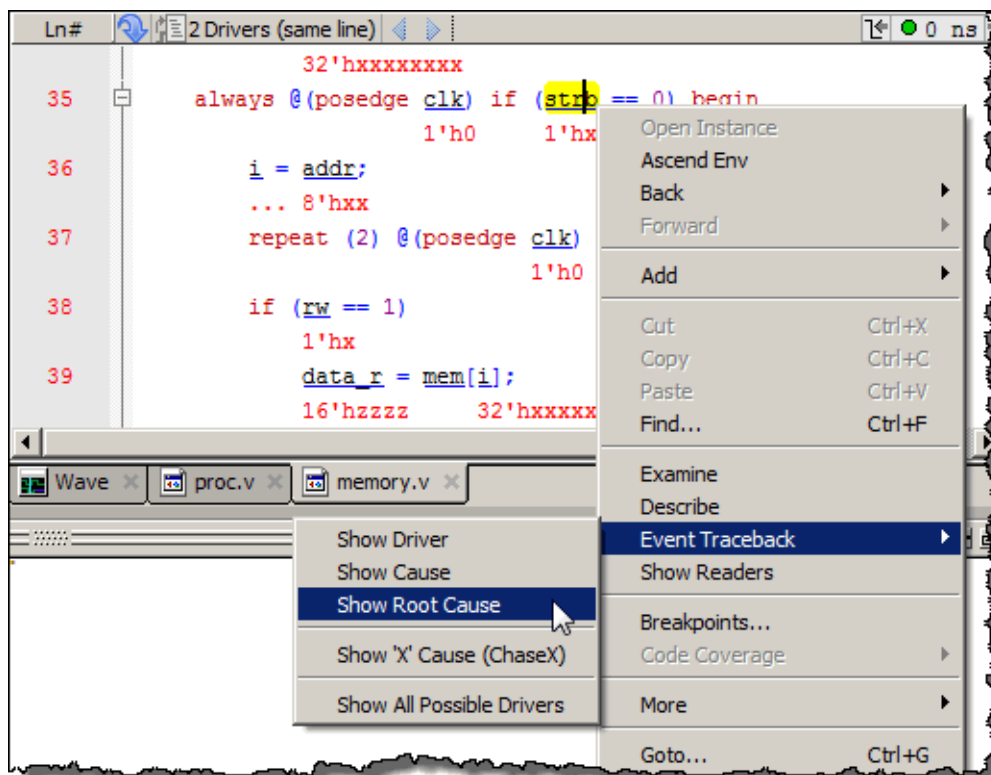
- The **Event Traceback > Show Driver** selection causes the Source window to jump to the source code defining the driver of the selected signal. If the Driver is in a different Source file, that file will open in a new Source window tab and the driver code will be highlighted. You can also jump to the driver of a signal by simply double-clicking the signal.

  If there is more than one driver for the signal, a Multiple Drivers dialog will open showing all driving processes (Figure 135).

Figure 135. Window Shows all Driving Processes
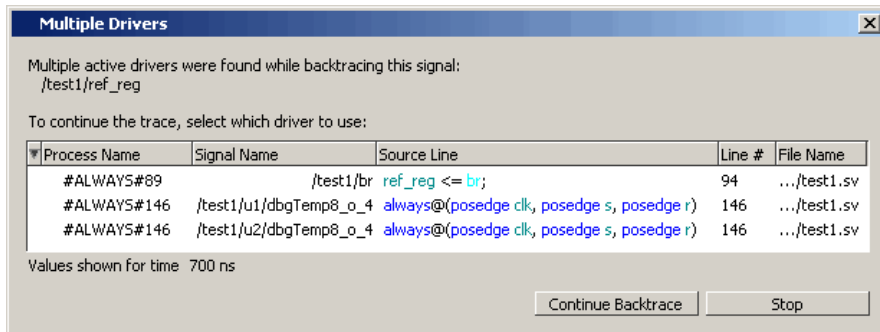
Select any driver to open the code for that driver.

- The **Show Readers** selection opens the Source Readers window. If there is more than one reader for the signal, all will be displayed (Figure 136).

Figure 136. Source Readers Dialog Displays All Signal Readers
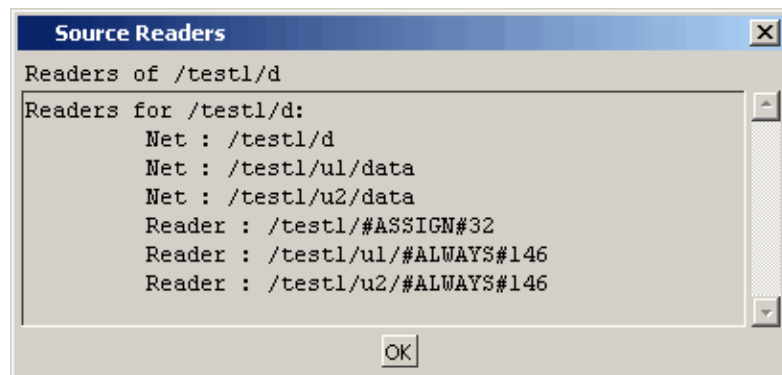
Limitations

The Source window's textual dataflow functions only work for pure HDL. It will not work for SystemC or for complex data types like SystemVerilog classes.

Setting File-Line Breakpoints with the GUI

You can easily set file-line breakpoints in your source code.

Procedure

1. In a Source window click in the line number column next to a red line number. A red ball denoting a breakpoint will appear (Figure 137).

Figure 137. Breakpoint in the Source Window
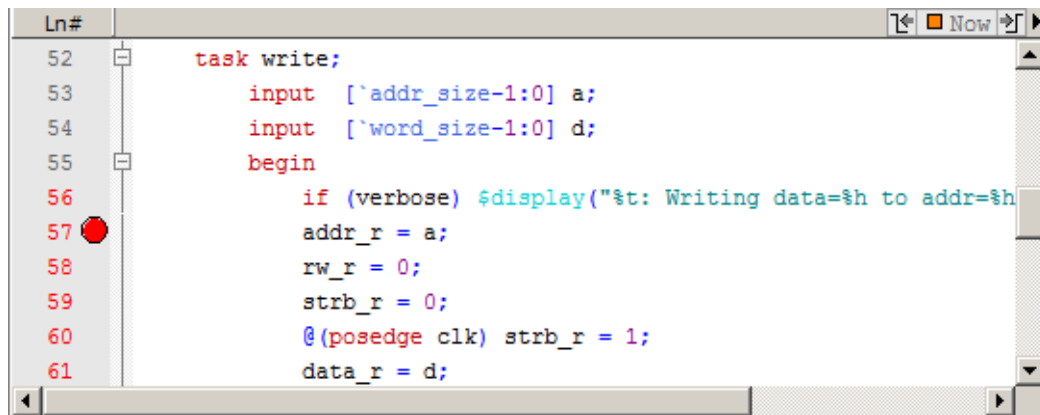
The breakpoint markers are toggles.

2. Click once to create the breakpoint; click again to disable or enable the breakpoint.

Note:
When running in full optimization mode, breakpoints may not be set. Run the design in non-optimized mode (or set +acc arguments) to enable you to set breakpoints in the design. Refer to Preservation of Object Visibility for Debugging and Design Object Visibility for Designs with PLI in the User's Manual.

To delete the breakpoint completely, right click the red breakpoint marker, and select **Remove Breakpoint**.

Other options on the context menu include:

- **Disable Breakpoint** — Deactivate the selected breakpoint.

- **Edit Breakpoint** — Open the File Breakpoint dialog to change breakpoint arguments.

- **Edit All Breakpoints** — Open the Modify Breakpoints dialog.

- **Run Until Here** — Run the simulation from the current simulation time up to the specified line of code. Refer to Run Until Here in the User's Manual for more information.

- **Add/Remove Bookmark** — Add or remove a file-line bookmark.

Adding File-Line Breakpoints with the bp Command

Use the bp command to add a file-line breakpoint from the VSIM> prompt.

For example:

bp top.vhd 147

sets a breakpoint in the source file *top.vhd* at line 147.

Editing File-Line Breakpoints

You can modify (or add) a breakpoint according to the line number in a source file.

Procedure

1. Perform one of the following actions:

   • Select **Tools > Breakpoints** from the Main menu.

   • Right-click a breakpoint and select **Edit All Breakpoints** from the popup menu.

   • Click the **Edit Breakpoints** toolbar button. See Simulate Toolbar.

   This displays the Modify Breakpoints dialog box shown in Figure 138.

Figure 138. Modifying Existing Breakpoints
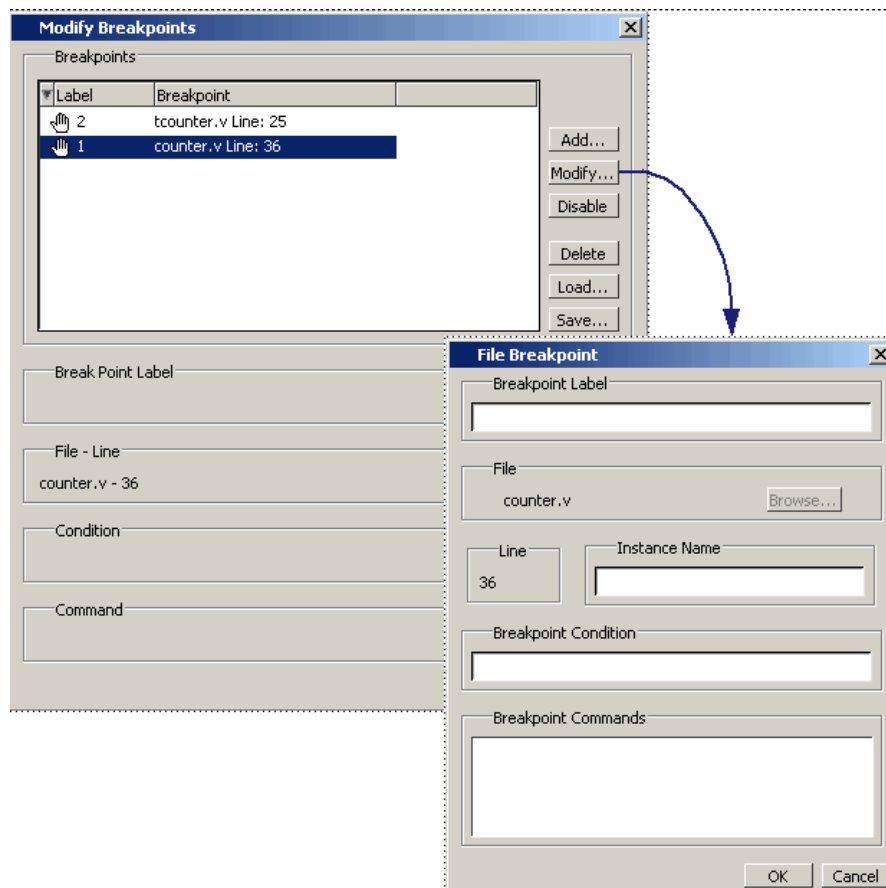
The Modify Breakpoints dialog box provides a list of all breakpoints in the design. Select a file-line breakpoint from the list.

2. Click **Modify**, which opens the File Breakpoint dialog box shown in Figure 138.

3. Fill out any of the following fields to modify the selected breakpoint:

Breakpoint Label — Designates a label for the breakpoint.

Instance Name — The full pathname to an instance that sets a breakpoint so it applies only to that specified instance.

Breakpoint Condition — One or more conditions that determine whether the breakpoint is observed. If the condition is true, the simulation stops at the breakpoint. If false, the simulation bypasses the breakpoint. A condition cannot refer to a VHDL variable (only a signal). Refer to the tip below for more information on proper syntax for breakpoints entered in the GUI.

Breakpoint Command — A string, enclosed in braces ({}) that specifies one or more commands to be executed at the breakpoint. Use a semicolon (;) to separate multiple commands.

Tip
: All fields in the File Breakpoint dialog box, except the Breakpoint Condition field, use the same syntax and format as the -inst switch and the command string of the bp command. Do not enclose the expression entered in the Breakpoint Condition field in quotation marks ( ). For more information on these command options, refer to the bp command in the QuestaSIM Reference Manual.

---

4. Click **OK** to close the File Breakpoints dialog box.

5. Click **OK** to close the Modify Breakpoints dialog box.

The Modify Breakpoints dialog box (Figure 138) includes Load and Save buttons that allow you to load or save breakpoints.

Setting Conditional Breakpoints

In dynamic class-based code, an expression can be executed by more than one object or class instance during the simulation of a design. You set a conditional breakpoint on the line in the source file that defines the expression and specifies a condition of the expression or instance you want to examine. You can write conditional breakpoints to evaluate an absolute expression or a relative expression.

Note:
Be sure to first compile and load your simulation before setting a conditional breakpoint.

You can use the SystemVerilog keyword *this* when writing conditional breakpoints to refer to properties, parameters or methods of an instance. The value of *this* changes every time the expression is evaluated based on the properties of the current instance. Your context must be within a local method of the same class when specifying the keyword *this* in the condition for a breakpoint. Strings are not allowed.

The conditional breakpoint examples below refer to the following SystemVerilog source code file *source.sv*:

Figure 139. Source Code for *source.sv*

```
1   class Simple;
2  integer cnt;
3  integer id;
4  Simple next;
5
6  function new(int x);
7   id=x;
8   cnt=0
9   next=null
10  endfunction
11
12  task up;
13   cnt=cnt+1;
14   if (next) begin
15    next.up;
16   end
17  endtask
18  endclass
19
20  module test;
21  reg clk;
22  Simple a;
23  Simple b;
24
25  initial
26  begin
27   a = new(7);
28   b = new(5);
29  end
30
31  always @(posedge clk)
32  begin
33   a.up;
34   b.up;
35   a.up
36  end;
37  endmodule
```

> **Note:**
> You must use the +acc switch when optimizing with vopt to preserve visibility of SystemVerilog class objects.

Setting a Breakpoint For a Specific Instance

Enter the following on the command line:

bp simple.sv 13 -cond {this.id==7}

Results

The simulation breaks at line 13 of the *simple.sv* source file (Figure 139) the first time module a hits the expression because the breakpoint is evaluating for an id of 7 (refer to line 27).

Setting a Breakpoint For a Specified Value of Any Instance

Enter the following on the command line:

> **bp simple.sv 13 -cond {this.cnt==8}**

Results

The simulation evaluates the expression at line 13 in the *simple.sv* source file (Figure 139), continuing the simulation run if the breakpoint evaluates to false. When an instance evaluates to true the simulation stops, the source is opened and highlights line 13 with a blue arrow. The first time cnt=8 evaluates to true, the simulation breaks for an instance of module Simple b. When you resume the simulation, the expression evaluates to cnt=8 again, but this time for an instance of module Simple a.

You can also set this breakpoint with the GUI:

1. Right-click line 13 of the simple.sv source file.

2. Select Edit Breakpoint 13 from the drop menu.

3. Enter

```
this.cnt==8
```

in the **Breakpoint Condition** field of the **Modify Breakpoint** dialog box. (Refer to Figure 138) Note that the file name and line number are automatically entered.

Checking Object Values and Descriptions

You can check the value or description of signals, indexes, macros, and other objects in the Source window.

There are two quick methods to determine the value and description of an object:

- Select an object, then right-click and choose **Examine** or **Describe** from the context menu.

- Pause the cursor over an object to see an examine pop-up

Figure 140. Source Window Description
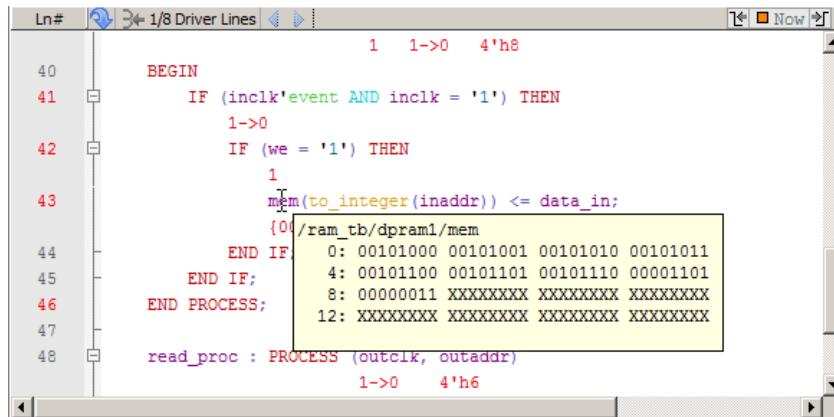
You can choose **Source > Examine Now** or **Source > Examine Current Cursor** to choose at what simulation time the object is examined or described.

You can also invoke the examine and/or describe commands on the command line or in a DO file script.

Marking Lines with Bookmarks

Source window bookmarks are blue flags that mark lines in a source file. These graphical icons may ease navigation through a large source file by highlighting certain lines.

As noted above in the discussion about finding text in the Source window, you can insert bookmarks on any line containing the text for which you are searching. The other method for inserting bookmarks is to right-click a line number and select **Add/Remove Bookmark**. To remove a bookmark, right-click the line number and select Add/Remove Bookmark again.

To remove all bookmarks from the Source window, select **Source > Clear Bookmarks** from the menu bar when the Source window is active.

Performing Incremental Search for Specific Code

The Source window includes a Find function that allows you to do an incremental search for specific code.

To activate the Find bar (Figure 141) in the Source window choose **Edit > Find** from the Main menus or

click the **Find** icon in the Main toolbar. For more information see Find and Filter Functions.

Figure 141. Source Window with Find Toolbar
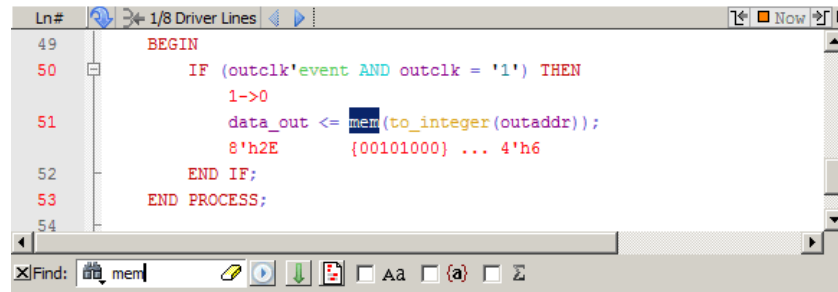
Customizing the Source Window

You can customize the appearance and behavior of the Source window in several ways.

- Changing a *modelsim.ini* variable: for example, character encoding of files is controlled by the variable.

- Changing a preference variable: for example,

  ◦ tab spacing: change the PrefSource(tabs) preference variable. Refer to Setting GUI Preferences for details on setting preference variables.

  ◦ Syntax highlighting: change the PrefSource(highlightExecutableLines) preference variable.

  ◦ Underlining of hyperlinked code: change the prefMain(HyperLinkingUnderline) preference variable.

- General Source window fonts and appearance: select **Souce > Tools > Edit Preferences**and make changes to the settings on the **By Window** tab.

Refer to the GUI Preferences appendix for more information.

Structure Window

To access:

- **View > Structure**

- view structure command

Use this window to view the hierarchical structure of the active simulation.

Description

The name of the structure window, as shown in the title bar or in the tab if grouped with other windows, can vary:

sim — This is the name shown for the Structure window for the active simulation.

dataset_name — The Structure window takes the name of any dataset you load through the **File > Datasets** menu item or the dataset open command.

By default, the Structure window opens in a tab group with the Library windows after starting a simulation. You can also open the Structure window with the View Objects Window Button.

The hierarchical view includes an entry for each object within the design. When you select an object in a Structure window, it becomes the current region.

By default, the coverage statistics displayed in the columns within the Structure window are recursive. You can select to view coverage statistics for local instances by deselecting **Code Coverage > Enable Recursive Sums**. Refer to Coverage Aggregation in the Structure Window in the User's Manual for details on coverage numbers.

The contents of several windows automatically update based on which object you select, including the Source window, Objects window, Processes window, and Locals window. All mouse button operations clear the current selection and select the item under the cursor.

Figure 142. Structure Window
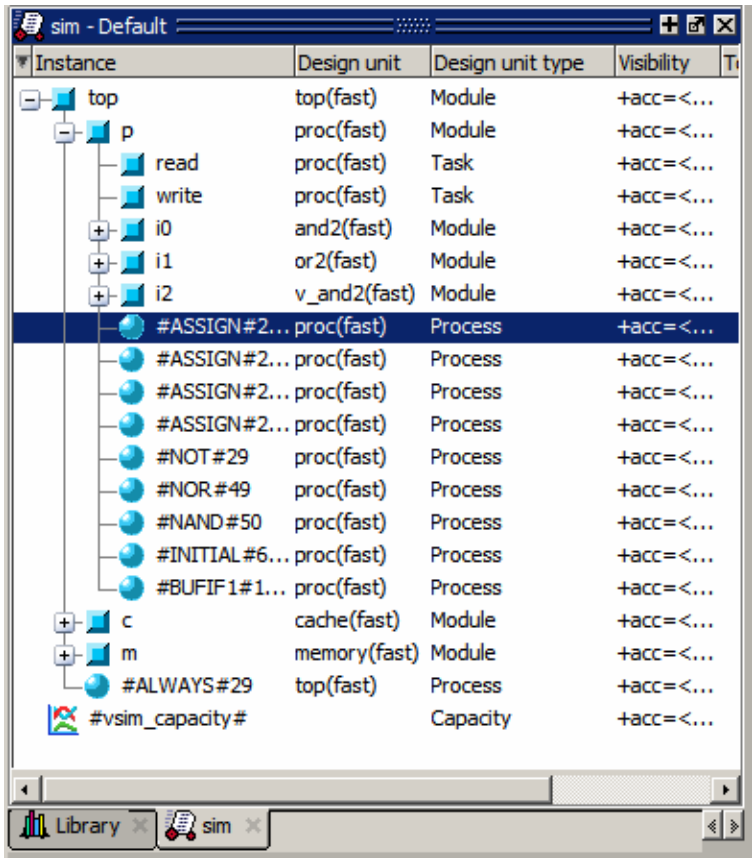
Right-click an object in the Structure window to display the popup menu and select an option in Table 4-57.

Top Category Column

The Top Category column allows you to:

- Determine which top level categories are expanded with the initial invocation of the Structure window (sim tab).

- Determine the default ordering of the top level categories in the Structure window.

To determine which category is expanded upon initial invocation, do the following:

1. Choose **Tools > Edit Preferences** from the menus. This opens the Preferences window.

2. Click the **By Name** tab in the Preferences window.

3. Click the '+' sign next to "Structure" to view a list of the preference variables for the Structure window.

4. Click the initExpandOff variable to select it and then click the **Change Value** button. This will open the Change Structure Preference Value dialog box.

Figure 143. Change Top Level Category Expansion
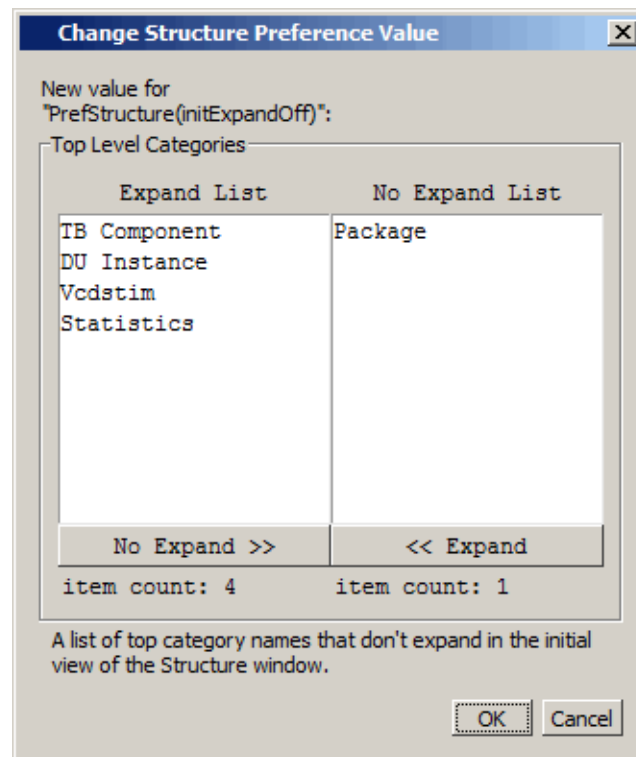
The Change Structure Preference Value dialog allows you to put a top level category in the Expand List or the No Expand List. Simply click to select a category, then click "No Expand" or "Expand" to move the selection into the list you want. By default, on the Package category is in the No Expand List.

To determine the default ordering of the top level categories in the Structure window simply click the "Top Category" heading of the column to open a new Change Structure Preference Value dialog.

Figure 144. Change Default Ordering of Structure Window
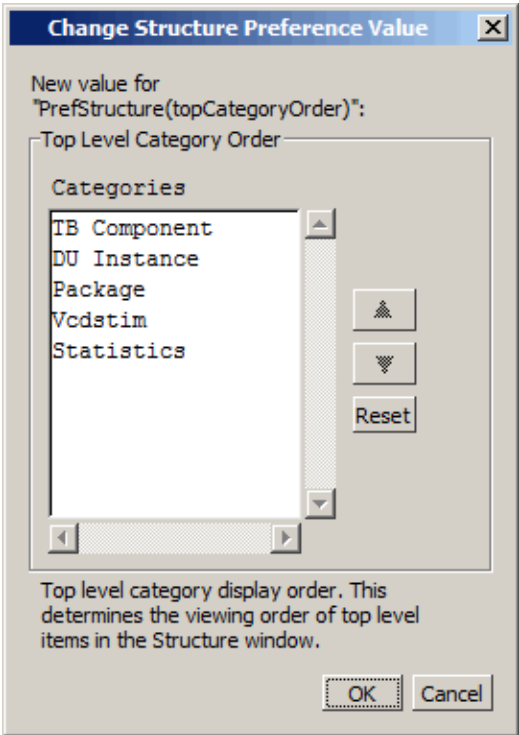
To change the default ordering of the top level categories simply select a category then use the up or down arrowheads to move the selection.

Objects

Table 111. Structure Window Popup Menu

Popup Menu Item	Description
View Declaration	Opens the source file and bookmarks the object.
View Instantiation	Opens the source file and bookmarks the object.
UVM	View UVM Details — Opens the UVM Details window.
	Display Objections — Prints the current UVM objections associated with the selected UVM component and its' children to the transcript window
UPF	Allows you to view additional information about UPF objects:
	• View Power Domain — Opens the source file for viewing UPF information regarding the power domain of the object.
	• Add Power Domain to wave — Adds UPF-based information related to the power domain of the object to the Wave window.
Add Wave	Adds the selected object(s) to the Wave window.

Table 111. Structure Window Popup Menu (continued)

Popup Menu Item	Description
Add Wave New	Creates a new instance of the Wave window and adds the selected object(s) to that window.
Add Wave To	Opens a drop down list of Wave windows when multiple windows exist. Adds the selected object(s) to the selected Wave window.
Add Dataflow	Adds the selected object(s) to a Dataflow window.
Add to	Add the selected object(s) to any one of the following: Wave window, List window, Log file, Schematic window, Dataflow window. You may choose to add only the Selected Signals, all Signals in Region, all Signals in Design.
Copy	Copies the object instance path to the clipboard
Find	Opens the Search Bar (at bottom of window) in the Find mode to make searching for objects easier, especially with large designs.
Save Selected	Saves all hierarchy under the selected instance.
Expand Selected	Displays the hierarchy of the object recursively.
Collapse Selected	Closes the hierarchy of the object.
Collapse All	Collapses the hierarchy to the top instance.
Code Coverage >	These menu items are only available when you run the simulation with the -coverage switch. • Code Coverage Reports — Opens the Coverage Text Report dialog • Clear Code Coverage Data — Clears all code coverage information collected during simulation • Enable Recursive Coverage Sums — Displays coverage data and graphs for each design object or file, recursively: enabled by default
Test Analysis	When a UCDB file is selected, allows you to do the following: • Find Tests with Least Coverage • Most Coverage • Zero Coverage • Non-Zero Coverage • Rank Most Effective Tests... • Summary...
XML Import Hint	Displays the XML Import Hint dialog box with information about the Link Type and Name

Table 111. Structure Window Popup Menu (continued)

Popup Menu Item	Description
Show	Lists the design unit types that are currently displayed. • Processes • Functions • Packages • Tasks • Statement • VPackages • VITypedef • SVClass • Class Instances • Capacity • Change Filter

Table 112. Columns in the Structure Window

Column name	Description
Assertion hits	Assertion hits shows different counts based on whether the -assertcounts or -assertdebug argument is used with the vsim command: • with -assertcounts or -assertdebug: number of assertions whose pass count is greater than 0 and whose fail count is equal to 0 • without -assertcounts or -assertdebug: number of assertions whose pass count is equal to 1 and whose fail count is equal to 0
Assertion misses	the number of assertions whose fail count is greater than 0 or whose fail and pass counts both are equal to 0 (vacuous assertions)
Assertion %	the number of hits from the total number of assertions, as a percentage
Assertion graph	a bar chart displaying the Assertion %; if the percentage is below 90%, the bar is red; 90% or more, the bar is green; you can change this threshold percentage by editing the PrefCoverage(cutoff) preference variable
Branch count	Files window — the number of executable branches in each file Structure window — the number of executable branches in each level and all levels under that level
Branches hit	the number of executable branches that have been executed in the current simulation
Branches missed	the number of executable branches that were not executed in the current simulation
Branch %	the current ratio of Branch hits to Branch count

Table 112. Columns in the Structure Window (continued)

Column name	Description
Branch graph	a bar chart displaying the Branch %; if the percentage is below 90%, the bar is red; 90% or more, the bar is green; you can change this threshold percentage by editing the PrefCoverage(cutoff) preference variable
Condition rows	Files window — the number of conditions in each file Structure window — the number of conditions in each level and all levels under that level
Conditions hit	Files window — the number of times the conditions in a file have been executed Structure window — the number of times the conditions in a level, and all levels under that level, have been executed
Conditions missed	Files window — the number of conditions in a file that were not executed Structure window — the number of conditions in a level, and all levels under that level, that were not executed
Condition %	the current ratio of Condition hits to Condition rows
Condition graph	a bar chart displaying the Condition %; if the percentage is below 90%, the bar is red; 90% or more, the bar is green; you can change this threshold percentage by editing the PrefCoverage(cutoff) preference variable
Covergroup %	the number of hits from the total number of covergroups, as a percentage
Cover hits	the number of cover directives whose count values are greater than or equal to the at_least value.
Cover misses	the number of cover directives whose count values are less than the at_least value
Cover %	the number of hits from the total number of cover directives, as a percentage
Cover graph	a bar chart displaying the Cover directive %; if the percentage is below 90%, the bar is red; 90% or more, the bar is green; you can change this threshold percentage by editing the PrefCoverage(cutoff) preference variable
Cover Options	The +cover settings used for compilation/simulation of that design unit
Design Unit	The name of the design unit
Design Unit Type	The type of design unit
Expression rows	the number of executable expressions in each level and all levels subsumed under that level
Expressions hit	the number of times expressions in a level, and each level under that level, have been executed
Expressions missed	the number of executable expressions in a level, and all levels under that level, that were not executed
Expression %	the current ratio of Expression hits to Expression rows

Table 112. Columns in the Structure Window (continued)

Column name	Description
Expression graph	a bar chart displaying the Expression %; if the percentage is below 90%, the bar is red; 90% or more, the bar is green; you can change this threshold percentage by editing the PrefCoverage(cutoff) preference variable
FEC Condition rows	Files window — the number of FEC conditions in each file Structure window — the number of conditions in each level and all levels under that level
FEC Conditions hit	Files window — the number of times the FEC conditions in a file have been executed Structure window — the number of times the conditions in a level, and all levels under that level, have been executed
FEC Conditions missed	Files window — the number of FEC conditions in a file that were not executed Structure window — the number of conditions in a level, and all levels under that level, that were not executed
FEC Condition %	the current ratio of FEC Condition hits to FEC Condition rows
FEC Condition graph	a bar chart displaying the FEC Condition %; if the percentage is below 90%, the bar is red; 90% or more, the bar is green; you can change this threshold percentage by editing the PrefCoverage(cutoff) preference variable
FEC Expression rows	Files window — the number of executable expressions in each file Structure window — the number of executable expressions in each level and all levels subsumed under that level
FEC Expressions hit	Files window — the number of times expressions in a file have been executed Structure window — the number of times expressions in a level, and each level under that level, have been executed
FEC Expressions missed	Files window — the number of executable expressions in a file that were not executed Structure window — the number of executable expressions in a level, and all levels under that level, that were not executed
FEC Expression %	the current ratio of FEC Expression hits to FEC Expression rows
FEC Expression graph	a bar chart displaying the FEC Expression %; if the percentage is below 90%, the bar is red; 90% or more, the bar is green; you can change this threshold percentage by editing the PrefCoverage(cutoff) preference variable
PD Name	Shows the name of any associated power domain.
States	Files window — the number of states encountered in each file Structure window — level
States hit	Files window — the number of times the states were hit Structure window — the number of times states in a level, and each level under that level, have been hit

Table 112. Columns in the Structure Window (continued)

Column name	Description
States missed	Files window — the number of states in a file that were not hit Structure window — the number of states in a level, and all levels under that level, that were not hit
State %	the current ratio of State hits to State rows
State graph	a bar chart displaying the State %; if the percentage is below 90%, the bar is red; 90% or more, the bar is green; you can change this threshold percentage by editing the PrefCoverage(cutoff) preference variable
Stmt count	the number of executable statements in each level and all levels under that level
Stmts hit	the number of executable statements that were executed in each level and all levels under that level
Stmts missed	the number of executable statements that were not executed in each level and all levels under that level
Stmt %	the current ratio of Stmt hits to Stmt count
Stmt graph	a bar chart displaying the Stmt %; if the percentage is below 90%, the bar is red; 90% or more, the bar is green; you can change this threshold percentage by editing the PrefCoverage(cutoff) preference variable
Toggle nodes	the number of points in each instance where the logic will transition from one state to another
Toggles hit	the number of nodes in each instance that have transitioned at least once
Toggles missed	the number of nodes in each instance that have not transitioned at least once
Toggle %	the current ratio of Toggle hits to Toggle nodes
Toggle graph	a bar chart displaying the Toggle %; if the percentage is below 90%, the bar is red; 90% or more, the bar is green; you can change this threshold percentage by editing the PrefCoverage(cutoff) preference variable
Top Category	displays the category that is expanded in the initial invocation of the Structure window; categories are DU Instance, TB component, Package, Vcdstim, and Statistics
Total Coverage	The weighted average of all the coverage types (functional coverage and code coverage) is recursive. Deselect **Code Coverage > Enable Recursive Coverage Sums** to view results for the local instance. See Calculation of Total Coverage in the User's Manual for coverage statistics details.
Transitions	Files window — the number of transitions encountered in each file Structure window — the number of states encountered in each level and all levels subsumed under that level
Transitions hit	Files window — the number of times the transitions were hit

Table 112. Columns in the Structure Window (continued)

Column name	Description
	Structure window — the number of times transitions in a level, and each level under that level, have been hit
Transitions missed	Files window — the number of transitions in a file that were not hit
	Structure window — the number of transitions in a level, and all levels under that level, that were not hit
Transition %	the current ratio of Transition hits to Transition rows
Transition graph	a bar chart displaying the State %; if the percentage is below 90%, the bar is red; 90% or more, the bar is green; you can change this threshold percentage by editing the PrefCoverage(cutoff) preference variable
Visibility	The +acc settings used for compilation/optimization of that design unit

Structure Window Tasks

This section describes tasks for using the Structure window.

Display Source Code of a Structure Window Object

You can highlight the line of code that declares a given object in several ways.

- **Double-click an object** — Opens the file in a new Source window, or activates the file if it is already open.

- **Single-click an object** — Highlights the code if the file is already showing in an active Source window.

Add Structure Window Objects to Other Windows

You can add objects from the Structure window to the Dataflow window, Schematic window, List window, Watch window or Wave window in multiple ways.

- Mouse — Drag and drop

- Menu Selection — **Add > To <window>**

- Toolbar — Add Selected to Window Button > **Add to window**

- Command — add list, add wave, or add dataflow

When you drag and drop objects from the Structure window to the Wave, Dataflow, or Schematic windows, the add wave, add dataflow, and add schematic commands will be reflected in the Transcript window.

Finding Items in the Structure Window

To find items in the Structure window, press Ctrl-F on your keyboard with the Structure window active. This opens the Find bar at the bottom of the window.

Refer to the Find and Filter Functions section for details. As you type in the Find field, a popup window opens to display a list of matches (Figure 145). With 'Search While Typing' enabled (the default) each keypress changes the pattern and restarts the search immediately.

Figure 145. Find Mode Popup Displays Matches
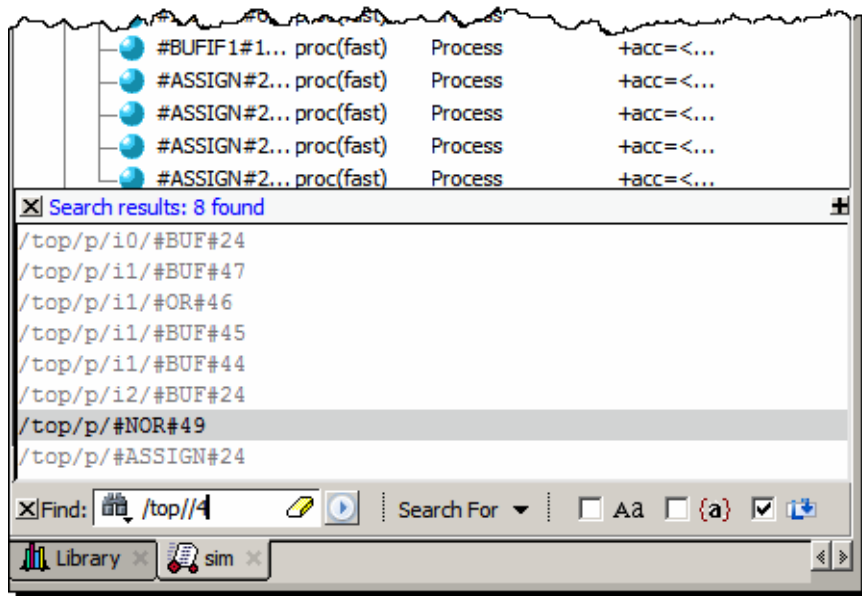
The Structure window Find bar supports hierarchical searching to limit the regions of a search. The forward slash (/) character is used to separate the search words. A double slash (//) is used to specify a recursive search from the double slash down the hierarchy. For example:

foo — search the entire design space for regions containing "foo" in the name.

foo?bar — search the entire design space for regions containing "foo" then any single alphanumeric character, followed by "bar"

foo*bar — search the design for a name containing "foo", a string of zero or more alphanumeric characters, followed by "bar. For example, to following names match a search for "foo*bar": "foobar", "foo_fred_bar", and "fooIsAbar". "fooISbad" does not match the search string.

/foo — search the top of the design hierarchy for regions containing "foo".

/foo/bar — search for regions containing "foo" at the top, and then regions containing "bar".

/foo//bar — search for regions containing "bar" recursively below all top level regions containing "foo".

To search for a name that contains the slash (/) character, escape the slash using a backslash (\). For example: \/bar.

When you double-click any item in the match list that item is highlighted in the Structure window and the popup is removed. The search can be canceled by clicking on the 'x' button in the popup window or by pressing the Esc key on your keyboard.

Filtering Structure Window Objects

You can control the types of information available in the Structure window through the **View > Filter** menu items.

Processes — Implicit wire processes

Functions — Verilog and VHDL Functions
Packages — VHDL Packages
Tasks — Verilog Tasks
Statement — Verilog Statements
VIPackage — Verilog Packages
VITypedef — Verilog Type Definitions
SVClass— SystemVerilog class instances
Cell Instances — Verilog cell instances or VHDL architecture instance.
Capacity — Memory capacity design unit
Assertion — VHDL and Verilog assertions
Subprogram — VHDL procedures and functions; Verilog functions and tasks

Code Coverage in the Structure Window

The Structure window displays code coverage information in the Structure (sim) window for any datasets being simulated. When coverage is invoked, several columns for displaying coverage data are added to these windows.

You can toggle columns on/off by right-clicking on a column name and selecting from the context menu that appears. Figure 146 shows a portion of the Structure window with code coverage data displayed.

Figure 146. Code Coverage Data in the Structure Window

You can sort code coverage information for any column by clicking the column heading. Clicking the column heading again will reverse the order.

Coverage information in the Structure window is dynamically linked to the Code Coverage Analysis windows. Click the left mouse button on any file in the Files window to display that file's un-executed statements, branches, conditions, expressions, and toggles in the Code Coverage Analysis windows. Lines from the selected file that are excluded from coverage statistics are also displayed in the Code Coverage Analysis windows.

For details on how the Total Coverage column statistics are calculated, refer to Calculation of Total Coverage in the User's Manual.

Transaction View Window

The Transaction View window displays information about a selected Questa Verification IP transaction instance.

It is only available for viewing transactions for Questa Verification IPs. For detailed information on this window, refer to Questa Verification IP Transaction Details in Transaction View Window in the User's Manual.

Transcript Window

To access: The Transcript window is always open in the Main window and cannot be closed.

The Transcript window maintains a running history of commands that are invoked and messages that occur as you work with Questa SIM. When a simulation is running, the Transcript displays a VSIM prompt, allowing you to enter command-line commands from within the graphic interface.

Description

You can scroll backward and forward through the current work history by using the vertical scrollbar. You can also use arrow keys to recall previous commands, or copy and paste using the mouse within the window (see Main and Source Window Mouse and Keyboard Shortcuts for details).

The **Transcrip**t tab contains the command line interface, identified by the Questa SIM prompt, and the simulation interface, identified by the VSIM prompt.

When undocked, the Transcript window allows access to the following toolbars:

- Standard Toolbar

- Help Toolbar

Objects

Table 113. Transcript Menu - Item Description

Menu Item	Description
Adjust Font Scaling	Displays the Adjust Scaling dialog box, which allows you to adjust how fonts appear for your display environment. Directions are available in the dialog box.
Transcript File	Allows you to change the default name used when saving the transcript file. The saved transcript file will contain all the text in the current transcript file.
Command History	Allows you to change the default name used when saving command history information. This file is saved at the same time as the transcript file.
Save File	Allows you to change the default name used when selecting **File > Save As**.
Saved Lines	Allows you to change how many lines of text are saved in the transcript window. Setting this value to zero (0) saves all lines.
Line Prefix	Allows you to change the character(s) that precedes the lines in the transcript.
Update Rate	Allows you to change the length of time (in ms) between transcript refreshes.
Questa SIM Prompt	Allows you to change the string used for the command line prompt.
VSIM Prompt	Allows you to change the string used for the simulation prompt.

Table 113. Transcript Menu - Item Description (continued)

Menu Item	Description
Paused Prompt	Allows you to change the string used for when the simulation is paused.

Transcript Window Tasks

You can perform multiple tasks in the Transcript window.

Saving the Transcript File

Colorizing the Transcript

Disabling Creation of the Transcript File

Performing an Incremental Search

Using Automatic Command Help

Using drivers and Readers Command Results

Saving the Transcript File

Variable settings determine the filename used for saving the transcript. If either PrefMain(file) in the *.modelsim* file or TranscriptFile in the *modelsim.ini* file is set, then the transcript output is logged to the specified file.

By default the TranscriptFile variable in *modelsim.ini* is set to *transcript*. If either variable is set, the transcript contents are always saved and no explicit saving is necessary.

If you would like to save an additional copy of the transcript with a different filename, click in the Transcript window and then choose **File > Save As**, or **File > Save**. The initial save must be made with the **Save As** selection, which stores the filename in the Tcl variable PrefMain(saveFile). Subsequent saves can be made with the **Save** selection. Since no automatic saves are performed for this file, it is written only when you invoke a Save command. The file is written to the specified directory and records the contents of the transcript at the time of the save.

Refer to Creating a Transcript File in the User's Manual for more information about creating, locating, and saving a transcript file.

Saving a Transcript File as a DO File

Perform the following steps to save a transcript file as a DO file.

1. Open a saved transcript file in a text editor.

2. Remove all commented lines leaving only the lines with commands.

3. Save the file as *<name>.do*.

Refer to the do command for information about executing a DO file.

Changing the Number of Lines Saved in the Transcript Window

By default, the Transcript window retains the last 5000 lines of output from the transcript. You can change this default by choosing **Transcript > Saved Lines**. Setting this variable to 0 instructs the tool to retain all lines of the transcript.

Colorizing the Transcript

By default, all Transcript window messages are printed in blue. You may colorized Transcript messages according to severity.

Procedure

1. Choose **Tools > Edit Preferences** from the Main window menus.

2. In the Preferences window click the **By Name** tab.

3. Expand the list of Preferences under "Main."

4. Select the colorizeTranscript preference and click the **Change Value** button.

5. Enter "1" in the Change Main Preference Value dialog and click **OK** (Figure 147).

Figure 147. Changing the colorizeTranscript Preference Value
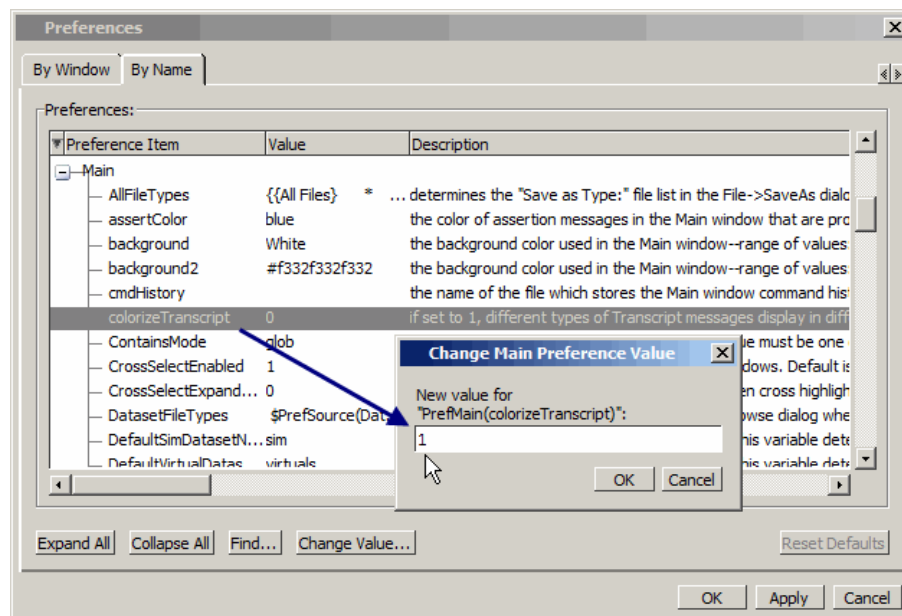
Disabling Creation of the Transcript File

You can disable the creation of the transcript file by using a Questa SIM command immediately after Questa SIM starts.

Procedure

Enter the following command at the prompt:

transcript file ""

Performing an Incremental Search

The **Transcript** tab includes an Find function that allows you to do an incremental search for specific text. To activate the Find bar choose **Edit > Find** from the menus or click the **Find** icon in the toolbar.

For more information see Find and Filter Functions.

Figure 148. Transcript Window with Find Toolbar
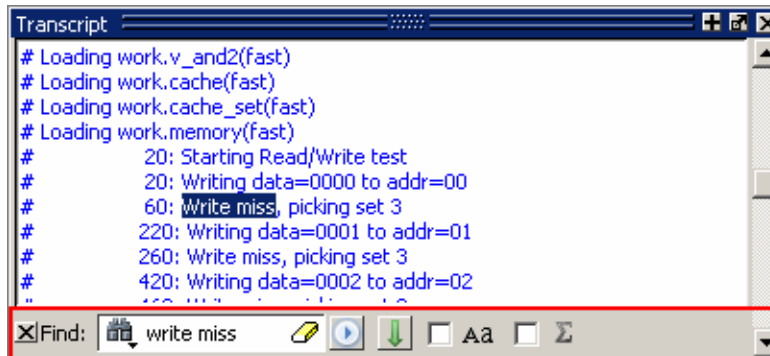
Using Automatic Command Help

When you start typing a command at the prompt, a dropdown box appears which lists the available commands matching what has been typed so far. You may use the Up and Down arrow keys or the mouse to select the desired command. When a unique command has been entered, the command usage is presented in the drop down box.

You can toggle this feature on and off by choosing **Help > Command Completion**.

Using drivers and Readers Command Results

The output from the drivers and readers commands, which is displayed in the Transcript window as hypertext links, allows you to right-click to open a drop-down menu and to quickly add signals to various windows. It also includes a "View Declaration" item to open the source definition of the signal.

Figure 149. drivers Command Results in Transcript
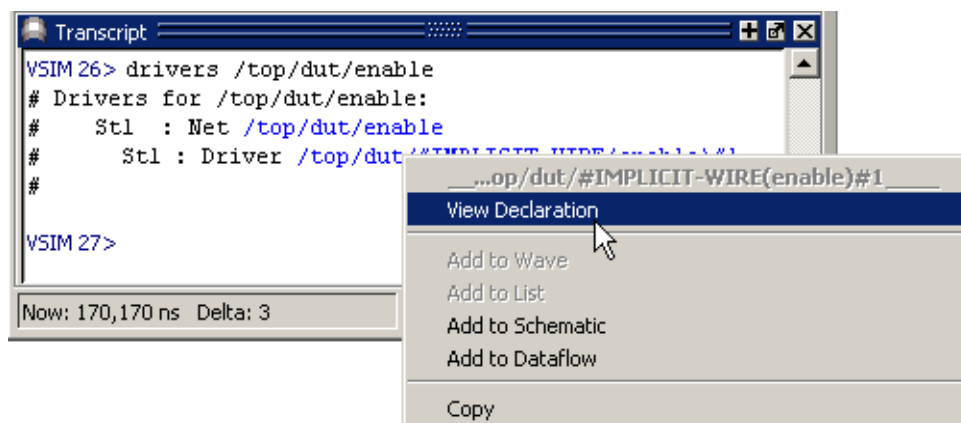
UVM Details Window

To access:

- **View > UVM Details**

- view uvmdetails command

The UVM Details Window displays information about UVM objects selected in the Structure window. The data displayed depends on the hierarchy level of the selected object, and the mode of the UVM details window. Multiple instances of this window are allowed, along with the option of following or staying fixed to the current context for each window instance. UVM objects do not exist until elaboration however, the UVM details window is able to remain open for a given UVM component that might go out of existence and then reappear as a user issues a restart to 0 followed by a run command.

Description

Simulate with vsim -uvmcontrol=all -classdebug

Elaborate the UVM testbench by executing the following command:

    run 0

You can choose between two viewing modes by clicking the mode button in the upper right corner of the window. The window header bar displays the current mode of the window, Streams or ConfigDBs, and the currently selected context. Refer to Figure 150.

Figure 150. UVM Details Window Mode Selection
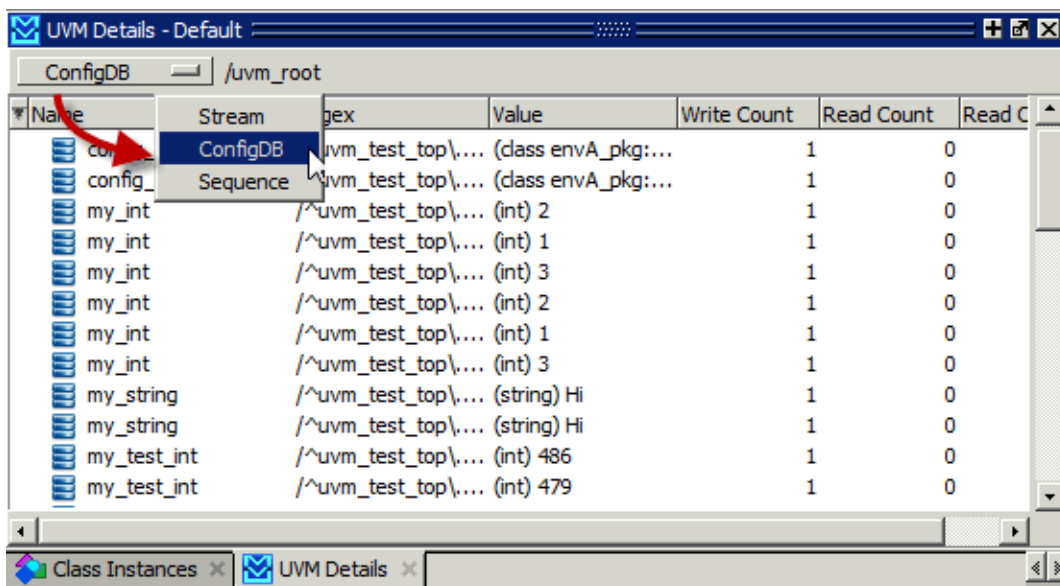
- **Stream** mode — (default) Displays any existing transaction streams that exist at or below the UVM component hierarchy selected in the Structure window. You can drag UVM stream objects into the Wave window.

- **Config DB** mode — Displays any config DB items that are accessible to the selected UVM component. Displays whether the configuration item is written to or read as well as the value and type. All of the configuration database items are listed when the uvm_root component is selected.

- **Sequence** Mode — Displays a list of sequencers and the active sequences running under them for the currently selected UVM hierarchy.

Objects

Table 114. UVM Details Window Columns

Column Title	Description
Name	name of the transaction stream or configuration database object
Regex	regular expression used to match the configuration object with a UVM component pathname
Value	value of the configuration object
Write Count	number of times the configuration object value has been written to
Read Count	number of times the configuration object has been read
Read Only	reports a value of 1 if the configuration object is marked as read-only

Table 115. UVM Details Window Streams Mode Popup Menu

Popup Menu Item	Description
Add Wave	adds the selected transaction stream(s) to the Wave Window
Add Wave New	adds the selected transaction stream(s) to a new instance of the Wave Window
Copy	copies the transaction stream(s)
Select All	selects all transaction streams in the window
Unselect All	unselects all transaction streams in the window
Find	opens the find toolbar in the active window
Environment	• Follow Context Selection • Fix to Current Context

Table 116. UVM Details Window ConfigDB Mode Popup Menu

Popup Menu Item	Description
Show Readers and Writers	prints readers and drivers of the selected variable to the transcript window
Copy	copies the selected variable(s)
Find	opens the find toolbar in the active window
Environment	• Follow Context Selection • Fix to Current Context

Table 117. UVM Details Window Sequence Mode Popup Menu

Popup Menu Item	Description
View Declaration	opens the source file and highlights the line of code where the sequence is declared
Break in Body for this instance	sets a source breakpoint on the UVM sequence body() method for this specific instance
Break in Body for any instance	sets a source breakpoint on the UVM sequence body() method for any sequence instance of this type
Show Full Path	toggles the display of the full UVM hierarchical path from the UVM Root on and off
Show UVM Path	toggles the display of the UVM leaf on and off
Copy	copies the sequence(s)
Select All	selects all sequences in the window
Unselect All	unselects all sequences in the window
Find	opens the find toolbar in the active window
Environment	• Follow Context Selection • Fix to Current Context

Table 118. UVM Details Menu

Menu Item	Description
Print Factory	prints the global UVM factory override information to the transcript window
Print Topology	prints the global UVM testbench topology information to the transcript window

Table 118. UVM Details Menu (continued)

Menu Item	Description
Display Objections	prints the current UVM objections associated with the selected UVM component and its' children to the transcript window

Table 119. UVM Details Window Icons

Icon	Description
	transaction object
	UVM configuration database object
	UVM sequencer

Verification Management Browser

To access:

- **View > Verification Management > Browser**

- view testbrowser Shell or vsim command

Use the Verification Management Browser window to display summary information for original test results in UCDBs, ranking files, and merged test results in a UCDB. You can also use it to has a feature for customizing and saving the organization of the tabs. It also supports features for re-running tests, generating HTML reports from test results, and executing merges and test ranking.

Description

For details on how the Total Coverage column statistics are calculated, refer to Calculation of Total Coverage" in the User's Manual. Coverage numbers presented in all columns are by instance, not by file.

Figure 151 shows the Verification Browser window using the Code Coverage column view setting. Refer to Controlling the Verification Browser Columns for more information.

Figure 151. Browser Tab

The Browser uses the following icons to identify the type of file loaded into the browser:

Table 120. Verification Browser Icons

Browser Icon	Description
[U]	Indicates the file is an unmerged UCDB file. A "P" notation in the upper right hand corner of the icon indicates that a Verification Plan is included in UCDB.
[R]	Indicates the file is a rank file.
[M] [M]	Indicates the file is a merged UCDB file. Notations on right hand side mean the following: • P - verification (test) plan is included in merged UCDB • 1 - Totals merge • 2 - Test-associated merge

Table 120. Verification Browser Icons (continued)

Browser Icon	Description
	Refer to Test-Associated Merge versus Totals Merge Algorithm in the User's Manual for more information.

For a description of the columns ending in "Incr", see the generic description for "<type>Incr".

The Browser allows access to the Column Layout Toolbar and the Help Toolbar.

The Verification Browser menu becomes available in the Main menu when the Verification Management Browser View window is active.

Right-click one of the UCDBs in the Browser window to display the popup menu and select one of the available options.

Objects

Table 121. Verification Management Browser Window Column Descriptions

Column Title	Description
Assertions	the number of hits from the total number of assertions, as a percentage
<type>Incr (AssertionsIncr, BranchesIncr, CoversIncr, and so on)	valid only for rows within a .rank file whose tests contribute in some measure to the overall coverage. Within those rows, the numbers indicate the incremental additional coverage contributed by the test in that row, relative to the previous test's coverage.
Branches	the number of executable branches in each level and all levels under that level
Compulsory	whether the UCDB is part of a required (compulsory) test for ranking
Covergroups	the number of hits from the total number of covergroups, as a percentage
Covers	the number of hits from the total number of cover directives, as a percentage
CPUTime	total CPU time consumed
CvgPeakMemory	the peak transient memory used by the covergroup.
CvgPeakTime	the simulation run time at which the peak transient memory usage occurred.
CvgTotalMemory	the persistent memory used by the covergroup.
Date	date the test was run
FecConditions	the number of FEC conditions in each level and all levels under that level
FecExpressions	the number of executable FEC expressions in each level and all levels subsumed under that level
OriginalFileName	UCDB filename

Table 121. Verification Management Browser Window Column Descriptions (continued)

Column Title	Description
Seed	random seed
SimTime	total simulation time
Statements	the number of executable statements in each level and all levels under that UCDB
States	the number of states encountered in each level and all levels subsumed under that UCDB
SystemC	SystemC coverage as a percentage
TestCmd	vsim command used to run simulation
TestComment	comments for the test
TestName	name of the test
TestplanCoverage	coverage percentage for that section of plan
TestStatus	status of the test
ToggleNodes	the number of points in the UCDB where the logic will transition from one state to another
TotalCoverage	the weighted average of all the coverage types (functional coverage and code coverage) is recursive. Deselect Code Coverage > Enable Recursive Coverage Sums to view results for the local instance. See "Calculation of Total Coverage" for coverage statistics details.
Transitions	the number of states encountered in each level and all levels subsumed under that level for the UCDB
UserName	name of user who ran simulation which created the UCDB
VsimArgs	arguments passed to vsim command

Table 122. Verification Browser View Menu

Browser View Menu Item	Description
Add File	Adds UCDB (.ucdb) and ranking results (.rank) files to the browser. Refer to Viewing Test Data in the GUI in the User's Manual for more information.
Remove File	Removes an entry from this window (**From Browser Only**), as well as from the file system (**Browser and File System**).
Remove Non-Contributing Test(s)	Operates only on ranked (.rank) files; menu selection is grayed out unless a ranked file is selected. Removes any tests that do not contribute toward the coverage.

Table 122. Verification Browser View Menu (continued)

Browser View Menu Item	Description
Testplan Import	Displays the XML Testplan Import Dialog Box, which allows you to import an XML testplan file into a UCDB which can then be merged with test results. Refer to Importing an XML Verification Plan in the Verification Management User's Manual.
Merge	Displays the Merge Files Dialog Box, which allows you to merge any selected UCDB files. Refer to Merging Coverage Data in the User's Manual for more information.
Rank	Displays the Rank Files Dialog Box, which allows you to create a ranking results file based on the selected UCDB files. Refer to Ranking Coverage Test Data in the User's Manual for more information.
HTML Report	Displays the HTML Coverage Report Dialog Box, which allows you to view your coverage statistics in an HTML viewer.
Command Execution	Allows you to re-run simulations based on the resultant UCDB file based on the simulation settings to create the file. You can rerun any test whose test record appears in an individual *.ucdb* file, a merged *.ucdb* file, or ranking results (*.rank*) file. Refer to Test Attribute Records in the UCDB in the User's Manual for more information on test records. • Setup — Displays the Command Setup Dialog box, which allows you to create and edit your own setups which can be used to control the execution of commands. "Restore All Defaults" removes any changes you make to the list of setups and the associated commands. • Execute on all — Executes the specified command(s) on all *.ucdb* files in the browser (through **TestReRun**), even those used in merged *.ucdb* files and *.rank* files. • Execute on selected — Executes the specified command(s) on the selected *.ucdb* file(s) through TestReRun.
Filter	Either opens the Filter Setup Dialog Box, or applies desired filter setups. • Setup — opens the Filter Setup dialog that allows you to save and edit filters to apply to the data. • Create button — opens the Create Filter dialog which allows you to select filtering criteria, and select the tests for application of the specified filters. • Apply — applies the selected filter(s) on the data.
Generate RMDB File	Generates Verification Run Management configuration file (*.rmdb*) including selected tests or all tests in the directory. Selecting either option brings up a dialog to enter the name to be used for the *.rmdb* file. • Save Selected Tests — Saves selected tests into a *.rmdb* or *.tcl* file to be executed by vrun command. • Save All Tests — Saves all tests in the directory into a *.rmdb* or *.tcl* file to be executed by vrun command.

Table 122. Verification Browser View Menu (continued)

Browser View Menu Item	Description
Show Full Path	Toggles whether the FileName column shows only the filename or its full path.
Set Precision	Controls the decimal point precision of the data in the Verification Browser window.
Trend Analysis	Active only when a Trend UCDB is selected, or when multiple UCDBs are selected. Allows access to trending functionality. • HTML Report — opens the Coverage Trend Report (HTML) dialog box, where you set: the HTML report's colorization threshold; the output directory path; and other HTML reporting options. See vcover report -html for detailed description of options. • XML Report — opens the Coverage Trend Report (XML) dialog box, where you set whether you are reporting on all (or specified) DUs, testplans, or instances; filtering; the coverage type; and the report pathname. • Text Report — opens the Coverage Trend Report (Text) dialog box, where you set whether you are reporting on all (or specified) DUs, testplans, or instances; filtering; the coverage type; and the report pathname. • Create Trend Database — opens the Create Trend Database dialog box, where you enter the name of the output trend database (the "Trend UCDB") and input UCDBs. • Export — opens the Coverage Trend Report Export dialog box, where you set the options as shown for XML or HTML report, and export the report into a comma separated value (csv) or other format file. • View Trender — opens the Trender window in the Main window. In this window, right click any coverage object to display a trend graph for that object.

Table 123. Verification Management Browser Window Popup Menu

Popup Menu Item	Description
Expand / Collapse Selected	Expand or collapse selected UCDBs.
Expand / Collapse All	Expand or collapse all UCDBs.
Save Format	Saves the current contents of the browser to a *.do* file.
Load	Loads a *.do* file that contains a previously saved browser layout.

Usage Notes

Controlling the Verification Browser Columns

You can customize the appearance of the Browser using either of the following methods:

- Use the Column Layout Toolbar to select from several pre-defined column arrangements.

- Right-click in the column headings to display a list of all column headings which allows you to toggle the columns on or off.

Saving Verification Browser Column and Filter Settings

Save your column layout and any filter settings to an external file (*browser_column_layout.do*) by choosing **File > Export > Column Layout** while the window is active. You can reload these settings with the do command. This export does not retain changes to column width.

Viewing Verification Browser Data in HTML

You can Export the view of the columns and verification data in the Browser, including any filter settings, to an HTML file (*browser.htm*) using the by choosing **File > Export > HTML** while the window is active. A dialog opens in which you can set both the name of the file and the directory to which it is saved. By default, the HTML file is saved into the current directory.

Verification Results Analysis Window

To access:

- **File > Open > Results Analysis Files (*.tdb)**

- Transcript window's vsim prompt: "triage view -name <tdb-filename>.tdb"

- Shell prompt: vsim <tdb-filename>.tdb

The Verification Results Analysis window is used for viewing and sorting message and test data information. It requires a "qvman" license to view.

Description

Figure 152 shows an example of the Verification Results Analysis window with a results analysis (triage) database, *top.tdb,* loaded.

Note:
You must have a TDB to view the Verification Results Analysis window. Refer to Creating a Results Analysis Database (TDB) in the Verification Management User's Manual.

Figure 152. Results Analysis Tab

Objects

Table 124. Results Analysis Window Columns

Column	Description
Assertion Expression	Assertion expression associated with the message
Assertion Filename	Name of the file where the assertion failure message originated
Assertion Line Number	Line number within the filename where the message originated

---

Table 124. Results Analysis Window Columns (continued)

Column	Description
Assertion Name	Name of the assertion associated with the message
Assertion Start Time	Start time of the assertion associated with the message
Category	Keyword for the various categories of messages: • DISPLAY • FLI • PA • PLI • SDF • TCHK • VCD • VITAL • WLF • MISC • <user-defined>
Comment	User comment
Compulsory	Whether the item is part of a required (compulsory) test for ranking
Count	Date the test was run (in UCDB format)
CPU Time	Total CPU time consumed
Date	Date the test was run
File Info	Filename related to the cause of the message, and in some cases the line number in parentheses
Host OS	Operating system in use by the host on which the test was run
Hostname	Name of host (server) on which the test was run
instance	Instance or region associated with the message
Iteration/Delta	Iteration (delta) in which the message occurs
LOG name	Name (path) to the generated log/transcript file
MEMUSAGE	Total memory used by the simulator for the test
Message	Organized tree-structure of the sorted messages, as well as, when expanded, the text of the messages

Table 124. Results Analysis Window Columns (continued)

Column	Description
Message ID	Message ID
Message ID Name	Message ID name
Process	Process or leaf associated with the message
Region	Hierarchical region related to the message, if any
run CWD	Directory in which the test was run
Seed	Random seed
Severity	Message severity, such as Warning, Note or Error
Sim Time	Total simulation time
sim Timeunits	Timeunit used by the simulation
Source File Name	Name of the file where the message originated
Source File Number	Declaration number of the file associated with the message
Source Line Number	Line number within "filename" where the message originated
Test Args	Application command used to generate the coverage data if the data was not generated by vsim, similar to how Vsim Args operates for vsim commands
Test Name	Name of the test
Test Status	Completion status (OK, Error, and so on)
Time	Time of simulation when the message was issued.
Timing Check Kind	Information about timing checks
User ID	Username under which the test was run
Verbosity	Verbosity information from $messagelog system tasks.
VRM Context	Username under which the test was run
Vsim Args	Arguments passed to vsim command
WLF Filename	Name of WLF file from which message was imported
WLF Name	Name (path) to the generated WLF file
WLF Raw Time	Simulation time (in ticks) associated with the message
WLF Time Unit	Simulation time unit

Table 125. Results Analysis Window Popup Menu

Popup Menu Item	Description
Modify Database	Opens the Modify Results Analysis Database dialog box, the fields in which correspond to switches and arguments in the triage dbfile command.
Rebuild Database	Rebuilds the TDB file from the initial UCDB/WLF source files with the triage dbfile command used initially.
Reload Viewer Data	Refreshes the display of TDB data in the viewer, reapplying filters and sorting/hierarchy configuration settings.

View
  • Verbose Message
  • Message Source
  • Log File
  • Object Declaration
  • Assertion Info
  • Go to Time in Wave
  • Add to Dataflow
	Opens selected item:
	Verbose Message dialog box with details about message.
	Source code at line number where message is
	Log file, in a Source window
	Object window, to view object declarations
	Assertion window, to view assertions
	Waveform window, at time of message
	Adds selected item to Dataflow window for viewing.

Run Manager • Save Tests to File	Opens Save Test List dialog for saving values in VRM Context column for the messages selected into a file.
Analysis Questions	Lists specific queries of the data. The [Configure Questions...] selection opens the Analysis Questions dialog box; used for saving and managing specific queries, including specified hierarchy, filter and column settings. Any configurations set here override those set in these other individual layout/configuration settings.
Column Layouts	Lists column layouts. The [Configure Columns...] selection opens the Configure Columns Layout dialog; used for creating, editing and managing the configuration of columns.
Filter Expressions	Lists filters available for use. The [Configure Filters...] selection opens the Filter Expressions dialog; used for creating, saving and managing filters. Refer to Filter Expressions Dialog Box for more information.
Hierarchy Configurations	Lists hierarchy configurations to select. The [Configure Hierarchy...] opens Hierarchy Configurations dialog box; used for creating, saving and managing particular hierarchy configurations of the data.
Sort Configurations	Lists sort configurations to select. The [Configure Sorting...] opens Sort Configurations dialog box; used for creating, saving and managing sort configurations of the data.
Show Titles in Hier Column	Toggles on and off displaying titles within the hierarchy column.
Global Options	Opens a dialog controlling the display of the viewer and the data.

Table 125. Results Analysis Window Popup Menu (continued)

Popup Menu Item	Description
Edit Transforms...	Opens a dialog which will open a transform file for editing with the Transform Rule File Editor.
Load/Save Setup File	Loads/Saves a particular setup to a name you specify.
Expand/Collapse Selected/All	Manipulates the expansion of the Messages column.

Usage Notes

Controlling the Results Analysis Columns

You can customize the appearance of the columns in the Results Analysis window by right-clicking in the column headings and choosing **Select Column Visibility** to display a list of all column headings which allows you to toggle the columns on or off.

Dragging Information from RA Window

In some cases, it can be helpful to drag and drop contents from a column in the Results Analysis window to the Transcript window. For example, you might begin a command in the Transcript, and drag the instance path, message number, and so on from the Results Analysis window to make the typing task less tedious.

Results Analysis Configuration of Data

The Results Analysis window contains a "drawer" of options for configuring the data, including analysis questions, column layouts, filter expressions, hierarchy of data, and sort configurations. The "drawer", where all these settings can be set in one convenient location, is opened with a small toggle button at the bottom of the window.

Figure 153. Configuration Options for Results Analysis
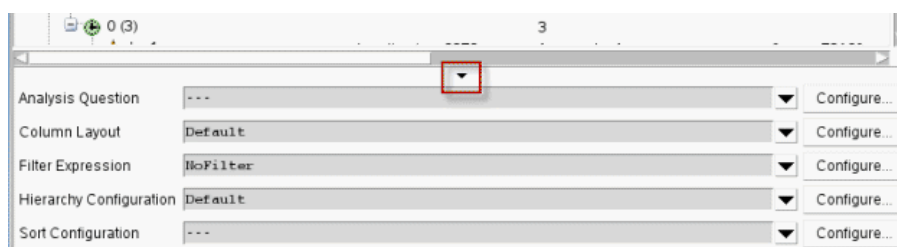
Saving Custom Hierarchy Configuration

To save your own custom column layout and any filter settings to an external file (*triageviewer_hier_config.do*), choose **File > Export > Hierarchy Configuration** while the window is active. You can reload these settings with the do command. This export does not retain changes to column width.

Verification Test Analysis Window

To access:

- In the **Verification Tracker** window or **Covergroups** window, right click any item in the window, and choose a **Test Analysis >** menu item.

- In the **Assertions** or**Cover Directives** window, right click any item in the window, and choose a **Test Analysis >** menu item.

- view test analysis command

The Verification Test Analysis window is used to perform several test and coverage analysis query functions. The functionality is also available through the command line with the coverage analyze command.

Note:
This window requires a "qvman" license.

Description

Refer to Calculation of Total Coverage in the User's Manual for coverage statistics calculation details.

Figure 154. Verification Test Analysis Window

Column Descriptions

The columns that appear in the Test Analysis window are the same as those in the Verification Management Browser window. They can be either added to the view or removed from view by selecting the Configure Column dropdown icon to the left of the first column name.

Data contained in the columns is can be filtered using RMB > Filter > Setup/Apply. For a description of how to set and apply filters, Refer to Filtering Results by User Attributesin the User's Manual.

The menu items are related to the Verification Test Analysis window, and are based on the coverage analyze command.

Objects

Table 126. Verification Test Analysis Window Menu Items

Menu Item	Description
Merge	displays the Merge Files Dialog Box, which allows you to merge any selected UCDB files. Refer to Merging Coverage Data in the User's Manual for more information.
Rank	displays the Rank Files Dialog Box, which allows you to create a ranking results file based on the selected UCDB files. Refer to Ranking Coverage Test Data in the User's Manual for more information.
HTML Report	displays the HTML Coverage Report Dialog Box, which allows you to view your coverage statistics in an HTML viewer.
Command Execution	allows you to re-run simulations based on the resultant UCDB file based on the simulation settings to create the file. You can rerun any test whose test record appears in an individual *.ucdb* file, a merged *.ucdb* file, or ranking results (*.rank*) file. Refer to Test Attribute Records in the UCDB in the User's Manual for more information on test records. • Setup — Displays the Command Setup Dialog box, which allows you to create and edit your own setups which can be used to control the execution of commands. "Restore All Defaults" removes any changes you make to the list of setups and the associated commands. • Execute on all — Executes the specified command(s) on all *.ucdb* files in the browser (through TestReRun), even those used in merged *.ucdb* files and *.rank* files. • Execute on selected — Executes the specified command(s) on the selected *.ucdb* file(s) through TestReRun.
Filter	either opens the Filter Setup Dialog Box, or applies desired filter setups. • Setup — opens the Filter Setup dialog that allows you to save and edit filters to apply to the data. • Create button — opens the Create Filter dialog which allows you to select filtering criteria, and select the tests for application of the specified filters. When you enter a Filter Name, and select Add, the Add/Modify Selection Criteria dialog box is displayed, where you can select the actual criteria to filter. Refer to Filtering Results by User Attributes in the User's Manual for an example. • Apply — applies the selected filter(s) on the data.
Generate Vrun Config	generates Verification Run Management configuration file (*.rmdb*) including selected tests or all tests in the directory. Selecting either option brings up a dialog to enter the name to be used for the *.rmdb* file. • Save Selected Tests — Saves selected tests into a *.rmdb* file to be executed by vrun command. • Save All Tests — Saves all tests in the directory into a *.rmdb* file to be executed by vrun command.
Show Full Path	toggles whether the FileName column shows only the filename or its full path.

Table 126. Verification Test Analysis Window Menu Items (continued)

Menu Item	Description
Set Precision	allows you to control the decimal point precision of the data in the Verification Browser window.

Verification Tracker Window

To access:

- **View > Verification Management > Tracker**

- view testtracker command

Use the Verification Tracker window to view and analyze test-associated merged data. The functionality is also available through the command line with the coverage analyze command.

> **Note:**
> This window requires a "qvman" license.

Description

This window is specific to the analysis of testplan coverage metrics. Therefore, you must have a verification testplan, you must have run your simulation with coverage collection enabled, and you must have merged the results with your testplan UCDB.

Requirements for viewing data in this window are that you must:

- Have a merged file (UCDB) containing testplan and test results, with testplan sections linked to coverage items, either automatically or manually:

  - automatically — When testplan is created with the Excel or OpenOffice Calc Add-In for testplan creation or during merge if the testplan UCDB was generated by either the Testplan Import facility or the xml2ucdb command

  - manually — using the coverage tag command

- Load the testplan merged file in Coverage View mode, either with vsim -viewcov or the GUI: **Verification Browser > Invoke Coverage Mode**.

For more detailed information, refer to Merging Verification Plan with Test Data in the Verification Management User's Manual.

Figure 155. Verification Tracker Window

The view inside the Verification Management Tracker window is of a testplan and all its descendant sections with their linked coverage items, including assertions, cover directives, covergroup bins (including virtual bins), coverpoints, and crosses. Virtual covergroup bins are identified by an orange "B" icon. The creation of these virtual bins relate to the Unimplemented column, and assure that coverage calculations appropriately reflect the participation of sections which are either missing from the plan or are not fully implemented.

Any scopes and descendant coverage items which do not contribute to the coverage calculations are greyed out in the Tracker GUI, as well as the HTML report.

Menu items related to the Verification Tracker window are primarily based on the coverage analyze command. When you select a section in the testplan and apply the menu items, a new window opens displaying the tests with the least, most, zero, and so on, coverage.

Objects

- Menu items.

  ◦ **View Source** — valid when any coverage item is selected. Grayed out when no source is available. Opens up source file for coverage item.

  ◦ **View Coverage Details** — valid when any covergroup, coverpoint, cross, assertion and or cover directive is selected in the testplan. Opens up a new tab of that item's coverage type, highlighting the coverage item, where further coverage details can be viewed.

  ◦ **Configure Selected...** — displays the Coverage Configurations Dialog Box, which allows you to set the goal and weight for the specified test(s).

  ◦ **Test Analysis** — analyzes the results to:

    ◦ Find Tests with Least Coverage

    ◦ Find Tests with Most Coverage

    ◦ Find Tests with Zero Coverage

    ◦ Find Tests with Non-Zero Coverage

    ◦ Rank Most Effective Test - Opens a dialog where you can select ranking criteria to apply.

  ◦ **Find Unlinked** — reports on testplan sections that are unlinked with coverage; or, conversely, design or coverage objects which have not been linked with a testplan. Refer to Finding Unlinked Items in the Verification Management User's Manual.

    ◦ Testplan Section — finds unlinked testplan sections.

    ◦ Functional Coverage — finds unlinked functional coverage items.

  ◦ **Filter** — either opens the Filter Setup Dialog Box, or applies desired filter setups.

○ Setup — opens the Filter Setup dialog that allows you to save and edit filters to apply to the data.

- Create button — opens the Create Filter dialog which allows you to select filtering criteria, and select the tests for application of the specified filters. When you enter a Filter Name, and select Add, the Add/Modify Selection Criteria dialog box is displayed, where you can select the actual criteria to filter. Refer to Filtering Results by User Attributes in the User's Manual for an example.

○ Apply — applies the selected filter(s) on the data.

○ **Summary** — displays the Tracker Summary Dialog Box, which allows you to create a summary report based on the selected UCDB files. Corresponds to the coverage analyze command's -summary argument.

○ **Re-import Testplan and Refresh** — re-imports selected verification plan and refreshes the contents. Refer to Refreshing Tracker after Changing a Plan in the Verification Management User's Manual for more details.

○ **Set Precision** — allows you to control the decimal point precision of the data in the Verification Browser window.

○ **Use CrossPrintMissing** — uses the value of option.cross_num_print_missing as the display limit for uncovered bins of covergroup cross scopes. By default, all bins are shown.

○ **Show Zero Weight Objects** — controls the display of any items with zero weights, including all zero-weighted coverage items and testplan sections. Has no effect on coverage numbers. By default, zero weight objects are displayed.

○ Hide Excluded Objects — controls the display of any excluded coverage items. Has no effect on coverage numbers. By default, excluded items are displayed.

○ **Expand / Collapse Selected** — Expand or collapse selected UCDBs.

○ **Expand / Collapse All** — Expand or collapse all UCDBs.

Table 127. Verification Management Tracker Window Column Descriptions

Column Title	Description
%Hit	total percentage of bins hit
% of Goal	lists the percentage of goal reached by the section of plan. By default, this number is calculated hierarchically: • For parent testplans without a Goal, this number is defined as weighted average of child "% of Goal" values. Any linked coverage items in the parent testplan scope are included in that calculation,

Table 127. Verification Management Tracker Window Column Descriptions (continued)

Column Title	Description
	where all the coverage items are considered together as a single entity for contributing to the parent testplan's "% of Goal" • For testplans with a Goal (either parent or leaf), "% of Goal" is simply Coverage / Goal
Bins	total number of bins in item
Class Type	lists parameterized class specialization attributes of linked coverage scopes
Coverage	lists coverage percentage for that section of plan: refer to Calculation of Total Coverage in the User's Manual for coverage statistics calculation details.
Coverage graph	graphical illustration of percentage of coverage
Description	a description of section in plan
Formal	the formal object associated with a testplan rule
Goal	lists the goal percentage for that section in plan; missing Goal value is displayed as "-". Double-click value to enter a new value. When you explicitly override the Goal of a testplan, "% of Goal" is defined as Coverage / Goal. Results from children are ignored
Hits	number of bins in item that were hit
Sec#	# of the section in plan
Testplan Section / Coverage Link	testplan section name or the name of the linked coverage item. Linked covergroup, coverpoint, and cross coverage scopes are expandable to the bin level
Type	the type for that section, from the Type column in plan
Link Status	lists status of the link from testplan section to coverage item: • Clean: All child testplan sections are properly linked, all local links are clean • Unlinked: There are no successfully linked child testplan sections or local links • Error: Link errors were detected during the UCDB merge process • Partial: One or more descendants has an Unlinked or Error status which needs attention
Weight	lists the weight assigned. Double-click value to enter a new value.

Usage Notes

Customizing Columns

By default, the Testplan column layout is used. However, you can customize the columns that appear in the Tracker window:

- Right-click in the column headings to display a list of all column headings.

- Toggle the columns on or off.

Creating Column Layouts

You can also create custom column layouts which will preserve the columns you commonly use. Refer to Viewing and Analyzing Verification Data in the User's Manual for information on using this window to analyze your merged testplan results.

Saving Verification Tracker Column and Filter Settings

Save your column layout and any filter settings to an external file (*tracker_column_layout.do*) by selecting **File > Export > Column Layout** while the window is active. You can reload these settings with the do command. This export does not retain changes to column width.

Viewing Verification Tracker Data in HTML

You can export the view of the columns and verification data in the Tracker, including any filter settings, to an HTML file (*tracker.htm*) using the by selecting **File > Export > HTML** while the window is active. A dialog opens in which you can set both the name of the file and the directory to which it is saved. By default, the HTML file is saved into the current directory.

Verification Trender Window

To access:

- Highlight one or more UCDBs

- **Trend Analysis > View Trender**

- view trender command

The Verification Trender window is used for analyzing coverage trends in verification. This window has a feature for user-customizable column headings. It requires a "qvman" license to view.

Description

Figure 156. Trender Window
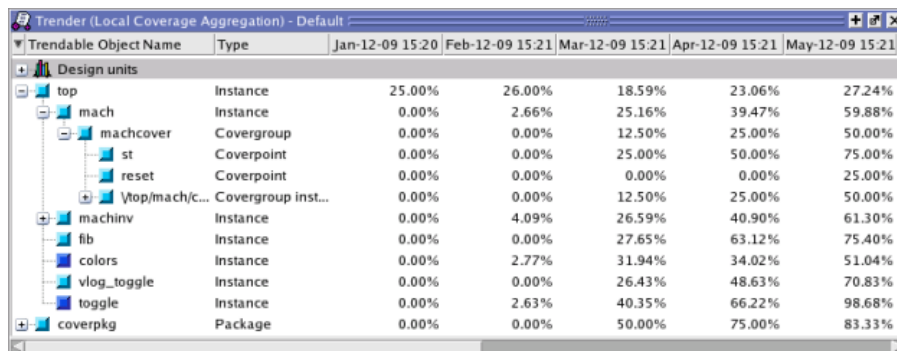
The columns that appear in the Trender window are the object name, the type of object and the dates when the data was saved. Right-click in the column headings to display a list of all column layout and configuration options. You can these options to create custom column layouts which will preserve the columns you commonly use.

Objects

Table 128. Trender Window Popup Menu

Popup menu Item	Description
View Trend Graph Using Local Coverage	Opens a Trend graph with local (default) coverage numbers
View Trend Graph Using Recursive Coverage	Opens a Trend graph with local (default) coverage numbers
Select Coverage and View Graph	Opens a Trend graph for selected coverage
Enable Recursive Coverage Sums	Shows recursively aggregated coverage numbers (local coverage is default)

Questa SIM GUI Reference Manual

Watch Window

To access:

- **View > Watch**

- view watch command

Use the Watch window to view values for signals and variables at the current simulation time and to explore the hierarchy of object oriented designs.

Description

Unlike the Objects or Locals windows, the Watch window allows you to view any signal or variable in the design regardless of the current context. You can view the following objects:

- VHDL objects — signals, aliases, generics, constants, and variables.

- Verilog objects — nets, registers, variables, named events, and module parameters.

- SystemC objects — primitive channels and ports.

- Virtual objects — virtual signals and virtual functions.

The primary graphical element of the watch window is the item box, which typically shows information about a single signal. The item can also be a group of signals created with the "group" popup window option.

Figure 157. Watch Window Item Box
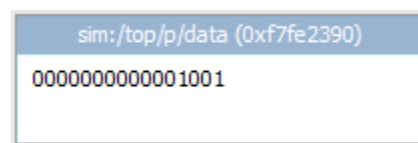
- **Header** — shows the signal name, followed, parenthetically, by its address (if it has one).

- **Body** — shows the current value of the signal. Values that are red have changed since the previous run command.

- **SystemVerilog Classes** — Items are displayed in a scrollable, hierarchical list, such as in Figure 158 on page 286 where extended SystemVerilog classes hierarchically display their super members.

- **Current Time Label** — Displays the Current Time or the Now (end of simulation) time. This is the time used to control state values annotated in the window. (For details, refer to Current Time Label.)

Figure 158. Scrollable Hierarchical Display
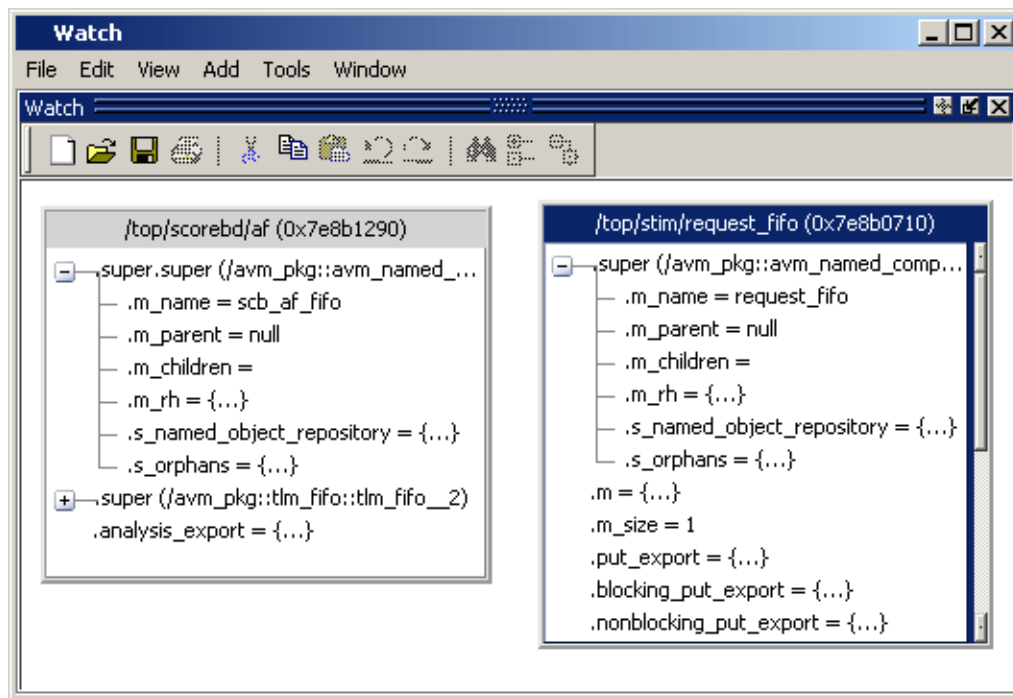
Two Ref handles that refer to the same object will point to the same Watch window box, even if the name used to reach the object is different. This means circular references will be drawn as circular.

This Watch Window menu becomes available in the Main menu when the Watch window is active.

Objects

Table 129. Watch Window Popup Menu

Popup menu Item	Description
Add	Add the selected item or items to the desired window
Force	Opens the Force Selected Signal dialog box, which allows you to force the signal to given value. Refer to the force command for details about the options.
NoForce	Removes the force on the selected object. Refer to the noforce command.
Clock	Performs actions related to the force command with the -repeat argument.
Change	Performs actions related to the change command.
Follow Selection Context	Changes the current context in the Structure window.
Save Format / Load Format	Saves a new (or loads an existing) .do file containing add watch commands to recreate the Watch window.

Table 129. Watch Window Popup Menu (continued)

Popup menu Item	Description
Group	Groups several selected signals into a single item.
UnGroup	Breaks a previously created group back into its individual signals
Properties	Opens the Properties dialog box, which allows you to alter the properties of the selected signal or group, including: • Header name • Radix type This option is not available when multiple signals are selected.
Delete Item	Removes the selected signal from the window. You can alternatively use the delete key.
Clear All	Removes all signals from the window.

Table 130. Watch Window Menu

Popup menu Item	Description
Force	Opens the Force Selected Signal dialog box, which allows you to force the signal to given value. Refer to the force command for details about the options.
NoForce	Removes the force on the selected object. Refer to the noforce command.
Clock	Performs actions related to the force command with the -repeat argument.
Change	Performs actions related to the change command.
Follow Selection Context	Changes the current context in the Structure window.
Save Format / Load Format	Saves a new (or loads an existing) .do file containing add watch commands to recreate the Watch window.
Group	Groups several selected signals into a single item.
UnGroup	Breaks a previously created group back into its individual signals
Tile	Reorganizes the items in the Watch window into different tiled formats.
Clear All	Removes all signals from the window.

Usage Notes

This section describes tasks for using the Watch window.

Adding Objects to the Watch Window

To add objects to the Watch window, drag -and-drop objects from the Structure window or from any of the following windows: List, Locals, Objects, Source, and Wave. You can also use the Add Selected to Window Button. You can also use the add watch command.

Expanding Objects to Show Individual Bits

If you add an array or record to the window, you can view individual bit values by double-clicking the array or record. As shown in Figure 159 on page 288, /ram_tb/spram4/mem has been expanded to show all the individual bit values. Notice the arrow that ties the array to the individual bit display.

Figure 159. Expanded Array

Wave Window

To access:

- Drag and drop.

- Click the middle mouse button when the cursor is over an object or group of objects in the Objects or Locals windows.

- Click-and-hold the Add Selected to Window Button to specify where selected signals are placed: above the Insertion Point Bar in the Pathnames Pane, appended after the Insertion Pointer in the Pathnames Pane, at the top or the end of the Pathnames Pane.

- add wave command

The Wave window, like the List window, allows you to view the results of your simulation. In the Wave window, however, you can see the results as waveforms and their values.

Description

Figure 160. Wave Window

For information about using this window with a Power Aware simulation, refer to "Power Aware Waveform Display" in the Power Aware Simulation User's Manual.

When you drag and drop objects into the Wave window, the add wave command is reflected in the Transcript window.

The Wave window is divided into a number of window panes. All window panes in the Wave window can be resized by clicking and dragging the bar between any two panes.

Pathname Pane

The pathname pane displays signal pathnames. Signals can be displayed with full pathnames, as shown here, or with any number of path elements. You can increase the size of the pane by clicking and dragging on the right border. The selected signal is highlighted, and a highlighted area appears in the scrollbar — on the right hand side of the Wave window. The default color for the scrollbar highlight is white. You can change the color of the scrollbar highlight by setting the Wave window mapSelectColor preference. See Setting GUI Preferences.

The white bar along the left margin indicates the selected Wave window or pane of a split wave window (refer to Splitting Wave Window Panes in the User's Manual).

Insertion Point Bar

You can select the location for inserting signals by placing the cursor over the left white bar in the Pathnames Pane. The white arrow and green bar indicate the selected location for the insertion pointer. Clicking the left mouse button sets the new insertion pointer.

Figure 161. Pathnames Pane
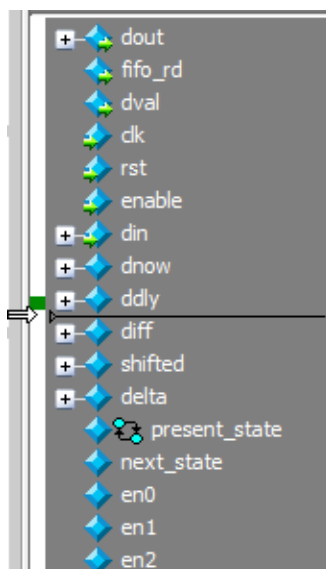
Values Pane

The values pane displays the values of the displayed signals. You can resize the values pane by clicking and dragging the right border. Some signals may be too wide (too many bits) for their values to be fully displayed. Use the scroll bar at the bottom of the pane to see the entire signal value. Small signal values will remain in view while scrolling.

The radix for each signal can be symbolic, binary, octal, decimal, unsigned, hexadecimal, ASCII, or default. The default radix for all signals can be set by choosing **Simulate > Runtime Options**.

Note:
When the symbolic radix is chosen for SystemVerilog reg and integer types, the values are treated as binary. When the symbolic radix is chosen for SystemVerilog bit and int types, the values are considered to be decimal.

To change the radix for just the selected signal or signals, choose **Wave > Format > Radix > Global Signal Radix** from the menus, or right-click the selected signal(s) and choose **Radix > Global Signal Radix** from the popup menu. This opens the Global Signal Radix dialog box (Figure 162 on page 291), where you may select a radix. This sets the radix for the selected signal(s) in the Wave window and every other window where the signal appears.

Figure 162. Setting the Global Signal Radix from the Wave Window
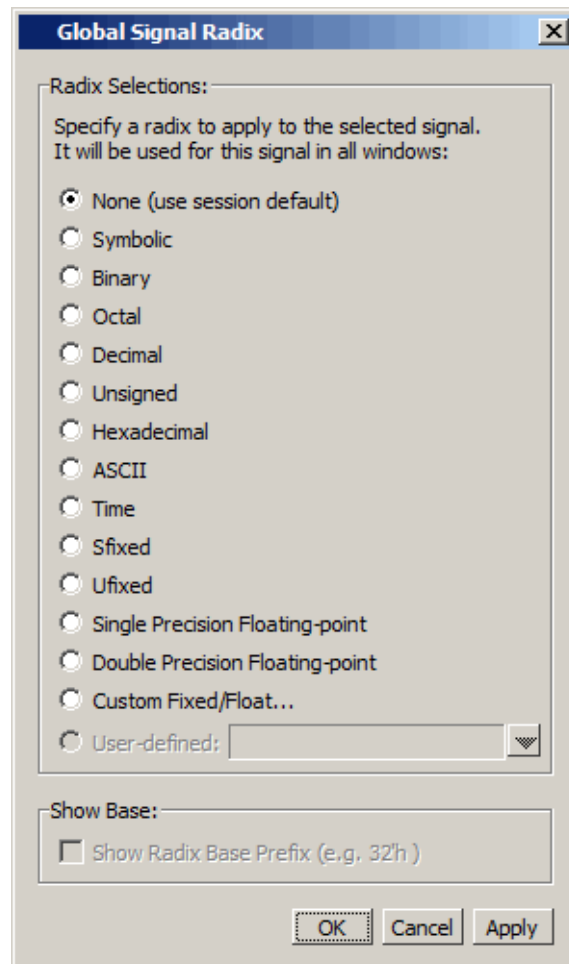
The data in this pane is similar to that shown in the Objects Window, except that the values change dynamically whenever a cursor in the waveform pane is moved.

Figure 163. Values Pane

Waveform Pane

Figure 164 shows waveform pane, which displays waveforms that correspond to the displayed signal pathnames. It can also display as many as 20 user-defined cursors. Signal values can be displayed in analog step, analog interpolated, analog backstep, literal, logic, and event formats. You can set the format of each signal individually by right-clicking the signal in the pathname or values panes and choosing **Format** from the popup menu. The default format is Logic.

If you place your mouse pointer on a signal in the waveform pane, a popup menu displays with information about the signal. You can toggle this popup on and off in the **Wave Window Properties** dialog box.

Dashed signal lines in the waveform pane indicate weak or ambiguous strengths of Verilog states.

Figure 164. Waveform Pane
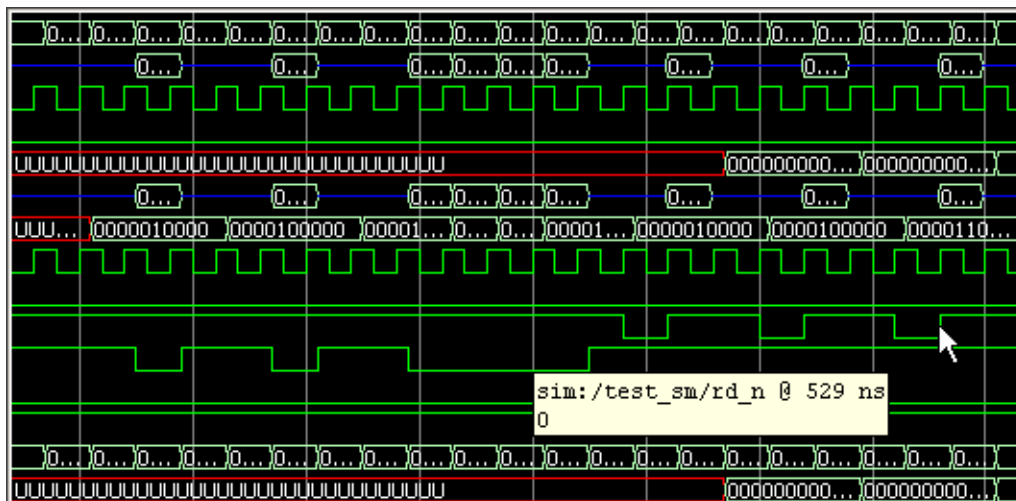
Analog Sidebar Toolbox

When the waveform pane contains an analog waveform, you can hover your mouse pointer over the left edge of the waveform to display the Analog Sidebar toolbox. This toolbox shows a group of icons that gives you quick access to actions you can perform on the waveform display.

Figure 165. Analog Sidebar Toolbox

Cursor Pane

The Cursor Pane displays cursor names, cursor values and the cursor locations on the timeline. You can link cursors so that they move across the timeline together. Refer to Linking Cursors in the User's Manual.

Figure 166. Cursor Pane
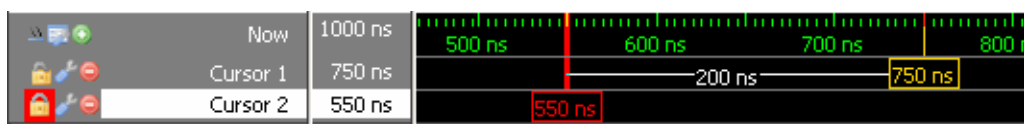
On the left side of this pane is a group of icons called the Cursor and Timeline Toolbox (refer to Working with Cursors in the User's Manual). This toolbox gives you quick access to cursor and timeline features and configurations. Refer to Measuring Time with Cursors in the Wave Window in the User's Manual for more information.

Messages Bar

The messages bar, located at the top of the Wave window, contains indicators pointing to the times at which a message was output from the simulator. By default, the indicators are not displayed. To turn on message indicators, use the -msgmode argument with the vsim command or use the msgmode variable in the *modelsim.ini* file.

Figure 167. Wave Window - Message Bar

The message indicators (the down-pointing arrows) are color-coded as follows:

- Red — Indicates an error.

- Yellow — Indicates a warning.

- Green — Indicates a note.

- Grey — Indicates any other type of message.

You can use the Message bar in the following ways.

- Move the cursor to the next message — You can do this in two ways:

  ◦ Click the word "Messages" in the message bar to cycle the cursor to the next message after the current cursor location.

  ◦ Click anywhere in the message bar, then use Tab or Shift-Tab to cycle the cursor between error messages either forward or backward, respectively.

- Display the Message Viewer Window — Double-click anywhere amongst the message indicators.

- Display, in the Message Viewer window, the message entry related to a specific indicator — Double-click any message indicator.

  This function only works if you are using the Message Viewer in flat mode. To display your messages in flat mode:

  a. Right-click in the Message Viewer and select **Display Options**

  a. In the Message Viewer Display Options dialog box, deselect **Display with Hierarchy**.

View Objects Window Button

This button opens the Objects window with a single click. However, if you click-and-hold the button you can access additional options via a dropdown menu.

Figure 168. View Objects Window Dropdown Menu
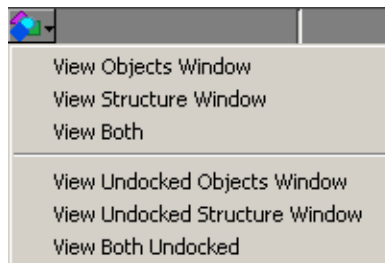
Assertions Debug Pane

The Assert Debug pane displays details on PSL and SVA assertion failures. See Analyzing Assertion Failures in the Assertion Debug Pane of the Wave Window for more information.

Objects You Can View in the Wave Window

The following types of objects can be viewed in the Wave window

- VHDL objects (indicated by a dark blue diamond) — signals, aliases, process variables, and shared variables

- Verilog objects (indicated by a light blue diamond) — nets, registers, variables, and named events

  The GUI displays inout variables of a clocking block separately, where the output of the inout variable is appended with "__o", for example you would see following two objects:

```
clock1.c1        /input portion of the inout c1
clock1.c1__o      /output portion of the inout c1
```

This display technique also applies to the Objects window

- Verilog and SystemVerilog transactions (indicated by a blue four point star)

- SystemC objects. Primitive channels and ports (indicated by a green diamond) and transaction streams and their element (indicated by a green four point star).

- Virtual objects (indicated by an orange diamond) — virtual signals, buses, and functions. Refer to Virtual Objects in the User's Manual for more information

- Comparison objects (indicated by a yellow triangle) — comparison region and comparison signals; see Waveform Compare for more information

- SystemVerilog and PSL assertions (indicated by a light-blue (SVA) or magenta (PSL) triangle) — see Viewing Assertions and Cover Directives in the Wave Window

- SystemVerilog and PSL cover directives (indicated by a light-blue (SVA) or magenta (PSL) chevron) — see Viewing Assertions and Cover Directives in the Wave Window

- Questa Verification IP objects (see Questa Verification IP Objects in the GUI) — see Questa Verification IP Transaction Viewing in the GUI for more information

- Created waveforms (indicated by a red dot on a diamond) — see Generating Stimulus with Waveform Editor

The data in the object values pane is very similar to the Objects window, except that the values change dynamically whenever a cursor in the waveform pane is moved.

At the bottom of the waveform pane you can see a time line, tick marks, and the time value of each cursor's position. As you click and drag to move a cursor, the time value at the cursor location is updated at the bottom of the cursor.

You can resize the window panes by clicking the bar between them and dragging the bar to a new location.

Waveform and signal-name formatting are easily changed via the Format menu. You can reuse any formatting changes you make by saving a Wave window format file (refer to Saving the Window Format in the User's Manual).

Objects

Table 131. Analog Sidebar Icons

Icon	Action	Description
	Open Wave Properties	Opens the Format tab of the Wave Properties dialog box, with the Analog format already selected. This dialog box duplicates the Wave Analog dialog box

Table 131. Analog Sidebar Icons (continued)

Icon	Action	Description
		displayed by choosing **Format > Format > ...Analog (custom)** fom the main menu.
	Toggle Row Height	Changes the height of the row that contains the analog waveform. Toggles the height between the Min and Max values (in pixels) you specified in the Open Wave Properties dialog box under Analog Display.
	Rescale to fit Y data	Changes the waveform height so that it fits top-to-bottom within the current height of the row.
	Show menu of other actions	Displays • View Min Y • View Max Y • Overlay Above • Overlay Below • Colorize All • Colorize Selected
	Drag to resize waveform height	Creates an up/down dragging arrow that you can use to temporarily change the height of the row containing the analog waveform.

Table 132. Window Icons

Icon shape	Example	Description
FSM button		opens the FSM Viewer window
Null		verilog/system verilog name event

Chapter 5
Keyboard Shortcuts and Mouse Actions

You can manipulate the user interface with various keyboard and mouse actions.

Window-Specific Keyboard Shortcuts

User-Defined Keyboard Shortcuts

Main and Source Window Mouse and Keyboard Shortcuts

List of Keyboard Shortcuts in GUI Windows

List Window Keyboard Shortcuts
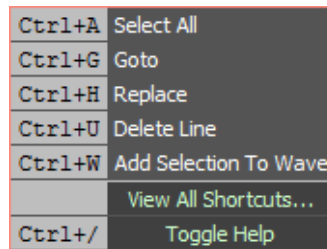
Wave Window Mouse and Keyboard Shortcuts

Window-Specific Keyboard Shortcuts

You can open a dynamic list of common (predefined) and user-defined keyboard sortcuts for many Questa SIM windows by entering "Ctrl+/" on your keyboard.

For example, Figure 169 shows the list of keyboard shortcuts provided for the Source window.

Figure 169. Keyboard Shortcuts for Source Window

You can find a complete list of all keyboard shortcuts—both predefined and user-defined—by clicking "View All Shortcuts" at the bottom of the list. Refer to User-Defined Keyboard Shortcuts for more information on how to create a customized shortcut.

User-Defined Keyboard Shortcuts

In addition to the predefined keyboard shortcuts, you can create your own shortcuts or modify predefined keyboard shortcuts with the Keyboard Shortcuts dialog box.

Shortcuts can be either window-specific (available only when the window is active) or global (available from anywhere in the tool). You can create a keyboard shortcut for any Questa SIM window.

Once a shortcut is defined, it will be available in all subsequent invocations. The dynamic nature of the architecture makes the keyboard shortcuts available to any Siemens EDA product that is based upon the Questa SIM GUI.

The Keyboard Shortcuts Dialog Box

Creating A Keyboard Shortcut

The Keyboard Shortcuts Dialog Box

The Keyboard Shortcuts dialog box lists all existing keyboard shortcuts. This dialog box distinguishes between shortcuts that are user-defined and shortcuts that come predefined with the Questa SIM simulator.

Figure 170 shows an example of the Keyboard Shortcuts dialog box, which you can display by choosing the following from the main menu:

Windows > Keyboard Shortcuts...

Figure 170. Keyboard Shortcuts Dialog Box
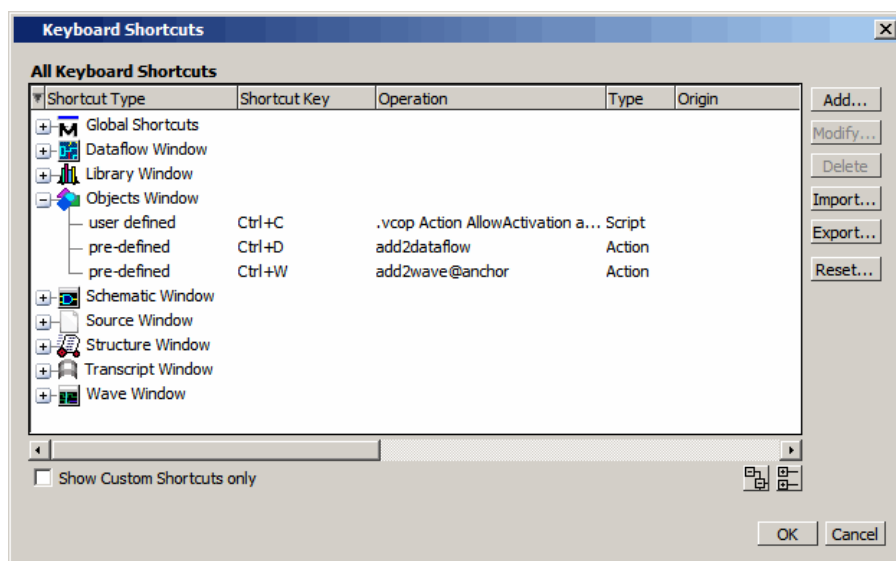
The Keyboard Shortcuts dialog box allows you to:

- Add a new user defined keyboard shortcut. Refer to Creating A Keyboard Shortcut for more information.

- Modify an existing keyboard shortcut. Any shortcut can be modified including predefined shortcuts.

- Delete a shortcut.

- Import shortcuts from a previously saved *bindings.do* file. You can also reload the keyboard shortcuts file with the do command.

- Export all user defined keyboard shortcuts to *bindings.do* file. Keyboard shortcuts saved in the file can be reloaded either by selecting the **Import button** in the Keyboard Shorcuts Dialog Box or by entering

    do bindings.do

  on the command line.

Creating A Keyboard Shortcut

You can create your own shortcut that is either global or that applies only to a specific window.

Procedure

1. If you are creating a window specific shortcut, the window must have been opened sometime during the simulation run.

2. Open the **Add Keyboard Shortcut** dialog box by selecting **Window > Keyboard Shortcuts**.

3. Click the **Add** button to open the Add Keyboard Shortcut dialog box.

Figure 171. Add Keyboard Shortcut Dialog Box
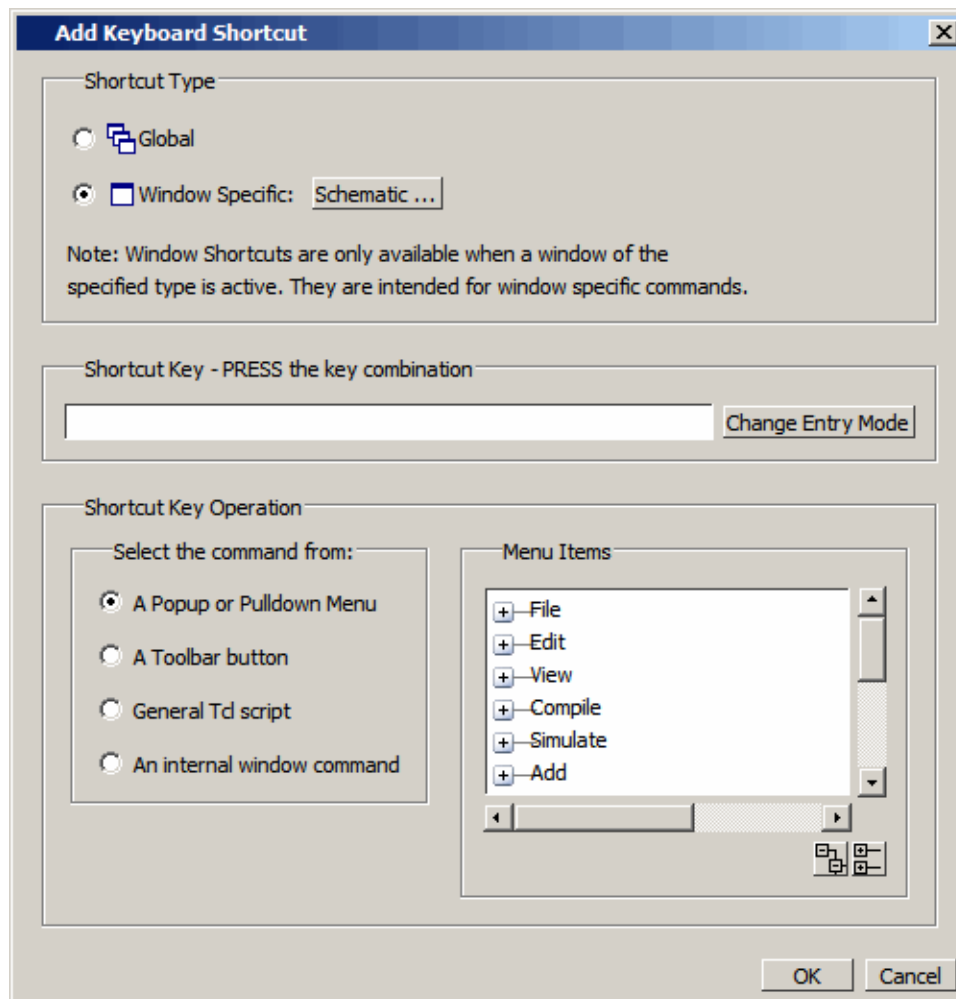
4. Select the Shortcut Type, either Global or Window. If you are creating a window specific shortcut, click the window button to open the **Select Window Type** dialog box. The dialog box displays every window that has been opened during the current simulation. If you do not see the window you are looking for, close both dialog boxes, open the window you want by entering **view** <window> on the command line, or by selecting the window from the **View** menu. Choosing Global or a specific window changes the options available in the **Shortcut Key Operation** field and the dynamically populated field to the right.

5. Enter the key combination in the **Shortcut Key** field. Or select the **Change Entry Mode** button to enter a key combination.

6. Choose the type of operation the shortcut will execute.

   • A Popup or Pulldown Menu — Opens the **Menu Items** dialog with a hierarchical list of all popup and pulldown menu items available either globally or for the window specified in step 4.

   • A Toolbar button — Opens the Toolbar Buttons dialog with a hierarchical list of all toolbar button actions available either globally or for the window specified in step 4.

- General Tcl script — Selecting this option opens the Tcl Script field to the right. You can enter any Tcl script or command line sequence.

- An Internal window command — This choice is available only for window specific commands. Refer to step 4. Opens the Window Action dialog on the right with a list of all window specific commands.

Main and Source Window Mouse and Keyboard Shortcuts

The following mouse actions and special keystrokes can be used to edit commands in the entry region of the Main window.

They can also be used in editing the file displayed in the Source window and all **Notepad** windows (enter the notepad command within Questa SIM to open the Notepad editor).

Table 133. Mouse Shortcuts

Mouse - UNIX and Windows	Result
Click the left mouse button	relocate the cursor
Click and drag the left mouse button	select an area
Shift-click the left mouse button	extend selection
Double-click the left mouse button	select a word
Double-click and drag the left mouse button	select a group of words
Ctrl-click the left mouse button	move insertion cursor without changing the selection
Click the left mouse button on a previous Questa SIM or VSIM prompt	copy and paste previous command string to current prompt
Click the middle mouse button	paste selection to the clipboard
Click and drag the middle mouse button	scroll the window

Table 134. Keyboard Shortcuts

Keystrokes - UNIX and Windows	Result
Left Arrow Right Arrow	move cursor left or right one character
Ctrl + Left Arrow Ctrl + Right Arrow	move cursor left or right one word
Shift + Any Arrow	extend text selection

Table 134. Keyboard Shortcuts (continued)

Keystrokes - UNIX and Windows	Result
Ctrl + Shift + Left Arrow Ctrl + Shift + Right Arrow	extend text selection by one word
Up Arrow Down Arrow	Transcript window: scroll through command history Source window: move cursor one line up or down
Ctrl + Up Arrow Ctrl + Down Arrow	Transcript window: moves cursor to first or last line Source window: moves cursor up or down one paragraph
Alt + /	Open a pop-up command prompt for entering commands.
Ctrl + Home	move cursor to the beginning of the text
Ctrl + End	move cursor to the end of the text
Backspace Ctrl + h (UNIX only)	delete character to the left
Delete Ctrl + d (UNIX only)	delete character to the right
Esc (Windows only)	cancel
Alt	activate or inactivate menu bar mode
Alt-F4	close active window
Home Ctrl + a	move cursor to the beginning of the line
Ctrl + Shift + a	select all contents of active window
Ctrl + b	move cursor left
Ctrl + d	delete character to the right
End Ctrl + e	move cursor to the end of the line
Ctrl + f (UNIX) Right Arrow (Windows)	move cursor right one character
Ctrl + k	delete to the end of line
Ctrl + n	move cursor one line down (Source window only under Windows)
Ctrl + o (UNIX only)	insert a new line character at the cursor

Table 134. Keyboard Shortcuts (continued)

Keystrokes - UNIX and Windows	Result
Ctrl + p	move cursor one line up (Source window only under Windows)
Ctrl + s (UNIX) Ctrl + f (Windows)	find
Ctrl + t	reverse the order of the two characters on either side of the cursor
Ctrl + u	delete line
Page Down Ctrl + v (UNIX only)	move cursor down one screen
Ctrl + x	cut the selection
Ctrl + s Ctrl + x (UNIX Only)	save
Ctrl + v	paste the selection
Ctrl + a (Windows Only)	select the entire contents of the widget
Ctrl + \	clear any selection in the widget
Ctrl + - (UNIX) Ctrl + / (UNIX) Ctrl + z (Windows)	undoes previous edits in the Source window
Meta + < (UNIX only)	move cursor to the beginning of the file
Meta + > (UNIX only)	move cursor to the end of the file
Page Up Meta + v (UNIX only)	move cursor up one screen
Ctrl + c	copy selection
F3	Performs a Find Next action in the Source window.
F4 Shift+F4	Change focus to next pane in main window Change focus to previous pane in main window
F5 Shift+F5	Toggle between expanding and restoring size of pane to fit the entire main window Toggle on/off the pane headers.

Table 134. Keyboard Shortcuts (continued)

Keystrokes - UNIX and Windows	Result
F8	search for the most recent command that matches the characters typed (Main window only)
F9	run simulation
F10	continue simulation
F11 (Windows only)	single-step
F12 (Windows only)	step-over

The Main window allows insertions or pastes only after the prompt; therefore, you do not need to set the cursor when copying strings to the command line.

List of Keyboard Shortcuts in GUI Windows

You can open a dynamic list of keyboard shortcuts, predetermined and user defined, for most windows by entering Ctrl-Shift-?

Figure 172. Schematic Window Keyboard Shortcuts
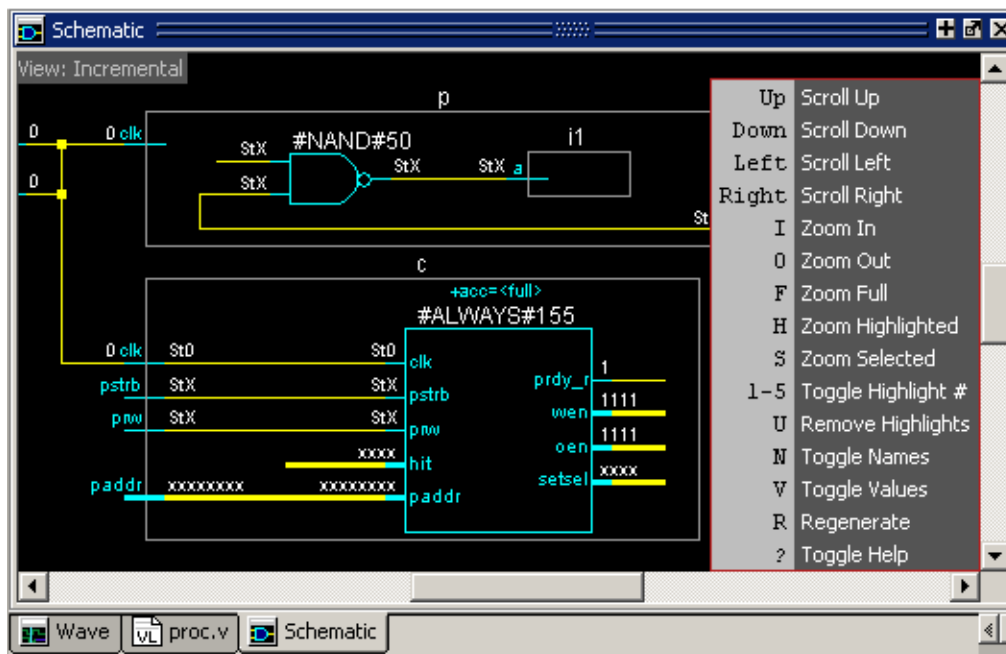
You can create user defined keyboard shortcuts and change predetermined shortcuts. Refer to User-Defined Keyboard Shortcuts for more information.

List Window Keyboard Shortcuts

Using the following keys when the mouse cursor is within the List window will cause the indicated actions:

Table 135. List Window Keyboard Shortcuts

Key - UNIX and Windows	Action
Left Arrow	scroll listing left (selects and highlights the item to the left of the currently selected item)
Right Arrow	scroll listing right (selects and highlights the item to the right of the currently selected item)
Up Arrow	scroll listing up
Down Arrow	scroll listing down
Page Up Ctrl + Up Arrow	scroll listing up by page
Page Down Ctrl + Down Arrow	scroll listing down by page
Tab	searches forward (down) to the next transition on the selected signal
Shift + Tab	searches backward (up) to the previous transition on the selected signal
Shift + Left Arrow Shift + Right Arrow	extends selection left/right
Ctrl + f (Windows) Ctrl + s (UNIX)	opens the Find dialog box to find the specified item label within the list display

Wave Window Mouse and Keyboard Shortcuts

The following mouse actions and keystrokes can be used in the Wave window.

Table 136. Wave Window Mouse Shortcuts

Mouse action[1]	Result
Ctrl + Click left mouse button and drag	zoom area (in)
Ctrl + Click left mouse button and drag	zoom out

Table 136. Wave Window Mouse Shortcuts (continued)

Mouse action[1]	Result
Ctrl + Click left mouse button and drag	zoom fit
Click left mouse button and drag	moves closest cursor
Ctrl + Click left mouse button on a scroll bar arrow	scrolls window to very top or bottom (vertical scroll) or far left or right (horizontal scroll)
Click middle mouse button in scroll bar (UNIX only)	scrolls window to position of click
Shift + scroll with middle mouse button	scrolls window

1. If you choose **Wave > Mouse Mode > Zoom Mode**, you do not need to press the Ctrl key.

Table 137. Wave Window Keyboard Shortcuts

Keystroke	Action
s	bring into view and center the currently active cursor
i Shift + i +	zoom in (mouse pointer must be over the cursor or waveform panes)
o Shift + o -	zoom out (mouse pointer must be over the cursor or waveform panes)
f Shift + f	zoom full (mouse pointer must be over the cursor or waveform panes)
l Shift + l	zoom last (mouse pointer must be over the cursor or waveform panes)
r Shift + r	zoom range (mouse pointer must be over the cursor or waveform panes)
m	zooms all open Wave windows to the zoom range of the active window.
Up Arrow Down Arrow	scrolls entire window up or down one line, when mouse pointer is over waveform pane scrolls highlight up or down one line, when mouse pointer is over pathname or values pane

Table 137. Wave Window Keyboard Shortcuts (continued)

Keystroke	Action
Left Arrow	scroll pathname, values, or waveform pane left
Right Arrow	scroll pathname, values, or waveform pane right
Page Up	scroll waveform pane up by a page
Page Down	scroll waveform pane down by a page
Tab	search forward (right) to the next transition on the selected signal - finds the next edge
Shift + Tab	search backward (left) to the previous transition on the selected signal - finds the previous edge
Ctrl+G	automatically create a group for the selected signals by region with the name Group\<n\>. If you use this shortcut on signals for which there is already a "Group\<n\>" they will be placed in that region's group rather than creating a new one.
Ctrl + F (Windows) Ctrl + S (UNIX)	open the find dialog box; searches within the specified field in the pathname pane for text strings
Ctrl + Left Arrow Ctrl + Right Arrow	scroll pathname, values, or waveform pane left or right by a page

Chapter 6
GUI Customization

The Questa SIM GUI is programmed using Tcl/Tk. The GUI is highly customizable—you can control a wide variety of display characteristics such as window size, position, color, the text of window prompts, and default output filenames.

Most user GUI preferences are stored as Tcl variables in the Registry on Windows platformsor the .modelsim file on Linux platforms. The variable values save automatically when you exit Questa SIM. Some of the variables are modified by actions you take with menus or windows (for example, resizing a window changes its geometry variable). Alternatively, you can edit the variables directly either from the prompt in the Transcript window or by using the **Tools > Edit > Preferences** menu item.

Simulator GUI Layout Customization
User-Defined Buttons and Menus
User-Defined Radices

Simulator GUI Layout Customization

GUI customization is saved in the form of layout modes.

Layout Modes

There are five predefined layout modes that the GUI will load dependent upon which part of the simulation flow you are currently in.

These modes include:

- NoDesign — This layout is the default view when you first open the GUI or quit out of an active simulation.

- Simulate — This layout appears after you have begun a simulation with vsim.

- Coverage — This layout appears after you have begun a simulation with the -coverage switch or loaded a UCDB dataset.

- VMgmt — This layout appears after you have loaded a dataset containing testplan information.

- VRM — This layout appears if you explicitly select it, for example from the **Layout > VRM** menu item, or by opening an *.rmdb* file from the **File > New > Regression** menu item. Note that once you set this layout, the GUI will not change to any other layout automatically; you will only be able to change the layout by selecting one from the Layout toolbar or the Layout menu.

These layout modes are fully customizable and the GUI stores your manipulations in the *.modelsim* file (Linux) or the registry (Windows) when you exit the simulation or change to another layout mode. The types of manipulations that are stored include: showing, hiding, moving, and resizing windows.

Layout Mode Loading Priority

The GUI stores your manipulations on a directory by directory basis and attempts to load a layout mode in the following order.

1. Directory — The GUI attempts to load any manipulations for the current layout mode based on your current working directory.

2. Last Used — If there is no layout related to your current working directory, the GUI attempts to load your last manipulations for that layout mode, regardless of your directory.

3. Default — If you have never manipulated a layout mode, or have deleted the **.modelsim** file or the registry, the GUI will load the default appearance of the layout mode.

Configure Window Layouts Dialog Box

The Configure Window Layouts dialog box allows you to alter the default behavior of the GUI layouts. You can display this dialog box by selecting the **Layout > Configure** menu item.

The elements of this dialog box include:

*   **Specify a Layout to Use** — This pane allows you to map which layout is used for the four actions. Refer to the section Changing Layout Mode Behavior for additional information.

*   **Save window layout automatically** — This option (on by default) instructs the GUI to save any manipulations to the layout mode upon exit or changing the layout mode.

*   **Save Window Layout by Current Directory** — This option (on by default) instructs the tool to save the final state of the GUI layout on a directory by directory basis. This means that the next time you open the GUI from a given directory, the tool will load your previous GUI settings.

*   **Window Restore Properties Button**— Opens the Window Restore Properties Dialog Box. Refer to Configuring Default Windows for Restored Layouts for more information.

Creating a Custom Layout Mode

You can create your own custom layout mode.

Procedure

1. Rearrange the GUI as you see fit.

2. Select **Layout > Save Layout As**.

    This displays the Save Current Window Layout dialog box.

3. In the Save Layout As field, type in a new name for the layout mode.

4. Click OK.

Results

The layout is saved to the *.modelsim* file or registry. You can then access this layout mode from the Layout menu or the Layout toolbar.

Changing Layout Mode Behavior

You can assign which predefined or custom layout appears in each mode.

Procedure

1. Create your custom layouts as described in Creating a Custom Layout Mode.

2. Select **Layout > Configure**.

   This displays the Configure Window Layouts Dialog Box.

3. Select which layout you want the GUI to load for each scenario. This behavior affects the Layout Mode Loading Priority.

4. Click **OK**.

Results

The layout assignment is saved to the *.modelsim* file or registry.

Resetting a Layout Mode to its Default

Revert a layout back to the default arrangement.

Procedure

1. Load the layout mode you want to reset via the Layout menu or the Layout toolbar.

2. Select **Layout > Reset**.

Deleting a Custom Layout Mode

You can delete one of your custom-made layout modes.

Procedure

1. Load a custom layout mode from the Layout menu or the Layout toolbar.

2. Select **Layout > Delete**.

   Displays the Delete Custom Layout dialog box.

3. Select the custom layout you wish to delete.

4. **Delete**.

Configuring Default Windows for Restored Layouts

The Window Restore Properties Dialog Box allows you to specify which windows will be restored when a layout is reloaded.

Procedure

1. Select **Layout > Configure** to open the **Configure Window Layouts** dialog box.

2. Click the **Window Restore Properties** button to open the **Window Restore Properties** dialog box

3. Select the windows you want to have opened when a new layout is loaded. Windows that are not selected will not load until specified with the view command or by selecting View > <window>.

4. You can also work with window layouts by specifying layout suppresstype <window>, layout restoretype, or layout showsuppresstypes. Refer to the layout command for more information.

Column Layout Configuration

Some windows allow you to configure the column layout using the Configure Column Layout dialog box.

Figure 173. Configure Column Layout Dialog
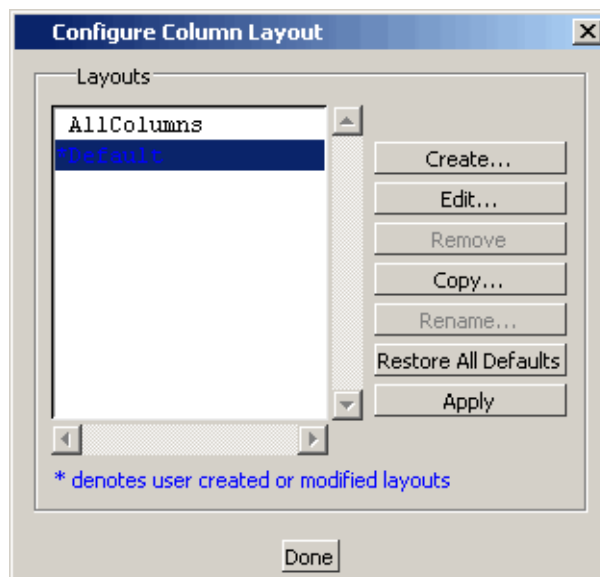
This dialog can also be opened by right-clicking a column heading and selecting Configure Column Layout from the popup menu; or by selecting Configure ColumnLayout from the drop-down list in the Column Layout Toolbar.

An asterisk (*) prefix and blue font indicate column layouts are in their default state and which have been added or modified.

Click the Edit button to open the Edit Column Layout dialog, where you can add and remove columns from the display and change their order (Figure 174).

Figure 174. Edit Column Layout Dialog
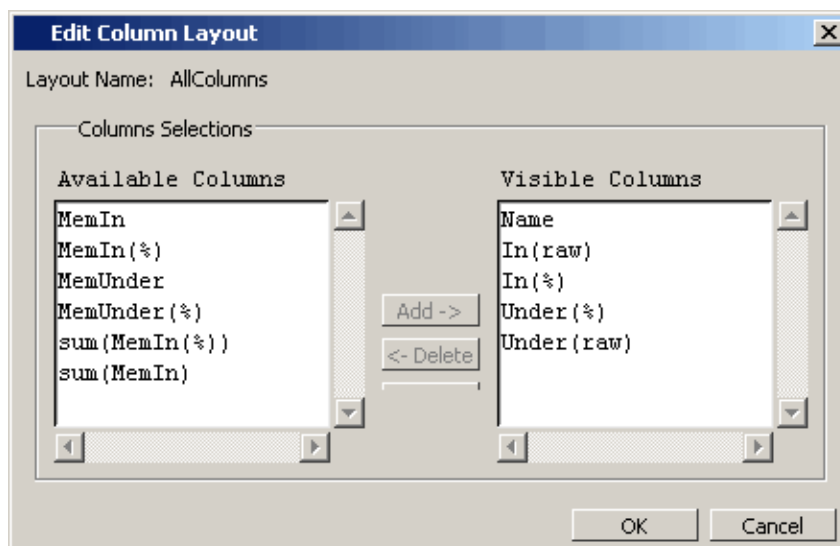
Or, click the Create button to create a customized column layout for your application. The Create Column Layout window allows you to select the columns that you want to appear in the customized layout. For example, in Figure 175, we have created a Memory Usage layout for the Ranked Profile window that includes only those columns related to memory usage.

Figure 175. Create Column Layout Dialog
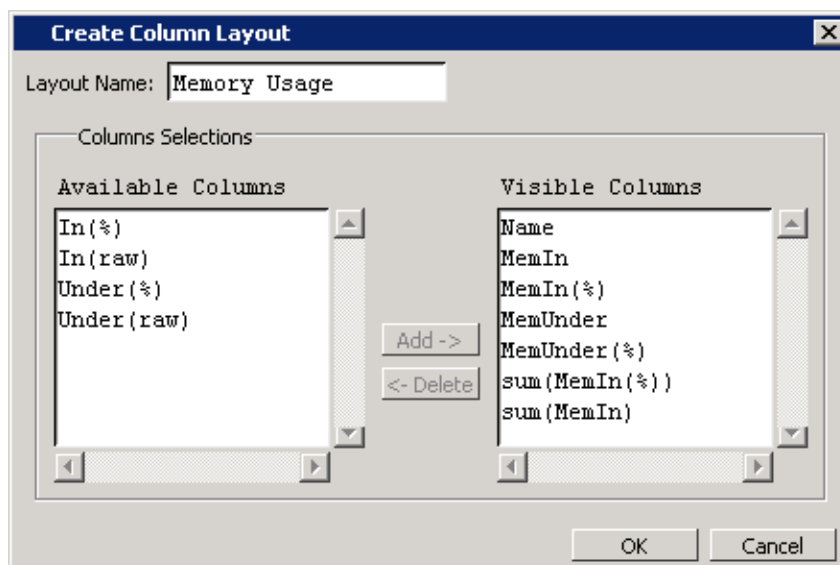
User-Defined Buttons and Menus

You can create Tcl processes (procs) that add user-defined buttons and menus to the main window of the Questa SIM GUI.

The Tcl procs are loaded after initialization by assigning them to the PrefMain(user_hook) preference variable. The procs must be saved in a *modelsim.tcl* file located in the install directory for Windows

platforms or in the directory from which you invoke the Questa SIM simulator for other platforms (refer to The modelsim.tcl File for more information). Buttons are added to the Standard toolbar and menus are added to the main menu bar in the GUI

The following Tcl example demonstrates two procs that create a menu and two buttons and the syntax for setting the procs with the PrefMain(user_hook) preference variable. Refer to (Figure 176).

Figure 176. User-Defined Buttons and Menus


```
proc AddMyMenus {wname} {
:     global myglobalvar
:     set cmd1 "echo my_own_thing $wname"
:     set cmd2 "echo my_to_upper $wname"
:     set cmd3 "echo my_to_lower $wname"
:
:#               WinName     Menu                 MenuItem label    Command
#               -------     ------------------   -----------------  -------------
---
add_menu       $wname      usrMenu
add_menuitem   $wname      usrMenu                               "Do My Own Thing"  $cmd1
add_separator  $wname      usrMenu                               ;#--------------  -------------
---
add_submenu    $wname      usrMenu               changeCase
add_menuitem   $wname      usrMenu.changeCase   "To Upper"         $cmd2
add_menuitem   $wname      usrMenu.changeCase   "To Lower"         $cmd3
add_submenu    $wname      usrMenu               vars
add_menucb     $wname      usrMenu.vars          "Feature One"      -variable      \
                myglobalvar    \
                -onvalue 1      \
                -offvalue 0     \
                -indicatoron 1 \
}
:
:proc my_buttons {args} {
:
:    add button "Run Long" "run 2 us"
:
    add button "Run Short" "run 2 ns"

    }


lappend PrefMain(user_hook) AddMyMenus my_buttons
```

The code above is available in the following *modelsim.tcl* file:

<install_dir>/examples/gui/addmenu/modelsim.tcl

- Menu proc

  Adds a menu to the Main menu bar containing a top-level item labeled "Do My Own Thing...",
  which prints "my_own_thing.signals." It adds a cascading submenu labeled "changeCase"
  with two entries, "To Upper" and "To Lower", which echo "my_to_upper" and "my_to_lower"
  respectively. The menu selection **UserMenu > Vars > Feature One** sets a checkbox that controls
  the value of myglobalvar (.signals:one).

- Button proc

  Adds two buttons to the Standard tool bar. "Run Long," runs the simulation for 2 us, "Run Short,"
  runs the simulation for 2 ns.

- The line:

  ```
  lappend PrefMain(user_hook) AddMyMenus my_buttons
  ```

  appends the two procs, AddMyMenus and my_buttons, to the user_hook variable when
  Questa SIM is finished initializing. Multiple procs are specified as a space separated list.

User-Defined Radices

A user definable radix is used to map bit patterns to a set of enumeration labels.

Using the radix define Command

Setting Global Signal Radix

Setting a Fixed Point Radix

Using the radix define Command

You can create or modify a radix with the radix define command.

The radix define command is used to create or modify a radix. It must include a radix name and a definition body, which consists of a list of number pattern, label pairs.

Optionally, it may include the -color argument for setting the radix color (see Example 2).

```
{
    <numeric-value>  <enum-label>,
    <numeric-value> <enum-label>
    -default <radix>
}
```

A <numeric-value> is any legitimate HDL integer numeric literal. To be more specific:

```
<base>#<base-integer># --- <base> is 2, 8, 10, or 16
<base>"bit-value"       --- <base> is B, O, or X
<integer>
<size>'<base><number>  --- <size> is an integer, <base> is b, d, o, or h.
```

Check the Verilog and VHDL LRMs for exact definitions of these numeric literals.

The comma (,) in the definition body is optional. The <enum-label> is any arbitrary string. It should be quoted (""), especially if it contains spaces.

The -default entry is optional. If present, it defines the radix to use if a match is not found for a given value. The -default entry can appear anywhere in the list, it does not have to be at the end.

Example 1 shows the radix define command used to create a radix called States, which will display state values in the List, Watch, and Wave windows instead of numeric values.

Example 1. Using the radix define Command

```
radix define States {
    11'b00000000001 "IDLE",
    11'b00000000010 "CTRL",
    11'b00000000100 "WT_WD_1",
    11'b00000001000 "WT_WD_2",
    11'b00000010000 "WT_BLK_1",
    11'b00000100000 "WT_BLK_2",
    11'b00001000000 "WT_BLK_3",
```

```
        11'b00010000000 "WT_BLK_4",
        11'b00100000000 "WT_BLK_5",
        11'b01000000000 "RD_WD_1",
        11'b10000000000 "RD_WD_2",
        -default hex
}
```

Figure 177 shows an FSM signal called */test-sm/sm_seq0/sm_0/state* in the Wave window with a binary radix and with the user-defined States radix (as defined in Example 1).

Figure 177. User-Defined Radix "States" in the Wave Window
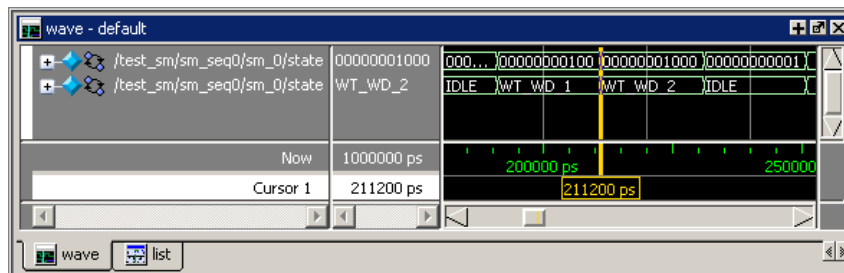
Figure 178 shows an FSM signal called */test-sm/sm_seq0/sm_0/state* in the List window with a binary radix and with the user-defined States radix (as defined in Example 1)

Figure 178. User-Defined Radix "States" in the List Window
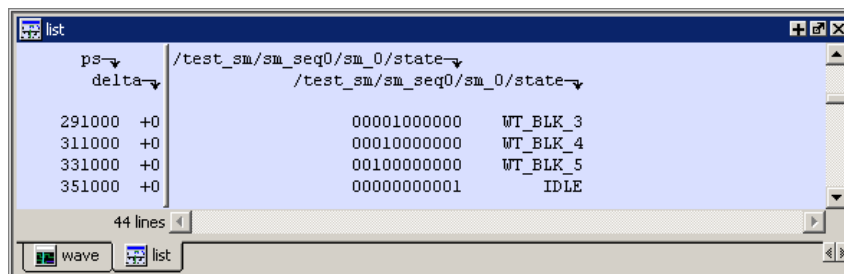
Using radix define to Specify Radix Color

The following example illustrates how to use the radix define command to specify the radix color:

Example 2. Using radix define to Specify Color

```
radix define States {
  11'b00000000001 "IDLE" -color yellow,
  11'b00000000010 "CTRL" -color #ffee00,
  11'b00000000100 "WT_WD_1" -color orange,
  11'b00000001000 "WT_WD_2" -color orange,
  11'b00000010000 "WT_BLK_1",
  11'b00000100000 "WT_BLK_2",
  11'b00001000000 "WT_BLK_3",
  11'b00010000000 "WT_BLK_4",
  11'b00100000000 "WT_BLK_5",
  11'b01000000000 "RD_WD_1" -color green,
```

```
  11'b10000000000 "RD_WD_2" -color green,
  -default hex
  -defaultcolor white
}
```

If a pattern/label pair does not specify a color, the normal wave window colors will be used. If the value of the waveform does not match any pattern, then -default <radix_type> and -defaultcolor will be used.

To specify a range of values, wildcards may be specified for bits or characters of the value. The wildcard character is '?', similar to the iteration character in a Verilog UDP, for example:

```
radix define {
 6'b01??00 "Write" -color orange,
 6'b10??00 "Read" -color green
}
```

In this example, the first pattern will match "010000", "010100", "011000", and "011100". In case of overlaps, the first matching pattern is used, going from top to bottom.

Setting Global Signal Radix

The Global Signal Radix feature allows you to set the radix for a selected signal or signals in the active window and in other windows where the signal appears.

Procedure

1. The Global Signal Radix can be set from the Locals, Objects, Schematic, or Wave windows as follows:

2. Select a signal or group of signals.

3. Right-click the selected signal(s) and click the following popup menu option:

   - Objects Window: **Radix**

   - Locals Window: **Global Signal Radix**

   - Wave Window: **Radix > Global Signal Radix**

   - Schematic Window: **Edit > Global Signal Radix**

   This opens the Global Signal Radix dialog box (), where you may select a radix. This sets the radix for the selected signal(s) in the active window and every other window where the signal appears.

   > **Note:**
   > When you select two or more signals in step 1 of this procedure the **Fixed/Float** option in the Global Signal Radix dialog box is not available.

Figure 179. Setting the Global Signal Radix
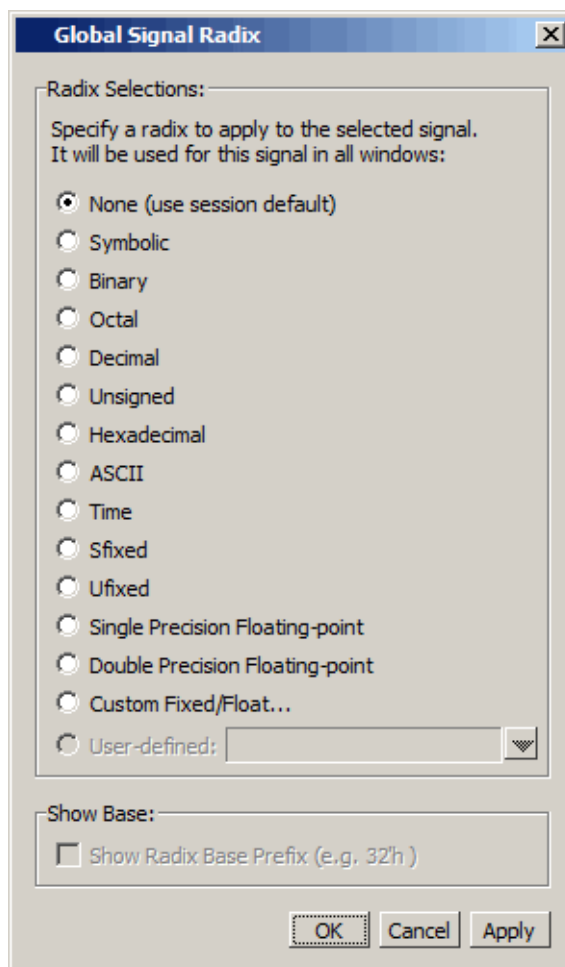
Sfixed and Ufixed indicate "signed fixed" and "unsigned fixed," respectively. To display an object as Sfixed or Ufixed the object must be an array of std_ulogic elements between 2 and 64 bits long with a descending range. The binary point for the value is implicitly located between the 0th and -1st elements of the array. The index range for the type need not include 0 or -1, for example (-4 downto -8) in which case the value will be extended for conversion, as appropriate. If the type does not meet these criteria the value will be displayed as decimal or unsigned, respectively.

Setting a Fixed Point Radix

Fixed point types are used in VHDL and SystemC to represents non-integer numbers without using a floating point format. Questa SIM automatically recognizes VHDL sfixed and ufixeda types as well as SystemC SC_FIXED and SC_UFIXED types and displays them correctly with a fixed point format.

In addition, a general purpose fixed point radix feature is available for displaying any vector, regardless of type, in a fixed point format in the Wave window. You simply have to specify how many bits to use as fraction bits from the whole vector.

With the Wave window active:

1. Select (LMB) a signal or signals in the Pathnames pane of the Wave window.

2. Right-click the selected signal(s) and select **Radix > Global Signal Radix**from the popup menu. This opens the Global Signal Radix dialog box shown in Figure 179.

3. Click the **Custom Fixed/Float** selection to open the Fixed Point Radix dialog box (Figure 180).

Figure 180. Fixed Point Radix Dialog Box
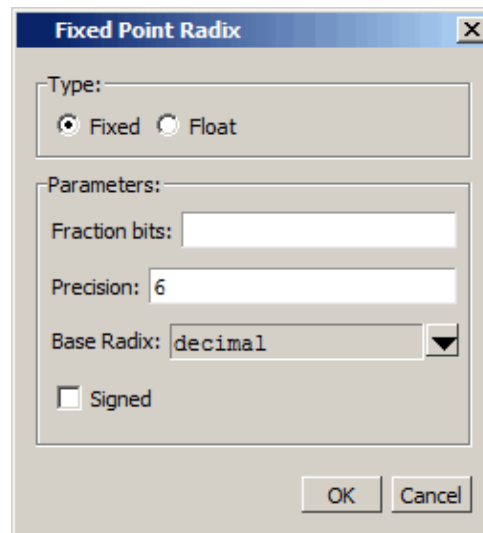
The Fixed Point Radix dialog box allows you to select either a fixed or floating point radix type, and to set the Fraction bits, Precision, and Base Radix parameters. The default Base Radix is decimal. You can also check the Signed box if you want the radix to be signed.

Chapter 7
GUI Preferences

You can control various aspects of the Graphical User Interface through the use of user-controlled preferences.

Setting GUI Preferences

You can set user-controlled preferences in the Questa SIM GUI with the Preferences dialog box.

Procedure

1. Invoke the Questa SIM GUI.

2. Select the **Tools > Edit Preferences** menu item to display the Preferences Dialog Box.

3. Alter the various preference options to customize your GUI.

    By Window tab:

    a. Make selections, from left to right, to change GUI element colors or global fonts.

    By Name tab:

    a. Hierarchically expand the specific category in the Preference Item column.

    b. Select the specific Preference Item you want to change.

    c. Click **Change Value** to display a dialog box specific to the item.

    d. Set the preference to your desired value.

    e. Click **OK**.

Results

The preferences are applied immediately and saved automatically upon exit in the registry of your Windows machineor the .modelsim file in your Linux home directory.

Preferences Dialog Box

Tools > Edit Preferences

Dialog box for controlling user-defined preferences of the Questa SIM GUI.

Figure 181. Preferences Dialog Box
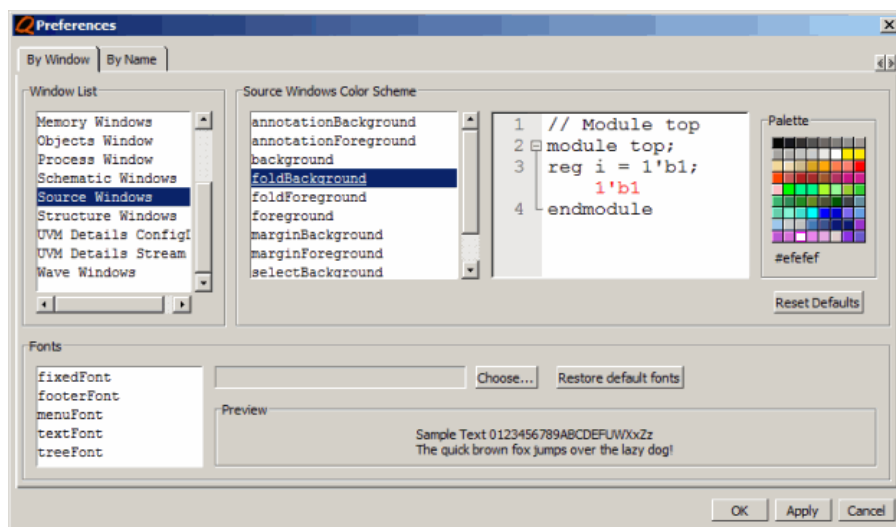
Objects

• Preferences Dialog Box

- ◦ By Window tab — Allows you to control the color of various parts of many GUI windows, as well as the global appearance of fonts.

  - ◦ Fonts — The specific font types are defined as:

    - • fixedFont — Text in Source window and in all text entry fields or boxes.

    - • footerFont — Footer text that appears in footer of Main window and all undocked windows.

    - • menuFont — Menu text.

    - • textFont — Transcript window text and text in list boxes.

    - • treeFont — Text that appears in any window that displays a hierarchical tree.

- ◦ By Name tab — Allows you to control many aspects of the different windows and features of the GUI.

Usage Notes

- In the By Window tab, the Palette box shows the default color below the color chooser.

- In the By Name tab, the Description column

Saving GUI Preferences in an Alternate Location

Using the MODELSIM_PREFERENCES environment variable in a Linux envrionment allows you to save GUI preferences to an alternate location.

You can use the MODELSIM_PREFERENCES environment Variable (refer to MODELSIM_PREFERENCESin the User's Manual) to designate a path and file for the preferences in place of the default location. Here are some additional points to keep in mind about this variable setting:

- The file does not need to exist before setting the variable as Questa SIM will initialize it.

- If the file is read-only, Questa SIM will not update or otherwise modify the file.

- This variable may contain a relative pathname, in which case the file is relative to the working directory at the time the tool is started.

The modelsim.tcl File

Older releases of the Questa SIM simulator saved user GUI preferences into a file named *modelsim.tcl*. Current releases will still search for and read a *modelsim.tcl* file, if it exists. The search order for this file is the following:

1. Use the MODELSIM_TCL environment variable (refer to MODELSIM_TCL in the User's Manual). If MODELSIM_TCL is a list of files, each file is loaded in the order that it appears in the list; else

2. Use *./modelsim.tcl*; else

3. Use $(HOME)/modelsim.tcl if it exists.

Note that in Release 6.1 and later, Questa SIM will save to the *.modelsim* file any variables it reads in from a *modelsim.tcl* file (except for user_hook variables). The values from the *modelsim.tcl* file will override like variables in the *.modelsim* file.

User_hook Variables

You can create Tcl processes (procs) that add persistent user-defined buttons and menus to the main window of Questa SIM. The Tcl procs are loaded after initialization by assigning them to the PrefMain(user_hook) preference variable.

The procs must be saved in a *modelsim.tcl* file located in the install directory for Windows platforms or in the directory from which Questa SIM is invoked for other platforms (refer to The modelsim.tcl File for more information). Buttons are added to the Standard Toolbar tab and menus are added to the main menu bar in the GUI.

The following Tcl example demonstrates two procs that create a menu and two buttons and the syntax for setting the procs with the PrefMain(user_hook) preference variable. Refer to (Figure 182).

Figure 182. Persistent Buttons and Menus


```
proc AddMyMenus {wname} {
:      global myglobalvar
:      set cmd1 "echo my_own_thing $wname"
:      set cmd2 "echo my_to_upper $wname"
:      set cmd3 "echo my_to_lower $wname"
:
:#             WinName    Menu                 MenuItem label        Command
#              -------    -----------------    --------------------  ----------
---
add_menu       $wname     usrMenu
add_menuitem   $wname     usrMenu              "Do My Own Thing..."  $cmd1
add_separator  $wname     usrMenu              ;#------------------  ----------
---
add_submenu    $wname     usrMenu              changeCase
add_menuitem   $wname     usrMenu.changeCase   "To Upper"            $cmd2
add_menuitem   $wname     usrMenu.changeCase   "To Lower"            $cmd3
add_submenu    $wname     usrMenu              vars
add_menucb     $wname      usrMenu.vars        "Feature One"         -variable
                     myglobalvar
                  -onvalue 1
                  -offvalue 0
                  -indicatoron 1
}

:proc my_buttons {args} {
:
:   add button "Run Long" "run 2 us"
:
:   add button "Run Short" "run 2 ns"

   }
lappend PrefMain(user_hook) AddMyMenus my_buttons
```

- Menu proc

  Adds a menu to the Main menu bar containing a top-level item labeled "Do My Own Thing...", which prints "my_own_thing.signals." It adds a cascading submenu labeled "changeCase" with two entries, "To Upper" and "To Lower", which echo "my_to_upper" and "my_to_lower" respectively. The menu selection **UserMenu > Vars > Feature One** sets a checkbox that controls the value of myglobalvar (.signals:one).

- Button proc

  Adds two buttons to the Standard tool bar. The first, "Run Long," runs the simulation for 2 us, the second, "Run Short," runs the simulation for 2 ns.

- The line:

```
lappend PrefMain(user_hook) AddMyMenus my_buttons
```

  appends the two procs AddMyMenus and my_buttons to the user_hook variable when Questa SIM is finished initializing. Multiple procs are specified as a space separated list.

This example is available in the following *modelsim.tcl* file:

<install_dir>/examples/gui/addmenu/modelsim.tcl

Wave Window Variables

The LogicStyleTable combined with the ListTranslateTable define how single bit waveforms are displayed in the Wave window.

The single value is first mapped into one of nine (9) possible states: U, 0, 1, X, Z, W, H, L, or '-' (Don't Care). Then the entry for the corresponding value in the LogicStyleTable is used to determine what is drawn in the Wave window. The line style is either Solid or DoubleDash. The line is drawn in the color specified. Lastly, the line is drawn at the top of the row (2), the middle of the row (1), or the bottom of the row (0).

The mapping of bit values to the 9 states is specified in the ListTranslateTable. The table is searched to find a matching value. When a match is found, the corresponding table entry defines the 9 state value used to define the style.

**Table 138. Default ListTranslateTable Values**

Mapped State	Bit Value
LOGIC_U	'U'
LOGIC_X	'X' 'x'
LOGIC_0	'0' FALSE
LOGIC_1	'1' TRUE
LOGIC_Z	'Z' 'z'
LOGIC_W	'W'
LOGIC_L	'L'
LOGIC_H	'H'
LOGIC_DC	'-'

Table 139. Default LogicStyleTable Values

Mapped State	Line Style	Color	Row Position
LOGIC_U	Solid	red	1
LOGIC_X	Solid	red	1
LOGIC_0	Solid	green	0
LOGIC_1	Solid	green	2
LOGIC_Z	Solid	blue	1
LOGIC_W	DoubleDash	red	1
LOGIC_L	DoubleDash	grey90	0
LOGIC_H	DoubleDash	grey90	2
LOGIC_DC	DoubleDash	blue	1

Figure 183. Modifying Signal Display Attributes in the Wave Window
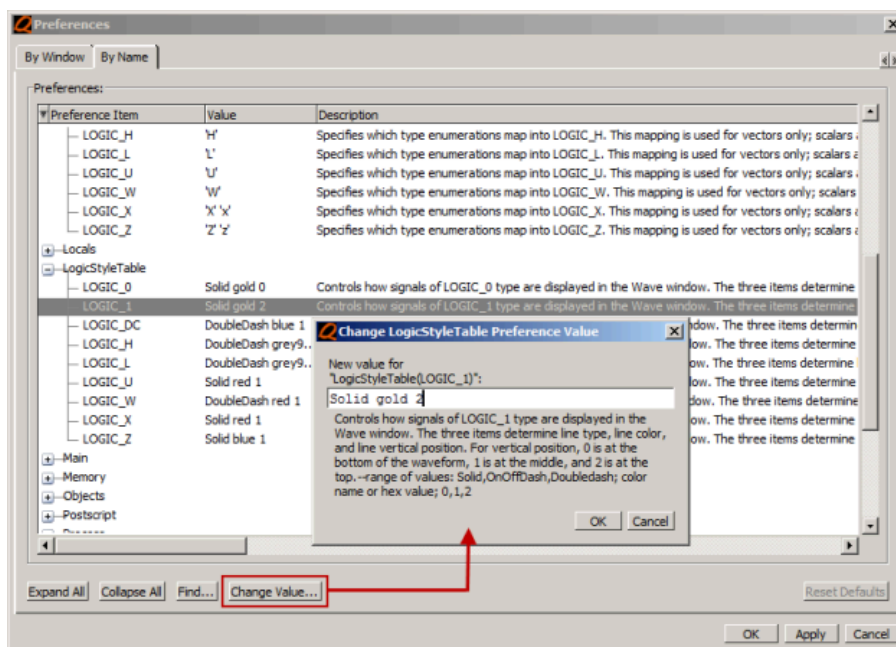
Modifying Wave Window Variables from the Command Line

Both the LogicStyleTable and the ListTranslateTable variables can be modified using the Tcl set command. Changes are automatically saved in the preference file ($HOME/.modelsim on UNIX/Linux platforms or the Registry on Windows platforms).

The Tcl set command equivalent of the ListTranslateTable is:

```
set ListTranslateTable(LOGIC_U)  {'U'}
set ListTranslateTable(LOGIC_X)  {'X' 'x'}
set ListTranslateTable(LOGIC_0)  {'0' FALSE}
set ListTranslateTable(LOGIC_1)  {'1' TRUE}
set ListTranslateTable(LOGIC_Z)  {'Z' 'z'}
set ListTranslateTable(LOGIC_W)  {'W'}
set ListTranslateTable(LOGIC_L)  {'L'}
set ListTranslateTable(LOGIC_H)  {'H'}
set ListTranslateTable(LOGIC_DC) {'-'}
```

The Tcl set command equivalent of the LogicStyleTable is:

```
set LogicStyleTable(LOGIC_U)  {Solid      red    1}
set LogicStyleTable(LOGIC_X)  {Solid      red    1}
set LogicStyleTable(LOGIC_0)  {Solid      green  0}
set LogicStyleTable(LOGIC_1)  {Solid      green  2}
set LogicStyleTable(LOGIC_Z)  {Solid      blue   1}
set LogicStyleTable(LOGIC_W)  {DoubleDash red    1}
set LogicStyleTable(LOGIC_L)  {DoubleDash grey90 0}
set LogicStyleTable(LOGIC_H)  {DoubleDash grey90 2}
set LogicStyleTable(LOGIC_DC) {DoubleDash blue   1}
```