# Table of Contents

## Chapter 1
### Overview
- General GUI Features
- Window Management
- Window Arrangement
- Column-Based Windows
- Bookmarks
- Font Management
- Find and Filter Functions
- General Visual Elements
- Elements of the Main Window
- Design Object Icons and Their Meanings
- Window Time Display

## Chapter 2
### Menus
- 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

## Chapter 3
### Toolbars
- Bookmarks Toolbar
- Compile Toolbar
- Dataflow Toolbar
- Help Toolbar
- Layout Toolbar
- Memory Toolbar
- Mode Toolbar
- Objectfilter Toolbar
- Process Toolbar
- Schematic Toolbar
- Simulate Toolbar
## Table of Contents

Source Toolbar .......................................................... 55  
Standard Toolbar ...................................................... 56  
Step Toolbar ............................................................ 58  
Wave Toolbar ........................................................... 59  
Wave Cursor Toolbar .................................................. 61  
Wave Edit Toolbar ...................................................... 62  
Wave Expand Time Toolbar ........................................... 63  
Zoom Toolbar ............................................................ 64  
Toolbar Tabs ............................................................... 65  
  Compile Toolbar Tab ................................................ 66  
  Debug Toolbar Tab .................................................. 67  
  Edit Toolbar Tab .................................................... 71  
  Home Toolbar Tab .................................................. 73  
  Layout Toolbar Tab ................................................. 75  
  Schematic and Dataflow Toolbar Tab ......................... 76  
  Simulate Toolbar Tab .............................................. 78  
Windows With Dedicated Toolbars ............................... 79  
  Toolbar Visibility and Layout .................................... 80  
Creating and Restoring Toolbar Tabs ......................... 81  
Customizing Button and Tab Location ......................... 81  
Tabbed Toolbar Mapping to Default Toolbars ............... 83

## Chapter 4

### Window Reference ................................................. 85

- Call Stack Window .................................................. 88  
- Class Graph Window ................................................ 91  
- Class Instances Window ......................................... 93  
- Class Tree Window ................................................ 96  
- Dataflow Window ................................................... 99  
- Dataflow Window Tasks .......................................... 100  
  - Selecting Objects in the Dataflow Window ................ 101  
  - Zooming the View of the Dataflow Window ............... 101  
  - Panning the View of the Dataflow Window ............... 102  
  - Displaying the Wave Viewer Pane ......................... 103  
- Files Window ........................................................ 106  
- Library Window .................................................... 109  
- List Window .......................................................... 111  
- List Window Tasks ................................................ 114  
  - Adding Data to the List Window ............................ 114  
  - Selecting Multiple Signals .................................. 115  
  - Setting Time Markers in the List Window ............... 115  
  - Searching in the List Window ............................... 118  
  - Searching for Values or Transitions ..................... 119  
  - Using the Expression Builder for Expression Searches . 120  
  - Saving an Expression to a Tcl Variable ................. 122  
  - Formatting the List Window ................................. 123  
  - Saving the Window Format ................................... 125  
  - Combining Signals into Buses ............................... 126
# Table of Contents

- Configuring New Line Triggering .............................................. 127
- Locals Window ................................................................. 132
- Memory Data Window ......................................................... 136
- Memory List Window ........................................................ 139
- Saving Memory Formats in a DO File. .................................... 143
- Saving Memories to the WLF File. ........................................ 144
- Message Viewer Window .................................................... 146
- Objects Window .................................................................. 154
- Objects Window Tasks .......................................................... 156
  - Interacting with Other Windows ......................................... 157
  - Setting Signal Radix .......................................................... 157
  - Finding Contents of the Objects Window ............................... 158
  - Filtering Contents of the Objects Window ............................... 158
  - Filtering by Signal Type .................................................... 159
- Processes Window ............................................................... 161
- Source Window .................................................................. 164
- Using the Source Window ...................................................... 166
  - Dragging and Dropping Objects into the Wave and List Windows 166
  - Setting your Context by Navigating Source Files .................. 167
  - Setting File-Line Breakpoints with the GUI ......................... 169
  - Adding File-Line Breakpoints with the bp Command ............. 170
  - Editing File-Line Breakpoints ........................................... 170
  - Setting Conditional Breakpoints ....................................... 172
  - Checking Object Values and Descriptions ............................ 174
  - Marking Lines with Bookmarks ......................................... 175
  - Performing Incremental Search for Specific Code ................ 175
  - Customizing the Source Window ....................................... 176
- Structure Window ............................................................... 177
- Structure Window Tasks ........................................................ 180
  - Display Source Code of a Structure Window Object ................ 180
  - Add Structure Window Objects to Other Windows ................. 181
  - Finding Items in the Structure Window ................................. 181
  - Filtering Structure Window Objects .................................... 182
- Transcript Window .............................................................. 184
- Transcript Window Tasks ..................................................... 185
  - Saving the Transcript File ................................................ 185
  - Colorizing the Transcript ................................................ 186
  - Disabling Creation of the Transcript File ............................. 187
  - Performing an Incremental Search ...................................... 187
  - Using Automatic Command Help ....................................... 188
  - Using drivers and Readers Command Results ....................... 188
- Watch Window ................................................................. 189
- Wave Window ................................................................. 194

## Chapter 5

### Keyboard Shortcuts and Mouse Actions ........................................... 203

- Window-Specific Keyboard Shortcuts ....................................... 203
- User-Defined Keyboard Shortcuts .......................................... 204
## Table of Contents

The Keyboard Shortcuts Dialog Box .................................................. 204
Creating A Keyboard Shortcut ......................................................... 205
Main and Source Window Mouse and Keyboard Shortcuts ................. 207
List of Keyboard Shortcuts in GUI Windows ................................... 210
List Window Keyboard Shortcuts ..................................................... 211
Wave Window Mouse and Keyboard Shortcuts ............................... 211

### Chapter 6

**GUI Customization** ................................................................. 215

- Simulator GUI Layout Customization ....................................... 215
  - Layout Modes ........................................................................ 215
  - Layout Mode Loading Priority ............................................. 215
- Configure Window Layouts Dialog Box .................................... 216
- Creating a Custom Layout Mode ............................................... 216
- Changing Layout Mode Behavior .............................................. 217
- Resetting a Layout Mode to its Default .................................... 217
- Deleting a Custom Layout Mode ............................................. 217
- Configuring Default Windows for Restored Layouts .................. 218
- User-Defined Buttons and Menus .......................................... 218
- User-Defined Radices ............................................................. 219
  - Using the radix define Command ....................................... 220
  - Setting Global Signal Radix ............................................... 222
  - Setting a Fixed Point Radix ............................................... 224

### Chapter 7

**GUI Preferences** ................................................................. 225

- Setting GUI Preferences .......................................................... 225
  - Preferences Dialog Box ....................................................... 225
  - Wave Window Variables ..................................................... 227

## Index

End-User License Agreement
List of Examples

Example 6-1. Using the radix define Command .................................................. 220
Example 6-2. Using radix define to Specify Color .............................................. 221
List of Figures

Figure 1-1. Graphical User Interface .................................................. 15
Figure 1-2. Window Header Handle .................................................. 17
Figure 1-3. Tab Handle ................................................................. 17
Figure 1-4. Window Undock Button .................................................. 18
Figure 1-5. Manage Bookmarks Dialog Box .......................................... 20
Figure 1-6. Bookmark Options Dialog Box ........................................... 21
Figure 1-7. Find Mode ................................................................. 22
Figure 1-8. Filter Mode ............................................................... 22
Figure 1-9. Find Mode Popup Displaying Matches ................................... 25
Figure 1-10. Find Options Popup Menu ............................................. 26
Figure 1-11. Main Window of the GUI ................................................. 27
Figure 1-12. Main Window — Menu Bar .............................................. 28
Figure 1-13. Main Window — Toolbar Frame ....................................... 28
Figure 1-14. Main Window — Toolbar .............................................. 29
Figure 1-15. GUI Windows ............................................................ 30
Figure 1-16. GUI Tab Group ........................................................... 31
Figure 1-17. Wave Window Panes .................................................... 32
Figure 1-18. Main Window Status Bar ............................................... 32
Figure 1-19. Current Time Label ..................................................... 35
Figure 1-20. Enter Current Time Value ............................................. 35
Figure 3-1. Bookmarks Toolbar ....................................................... 47
Figure 3-2. Compile Toolbar .......................................................... 48
Figure 3-3. Dataflow Toolbar ......................................................... 49
Figure 3-4. Help Toolbar ............................................................... 49
Figure 3-5. Layout Toolbar ............................................................. 50
Figure 3-6. Memory Toolbar .......................................................... 50
Figure 3-7. Mode Toolbar .............................................................. 51
Figure 3-8. Objectfilter Toolbar ....................................................... 52
Figure 3-9. Process Toolbar ........................................................... 52
Figure 3-10. Schematic Toolbar ....................................................... 53
Figure 3-11. Simulate Toolbar ........................................................ 54
Figure 3-12. Source Toolbar .......................................................... 56
Figure 3-13. Standard Toolbar ......................................................... 56
Figure 3-14. The Add Selected to Window Dropdown Menu ..................... 58
Figure 3-15. Step Toolbar .............................................................. 59
Figure 3-16. Wave Toolbar .............................................................. 59
Figure 3-17. Wave Cursor Toolbar .................................................... 61
Figure 3-18. Wave Edit Toolbar ....................................................... 62
Figure 3-19. Wave Expand Time Toolbar .......................................... 63
Figure 3-20. Zoom Toolbar ............................................................ 64
List of Figures

Figure 3-21. Toolbar Tabs and Overflow Menu ................................................. 66
Figure 3-22. Compile Tab .............................................................................. 66
Figure 3-23. Debug Tab ................................................................................. 67
Figure 3-24. Edit Tab ..................................................................................... 72
Figure 3-25. Home Tab ............................................................................... 73
Figure 3-26. Layout Tab ................................................................................. 76
Figure 3-27. Schematic Tab ................................................................. 76
Figure 3-28. Simulate Tab ............................................................................. 78
Figure 3-29. Window Specific Buttons ................................................................. 80
Figure 3-30. Toolbar Widget Toolbox ................................................................. 82
Figure 4-1. Call Stack Window ........................................................................... 88
Figure 4-2. Class Graph Window ..................................................................... 91
Figure 4-3. Class Instances Window ................................................................. 94
Figure 4-4. Class Tree Window ....................................................................... 96
Figure 4-5. Dataflow Window - ModelSim .......................................................... 100
Figure 4-6. Dataflow Window and Panes ........................................................... 104
Figure 4-7. Files Window .............................................................................. 106
Figure 4-8. Library Window ......................................................................... 109
Figure 4-9. Tabular Format of the List Window .................................................. 111
Figure 4-10. List Window .............................................................................. 112
Figure 4-11. Time Markers in the List Window ................................................... 116
Figure 4-12. List Window After configure list -delta none Option is Used ................. 117
Figure 4-13. List Window After configure list -delta collapse Option is Used ............. 117
Figure 4-14. List Window After write list -delta all Option is Used ....................... 118
Figure 4-15. List Window After write list -event Option is Used ......................... 118
Figure 4-16. Wave Signal Search Dialog Box ..................................................... 120
Figure 4-17. Expression Builder Dialog Box ...................................................... 121
Figure 4-18. Selecting Signals for Expression Builder .......................................... 122
Figure 4-19. Modifying List Window Display Properties ....................................... 123
Figure 4-20. List Signal Properties Dialog ....................................................... 124
Figure 4-21. Changing the Radix in the List Window ........................................... 125
Figure 4-22. Line Triggering in the List Window ................................................. 127
Figure 4-23. Setting Trigger Properties ............................................................ 128
Figure 4-24. Trigger Gating Using Expression Builder .......................................... 130
Figure 4-25. Select Signal for Expression Dialog Box ........................................... 130
Figure 4-26. Locals Window .......................................................................... 132
Figure 4-27. Change Selected Variable Dialog Box ............................................. 132
Figure 4-28. Memory Data Window .................................................................. 136
Figure 4-29. Split Screen View of Memory Contents ........................................... 138
Figure 4-30. Memory List Window .................................................................. 141
Figure 4-31. Message Viewer Window .............................................................. 147
Figure 4-32. Message Viewer Window — Tasks ................................................ 148
Figure 4-33. Configuration Options for Message Viewer ....................................... 151
Figure 4-34. Edit Filter Expression Dialog Box .................................................... 152
Figure 4-35. Objects Window ......................................................................... 154
Figure 4-36. Object Window Toolbar .................................................. 155
Figure 4-37. Setting the Global Signal Radix from the Objects Window .................................................. 158
Figure 4-38. Processes Window .......................................................... 161
Figure 4-39. Source Window .............................................................. 164
Figure 4-40. Displaying Multiple Source Files ..................................... 166
Figure 4-41. Setting Context from Source Files .................................... 167
Figure 4-42. Breakpoint in the Source Window ..................................... 170
Figure 4-43. Modifying Existing Breakpoints ....................................... 171
Figure 4-44. Source Code for source.sv ............................................... 173
Figure 4-45. Source Window Description ............................................. 175
Figure 4-46. Source Window with Find Toolbar .................................... 176
Figure 4-47. Structure Window ......................................................... 178
Figure 4-48. Find Mode Popup Displays Matches ................................... 181
Figure 4-49. Changing the colorizeTranscript Preference Value ................ 187
Figure 4-50. Transcript Window with Find Toolbar ............................... 188
Figure 4-51. drivers Command Results in Transcript .............................. 188
Figure 4-52. Watch Window .............................................................. 189
Figure 4-53. Scrollable Hierarchical Display ........................................ 190
Figure 4-54. Expanded Array ............................................................. 193
Figure 4-55. Wave Window ............................................................... 194
Figure 4-56. Pathnames Pane ............................................................ 195
Figure 4-57. Setting the Global Signal Radix from the Wave Window ........ 196
Figure 4-58. Values Pane ................................................................. 197
Figure 4-59. Waveform Pane ............................................................ 197
Figure 4-60. Analog Sidebar Toolbox .................................................. 198
Figure 4-61. Cursor Pane ................................................................. 198
Figure 4-62. Wave Window - Message Bar .......................................... 198
Figure 4-63. View Objects Window Dropdown Menu ............................. 199
Figure 5-1. Keyboard Shortcuts for Source Window .............................. 203
Figure 5-2. Keyboard Shortcuts Dialog Box ......................................... 204
Figure 5-3. Add Keyboard Shortcut Dialog Box ..................................... 206
Figure 5-4. Schematic Window Keyboard Shortcuts ............................... 210
Figure 6-1. User-Defined Buttons and Menus ...................................... 218
Figure 6-2. User-Defined Radix “States” in the Wave Window ................ 221
Figure 6-3. User-Defined Radix “States” in the List Window ................... 221
Figure 6-4. Setting the Global Signal Radix ......................................... 223
Figure 6-5. Fixed Point Radix Dialog Box ............................................. 224
Figure 7-1. Preferences Dialog Box .................................................... 226
Figure 7-2. Modifying Signal Display Attributes in the Wave Window ........ 228
Table 1-1. Graphic Elements of Toolbar in Find Mode ........................................ 23
Table 1-2. Graphic Elements of Toolbar in Filter Mode ...................................... 25
Table 1-3. Information Displayed in Status Bar .................................................. 32
Table 1-4. Design Object Icons ................................................................. 33
Table 1-5. Icon Shapes and Design Object Types ............................................. 33
Table 2-1. File Menu — Item Description ........................................................ 37
Table 2-2. Edit Menu — Item Description ....................................................... 39
Table 2-3. View Menu — Item Description ....................................................... 40
Table 2-4. Compile Menu — Item Description .................................................. 40
Table 2-5. Simulate Menu — Item Description .................................................. 41
Table 2-6. Add Menu — Item Description ......................................................... 42
Table 2-7. Tools Menu — Item Description ....................................................... 42
Table 2-8. Layout Menu — Item Description ..................................................... 42
Table 2-9. Bookmarks Menu — Item Description ............................................. 43
Table 2-10. Window Menu — Item Description ............................................... 43
Table 2-11. Help Menu — Item Description ....................................................... 44
Table 3-1. Bookmarks Toolbar Buttons ........................................................... 47
Table 3-2. Compile Toolbar Buttons ............................................................... 48
Table 3-3. Dataflow Toolbar Buttons .............................................................. 49
Table 3-4. Help Toolbar Buttons ................................................................. 50
Table 3-5. Layout Toolbar Buttons ............................................................... 50
Table 3-6. Memory Toolbar Buttons ............................................................... 51
Table 3-7. Mode Toolbar Buttons ................................................................. 51
Table 3-8. Objectfilter Toolbar Buttons .......................................................... 52
Table 3-9. Process Toolbar Buttons ............................................................... 53
Table 3-10. Schematic Toolbar Buttons .......................................................... 53
Table 3-11. Simulate Toolbar Buttons ............................................................. 54
Table 3-12. Source Toolbar Buttons ............................................................... 56
Table 3-13. Standard Toolbar Buttons ............................................................ 56
Table 3-14. Step Toolbar Buttons ................................................................. 59
Table 3-15. Wave Toolbar Buttons ............................................................... 60
Table 3-16. Wave Cursor Toolbar Buttons ..................................................... 62
Table 3-17. Wave Edit Toolbar Buttons .......................................................... 63
Table 3-18. Wave Expand Time Toolbar Buttons ............................................ 64
Table 3-19. Zoom Toolbar Buttons ............................................................... 64
Table 3-20. Compile Toolbar Tab Buttons ....................................................... 67
Table 3-21. Debug Toolbar Tab Buttons .......................................................... 68
Table 3-22. Edit Toolbar Tab Buttons ............................................................. 72
Table 3-23. Home Toolbar Tab Buttons ......................................................... 73
Table 3-24. Layout Toolbar Tab Buttons ........................................................ 76
List of Tables

Table 3-25. Schematic Toolbar Tab Buttons ........................................ 76
Table 3-26. Simulate Toolbar Tab Buttons ......................................... 78
Table 3-27. Toolbar Tab Popup Menu ................................................ 81
Table 3-28. Pre-10.3 Toolbar Mapping to Toolbar Tab Location .............. 83
Table 4-1. GUI Windows .................................................................. 85
Table 4-2. Call Stack Window Columns .............................................. 88
Table 4-3. Commands Related to the Call Stack Window ......................... 89
Table 4-4. Class Graph Window Popup Menu ...................................... 92
Table 4-5. Class Instances Window Popup Menu .................................. 95
Table 4-6. Class Tree Window Icons ................................................ 97
Table 4-7. Class Tree Window Columns .............................................. 97
Table 4-8. Class Tree Window Popup Menu ........................................ 97
Table 4-9. Files Window Columns .................................................... 107
Table 4-10. Files Window Popup Menu ............................................. 107
Table 4-11. Files Menu .................................................................. 107
Table 4-12. Library Window Columns ............................................... 109
Table 4-13. Library Window Popup Menu .......................................... 109
Table 4-14. List Window Popup Menu ................................................ 113
Table 4-15. Actions for Time Markers ............................................... 116
Table 4-16. Triggering Options ....................................................... 128
Table 4-17. Locals Window Columns ................................................ 133
Table 4-18. Locals Window Popup Menu ........................................... 133
Table 4-19. Memory Data Popup Menu — Address Pane ....................... 136
Table 4-21. Memory Data Menu ..................................................... 137
Table 4-20. Memory Data Popup Menu — Data Pane ............................ 137
Table 4-22. Memory Identification — ModelSim ................................ 140
Table 4-23. Memory List Window Columns ....................................... 142
Table 4-24. Memory List Popup Menu .............................................. 142
Table 4-25. Memories Menu ........................................................... 142
Table 4-26. Message Viewer Tasks ................................................... 148
Table 4-27. Message Viewer Window Columns ................................... 149
Table 4-28. Message Viewer Window Popup Menu ............................... 150
Table 4-29. Columns in the Objects Window ...................................... 155
Table 4-30. Objects Window Popup Menu ......................................... 155
Table 4-31. Object Window Toolbar Buttons ...................................... 156
Table 4-32. Processes Window Column Descriptions ............................ 161
Table 4-33. Structure Window Popup Menu ....................................... 179
Table 4-34. Columns in the Structure Window .................................... 180
Table 4-35. Watch Window Popup Menu ........................................... 191
Table 4-36. Watch Window Menu ................................................... 192
Table 4-37. Analog Sidebar Icons .................................................... 201
Table 4-38. Window Icons .............................................................. 201
Table 5-1. Mouse Shortcuts ............................................................. 207
Table 5-2. Keyboard Shortcuts .......................................................... 208
Table 5-3. List Window Keyboard Shortcuts ....................................... 211
<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 5-4. Wave Window Mouse Shortcuts</td>
<td>211</td>
</tr>
<tr>
<td>Table 5-5. Wave Window Keyboard Shortcuts</td>
<td>212</td>
</tr>
<tr>
<td>Table 7-1. Default ListTranslateTable Values</td>
<td>227</td>
</tr>
<tr>
<td>Table 7-2. Default LogicStyleTable Values</td>
<td>227</td>
</tr>
</tbody>
</table>
The ModelSim 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 ModelSim Main window.

*Figure 1-1. Graphical User Interface*
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 ModelSim, 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 ModelSim will remember your settings upon subsequent invocations. You can restore ModelSim 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 1-2. Window Header Handle**

   ![objects_tab](image)

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 1-3. Tab Handle**

   ![objects_tab](image)

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.
Overview
General GUI Features

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 1-4. Window Undock Button

Column-Based Windows

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

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

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 Bookmark Options Dialog Box.

- **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 Bookmark Options Dialog Box.
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 1-5).

Figure 1-5. 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 1-6).
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:
# 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 (see `Saving and Restoring Breakpoints`) when invoked with the `do` command in subsequent simulation runs. The syntax is:

```
write format restart <filename>
```

If the `ShutdownFile modelsim.ini` variable is set to this `.do` filename, it will call the `write format restart` command upon exit.

# 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 ModelSim 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 ModelSim 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 1-7. The filtering function is denoted by a “Contains” field (Figure 1-8).

Windows that support both Find (Figure 1-7) and Filter modes (Figure 1-8) 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 1-1.

---

**Note**

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

---

**Table 1-1. Graphic Elements of Toolbar in Find Mode**

<table>
<thead>
<tr>
<th>Graphic Element</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Find Icon]</td>
<td>opens the find toolbar in the active window</td>
</tr>
</tbody>
</table>
# Overview

## General GUI 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 1-9). Refer to Finding Items in the Structure Window for more information.

## Table 1-1. Graphic Elements of Toolbar in Find Mode (cont.)

<table>
<thead>
<tr>
<th>Graphic Element</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Close" /></td>
<td>closes the find toolbar</td>
</tr>
<tr>
<td><img src="image" alt="Find entry field" /></td>
<td>allows entry of find parameters</td>
</tr>
<tr>
<td><img src="image" alt="Find Options" /></td>
<td>opens the Find Options popup menu at the bottom of the active window. The contents of the menu changes for each window.</td>
</tr>
<tr>
<td><img src="image" alt="Clear Entry Field" /></td>
<td>clears the entry field</td>
</tr>
<tr>
<td><img src="image" alt="Execute Search" /></td>
<td>initiates the search</td>
</tr>
<tr>
<td><img src="image" alt="Toggle Search Direction" /></td>
<td>toggles search direction upward or downward through the active window</td>
</tr>
<tr>
<td><img src="image" alt="Find All Matches; Bookmark All Matches (for Source window only)" /></td>
<td>highlights every occurrence of the find item; for the Source window only, places a blue flag (bookmark) at every occurrence of the find item</td>
</tr>
<tr>
<td><img src="image" alt="Search For" /></td>
<td>Click and hold the button to open a drop down menu with the following options: • Instance • Design Unit</td>
</tr>
<tr>
<td><img src="image" alt="Match Case" /></td>
<td>search must match the case of the text entered in the Find field</td>
</tr>
<tr>
<td><img src="image" alt="Exact (whole word)" /></td>
<td>searches for whole words that match those entered in the Find field</td>
</tr>
<tr>
<td><img src="image" alt="Regular Expression" /></td>
<td>Searches for a regular expression; Source window only.</td>
</tr>
<tr>
<td><img src="image" alt="Wrap Search" /></td>
<td>Searches from cursor to bottom of window then continues search from top of the window.</td>
</tr>
</tbody>
</table>
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 1-2. Graphic Elements of Toolbar in Filter Mode

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Filter Regular Expression</td>
<td>None</td>
<td>A drop down menu that allows you to set the wildcard mode. A text entry box for your filter string.</td>
</tr>
<tr>
<td></td>
<td>Clear Filter</td>
<td>None</td>
<td>Clears the text entry box and removes the filter from the active window.</td>
</tr>
</tbody>
</table>

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>
Overview

General Visual Elements

\(<x>\) — 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 1-10).

Figure 1-10. 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

The following sections outline the GUI terminology used in this manual.

- **Menu Bar**
- **Toolbar Frame**
- **Toolbar**
- **Window**
- **Tab Group**
- **Pane**

The Main window is the primary access point in the GUI. Figure 1-11 shows an example of the Main window during a simulation run.

**Figure 1-11. 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 1-12 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 1-12. Main Window — Menu Bar**

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

**Figure 1-13. Main Window — Toolbar Frame**

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 1-14** highlights the Compile toolbar in the toolbar frame.

**Figure 1-14. Highlighting the Compile Toolbar**
Window

ModelSim 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 1-15 shows an example of a layout with five windows visible; the Structure, Objects, Processes, Wave and Transcript 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 1-16 shows a tab group of the Library, Files, Capacity and Structure windows, with the Structure (sim) window visible.
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 1-17 shows the Wave window with its the three panes.
Overview
General Visual Elements

Figure 1-17. Wave Window Panes

Main Window Status Bar

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

Table 1-3. Information Displayed in Status Bar

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project</td>
<td>name of the current project</td>
</tr>
<tr>
<td>Now</td>
<td>the current simulation time</td>
</tr>
</tbody>
</table>
Design Object Icons and Their Meanings

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

Table 1-4 shows the icon colors and the languages they indicate.

<table>
<thead>
<tr>
<th>Icon color</th>
<th>Design Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>light blue</td>
<td>Verilog or SystemVerilog</td>
</tr>
<tr>
<td>dark blue</td>
<td>VHDL</td>
</tr>
<tr>
<td>orange</td>
<td>virtual object</td>
</tr>
</tbody>
</table>

Here is a list of icon shapes and the design object types they indicate:

<table>
<thead>
<tr>
<th>Icon shape</th>
<th>Example</th>
<th>Design Object Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Square</td>
<td><img src="square.png" alt="Square Icon" /></td>
<td>any scope (VHDL block, Verilog named block, SC module, class, interface, task, function, and so forth.)</td>
</tr>
<tr>
<td>Square and red asterix</td>
<td><img src="asterisk.png" alt="Square Asterisk Icon" /></td>
<td>SystemVerilog object, OVM, and UVM test bench scope or object</td>
</tr>
<tr>
<td>Circle</td>
<td><img src="circle.png" alt="Circle Icon" /></td>
<td>process</td>
</tr>
<tr>
<td>Diamond</td>
<td><img src="diamond.png" alt="Diamond Icon" /></td>
<td>valued object (signals, nets, registers, and so forth.)</td>
</tr>
<tr>
<td>Diamond and yellow pulse on red dot</td>
<td><img src="pulse.png" alt="Diamond Pulse Icon" /></td>
<td>an editable waveform created with the waveform editor</td>
</tr>
</tbody>
</table>
Overview

General Visual Elements

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 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. The windows include the FSM, Objects, Source, and Watch windows.

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 1-19).

<table>
<thead>
<tr>
<th>Icon shape</th>
<th>Example</th>
<th>Design Object Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diamond and red asterix</td>
<td><img src="image" alt="Diamond and red asterix" /></td>
<td>valued object (abstract)</td>
</tr>
<tr>
<td>Diamond and green arrow</td>
<td><img src="image" alt="Diamond and green arrow" /></td>
<td>indicates mode (In, Inout, Out) of an object port</td>
</tr>
<tr>
<td>Triangle</td>
<td><img src="image" alt="Triangle" /></td>
<td>caution sign on comparison object</td>
</tr>
<tr>
<td>Star</td>
<td><img src="image" alt="Star" /></td>
<td>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</td>
</tr>
</tbody>
</table>

Table 1-5. Icon Shapes and Design Object Types (cont.)
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 on the time display in the Current Time label to open a drop down menu (Figure 1-20) or select View > Time Mode.
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

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>
<thead>
<tr>
<th>Table 2-1. File Menu — Item Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Menu Item</strong></td>
</tr>
<tr>
<td>New</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Open</td>
</tr>
<tr>
<td>Load</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Close</td>
</tr>
</tbody>
</table>
### Table 2-1. File Menu — Item Description (cont.)

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
</table>
| **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.  
- Column Layout — apply a previously saved column layout to the active window  
- | |
| **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  
- HTML — opens up a dialog where you can specify the name of an HTML file and the directory where it is saved |
| **Save** | These menu items change based on the active window. |
| **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. |
### Edit Menu

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

#### Table 2-2. Edit Menu — Item Description

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Undo</td>
<td>Alter your previous edit in a Source window.</td>
</tr>
<tr>
<td>Redo</td>
<td></td>
</tr>
<tr>
<td>Cut</td>
<td>Use or remove selected text.</td>
</tr>
<tr>
<td>Copy</td>
<td></td>
</tr>
<tr>
<td>Paste</td>
<td></td>
</tr>
<tr>
<td>Delete</td>
<td>Remove an object from the Wave and List windows</td>
</tr>
<tr>
<td>Clear</td>
<td>Clear the Transcript window</td>
</tr>
<tr>
<td>Select All</td>
<td>Change the selection of items in a window</td>
</tr>
<tr>
<td>Unselect All</td>
<td></td>
</tr>
<tr>
<td>Expand</td>
<td>Expand or collapse hierarchy information</td>
</tr>
<tr>
<td>Goto</td>
<td>Goto a specific line number in the Source window</td>
</tr>
<tr>
<td>Find</td>
<td>Open the find toolbar. Refer to the section “Find and Filter Functions” for more information</td>
</tr>
<tr>
<td>Replace</td>
<td>Find and replace text in a Source window.</td>
</tr>
</tbody>
</table>
View Menu

The View menu provides options for manipulating the GUI windows.

### Table 2-3. View Menu — Item Description

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>window name</td>
<td>Displays the selected window</td>
</tr>
<tr>
<td>New Window</td>
<td>Open additional instances of the Wave, List, or Dataflow windows</td>
</tr>
<tr>
<td>Sort</td>
<td>Change the sort order of the Wave window</td>
</tr>
<tr>
<td>Filter</td>
<td>Filters information from the Objects and Structure windows.</td>
</tr>
<tr>
<td>Justify</td>
<td>Change the alignment of data in the selected window.</td>
</tr>
<tr>
<td>Properties</td>
<td>Displays file property information from the Files or Source windows.</td>
</tr>
</tbody>
</table>

Compile Menu

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

### Table 2-4. Compile Menu — Item Description

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compile</td>
<td>Compile source files</td>
</tr>
<tr>
<td>Compile Options</td>
<td>Set various compile options.</td>
</tr>
<tr>
<td>Compile All</td>
<td>Compile all files in the open project. Disabled if you don’t have a project open</td>
</tr>
<tr>
<td>Compile Selected</td>
<td>Compile the files selected in the project tab. Disabled if you don’t have a project open</td>
</tr>
<tr>
<td>Compile Order</td>
<td>Set the compile order of the files in the open project. Disabled if you don’t have a project open</td>
</tr>
</tbody>
</table>
Simulate Menu

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

Table 2-5. Simulate Menu — Item Description

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design Optimization</td>
<td>Open the Design Optimization dialog to configure simulation optimizations</td>
</tr>
<tr>
<td>Start Simulation</td>
<td>Load the selected design unit</td>
</tr>
<tr>
<td>Runtime Options</td>
<td>Set various simulation runtime options</td>
</tr>
<tr>
<td>Run</td>
<td>• Run &lt;default&gt; — run simulation for one default run length; change the run length with <code>Simulate &gt; Runtime Options</code>, or use the Run Length text box on the toolbar</td>
</tr>
<tr>
<td></td>
<td>• Run -All — run simulation until you stop it</td>
</tr>
<tr>
<td></td>
<td>• Continue — continue the simulation</td>
</tr>
<tr>
<td></td>
<td>• Run -Next — run to the next event time</td>
</tr>
<tr>
<td></td>
<td>• Step — single-step the simulator</td>
</tr>
<tr>
<td></td>
<td>• Step -Over — execute without single-stepping through a subprogram call</td>
</tr>
<tr>
<td></td>
<td>• 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, etc.)</td>
</tr>
<tr>
<td>Break</td>
<td>Stop the current simulation run</td>
</tr>
<tr>
<td>End Simulation</td>
<td>Quit the current simulation run</td>
</tr>
</tbody>
</table>
Add Menu

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

Table 2-6. Add Menu — Item Description

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>To Wave</td>
<td>Add information to the Wave window</td>
</tr>
<tr>
<td>To List</td>
<td>Add information to the List window</td>
</tr>
<tr>
<td>To Log</td>
<td>Add information to the Log file</td>
</tr>
<tr>
<td>To Dataflow</td>
<td>Add information to the Dataflow window</td>
</tr>
<tr>
<td>Window Pane</td>
<td>Add an additional pane to the Wave window. You can remove this pane by selecting Wave &gt; Delete Window Pane.</td>
</tr>
</tbody>
</table>

Tools Menu

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

Table 2-7. Tools Menu — Item Description

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breakpoints</td>
<td>Manage breakpoints</td>
</tr>
<tr>
<td>Trace</td>
<td>Perform signal trace actions.</td>
</tr>
<tr>
<td>Dataset Snapshot</td>
<td>Enable periodic saving of simulation data to a .wlf file.</td>
</tr>
<tr>
<td>Tcl</td>
<td>Execute or debug a Tcl macro.</td>
</tr>
<tr>
<td>Wildcard Filter</td>
<td>Refer to the section “Using the WildcardFilter Preference Variable” for more information</td>
</tr>
<tr>
<td>Edit Preferences</td>
<td>Set GUI preference variables. Refer to the section “Setting GUI Preferences” for more information</td>
</tr>
</tbody>
</table>

Layout Menu

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

Table 2-8. Layout Menu — Item Description

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reset</td>
<td>Reset the GUI to the default appearance for the selected layout,</td>
</tr>
</tbody>
</table>
Table 2-8. Layout Menu — Item Description (cont.)

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Save Layout As</td>
<td>Save your reorganized view to a custom layout. Refer to the section “Simulator GUI Layout Customization” for more information.</td>
</tr>
<tr>
<td>Configure</td>
<td>Configure the layout-specific behavior of the GUI. Refer to the section “Configure Window Layouts Dialog Box” for more information.</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete a customized layout. You can not delete any of the five standard layouts.</td>
</tr>
</tbody>
</table>

| layout name     | Select a standard or customized layout.                                      |

Bookmarks Menu

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

Table 2-9. Bookmarks Menu — Item Description

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Clicking this button bookmarks the current view of the Wave window.</td>
</tr>
<tr>
<td>Add Custom</td>
<td>Opens the New Bookmark dialog box.</td>
</tr>
<tr>
<td>Manage</td>
<td>Opens the Manage Bookmarks dialog box.</td>
</tr>
</tbody>
</table>
| 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 2-10. Window Menu — Item Description

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cascade</td>
<td>Arrange all undocked windows. These options do not impact any docked windows.</td>
</tr>
<tr>
<td>Tile Horizontally</td>
<td></td>
</tr>
<tr>
<td>Tile Vertically</td>
<td></td>
</tr>
</tbody>
</table>
Menus

Help Menu

The Help menu provides access to various types of documentation.

### Table 2-11. Help Menu — Item Description

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>About</td>
<td>Display ModelSim application information.</td>
</tr>
<tr>
<td>Release Notes</td>
<td>Display the current Release Notes in the ModelSim Notepad editor. You can find past release notes in the <code>&lt;install_dir&gt;/docs/rlsnote/</code> directory.</td>
</tr>
<tr>
<td>Welcome Window</td>
<td>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.</td>
</tr>
</tbody>
</table>
Table 2-11. Help Menu — Item Description (cont.)

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command Completion</td>
<td>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.</td>
</tr>
<tr>
<td>Register File Types</td>
<td>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.</td>
</tr>
<tr>
<td>ModelSim Documentation - PDF Bookcase</td>
<td>Open the PDF-based portal for the most commonly used PDF documents.</td>
</tr>
<tr>
<td>Tcl Help</td>
<td>Open the Tcl command reference (man pages) in Windows help format.</td>
</tr>
<tr>
<td>Tcl Syntax</td>
<td>Open the Tcl syntax documentation in your web browser.</td>
</tr>
<tr>
<td>Tcl Man pages</td>
<td>Open the Tcl/Tk manual in your web browser.</td>
</tr>
<tr>
<td>Technotes</td>
<td>Open a technical note in the ModelSim Notepad editor.</td>
</tr>
</tbody>
</table>
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.

**Bookmarks Toolbar**

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

![Figure 3-1. Bookmarks Toolbar](image)

**Table 3-1. Bookmarks Toolbar Buttons**

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
</table>
| ![Add Bookmark](image) | 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](image) | 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 |
Toolbars

Compile Toolbar

The Compile toolbar provides access to compile and simulation actions.

Figure 3-2. Compile Toolbar

Table 3-2. Compile Toolbar Buttons

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Manage Bookmarks</td>
<td>None</td>
<td>Displays the Manage Bookmarks dialog box.</td>
</tr>
<tr>
<td></td>
<td>Reload from File</td>
<td>None</td>
<td>Reloads bookmarks from the <code>bookmarks.do</code> file.</td>
</tr>
<tr>
<td></td>
<td>Jump to Bookmark</td>
<td><code>CommandWave window only: bookmark goto wave &lt;name&gt;</code></td>
<td>Displays bookmarks grouped by window. Select the bookmark you want to display.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Dataflow Toolbar

The Dataflow toolbar provides access to various tools to use in the Dataflow window.
Figure 3-3. Dataflow 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 3-4. Help Toolbar

Table 3-3. Dataflow Toolbar Buttons

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Trace Input Net to Event</td>
<td>Menu: Tools &gt; Trace &gt; Trace next event</td>
<td>Move the next event cursor to the next input event driving the selected output.</td>
</tr>
<tr>
<td></td>
<td>Trace Set</td>
<td>Menu: Tools &gt; Trace &gt; Trace event set</td>
<td>Jump to the source of the selected input event.</td>
</tr>
<tr>
<td></td>
<td>Trace Reset</td>
<td>Menu: Tools &gt; Trace &gt; Trace event reset</td>
<td>Return the next event cursor to the selected output.</td>
</tr>
<tr>
<td></td>
<td>Trace Net to Driver of X</td>
<td>Menu: Tools &gt; Trace &gt; TraceX</td>
<td>Step back to the last driver of an unknown value.</td>
</tr>
<tr>
<td></td>
<td>Expand Net to all Drivers</td>
<td>None</td>
<td>Display driver(s) of the selected signal, net, or register.</td>
</tr>
<tr>
<td></td>
<td>Expand Net to all Drivers and Readers</td>
<td>None</td>
<td>Display driver(s) and reader(s) of the selected signal, net, or register.</td>
</tr>
<tr>
<td></td>
<td>Expand Net to all Readers</td>
<td>None</td>
<td>Display reader(s) of the selected signal, net, or register.</td>
</tr>
<tr>
<td></td>
<td>Show Wave</td>
<td>Menu: Dataflow &gt; Show Wave</td>
<td>Display the embedded wave viewer pane.</td>
</tr>
</tbody>
</table>
Table 3-4. Help Toolbar Buttons

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Help" /></td>
<td>Search Documentation</td>
<td>None</td>
<td>A text entry box for your search string.</td>
</tr>
<tr>
<td><img src="image" alt="Search" /></td>
<td>Search Documentation</td>
<td><strong>Hotkey</strong>: Enter</td>
<td>Activates the search for the term you entered into the text entry box.</td>
</tr>
</tbody>
</table>

**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 3-5. Layout Toolbar](image)

Table 3-5. Layout Toolbar Buttons

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
</table>
| ![Simulate](image) | Change Layout                 | **Menu**: Layout >  
<layoutName> | A dropdown box that allows you to select a GUI layout.  
- NoDesign  
- Simulate  |

**Memory Toolbar**

The Memory toolbar provides access to common functions.

![Figure 3-6. Memory Toolbar](image)
# Mode Toolbar

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

## Table 3-6. Memory Toolbar Buttons

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Split Screen</td>
<td><strong>Menu: Memory &gt; Split Screen</strong></td>
<td>Splits the memory window.</td>
</tr>
<tr>
<td></td>
<td>Goto Address</td>
<td></td>
<td>Highlights the first element of the specified address.</td>
</tr>
</tbody>
</table>

## Figure 3-7. Mode Toolbar

![Mode Toolbar](image)

## Table 3-7. Mode Toolbar Buttons

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Select Mode</td>
<td><strong>Menu: Dataflow &gt; Mouse Mode &gt; Select Mode</strong></td>
<td>Set the left mouse button to select mode and middle mouse button to zoom mode.</td>
</tr>
<tr>
<td></td>
<td>Zoom Mode</td>
<td><strong>Menu: Dataflow &gt; Mouse Mode &gt; Zoom Mode</strong></td>
<td>Set left mouse button to zoom mode and middle mouse button to pan mode.</td>
</tr>
<tr>
<td></td>
<td>Pan Mode</td>
<td><strong>Menu: Dataflow &gt; Mouse Mode &gt; Pan Mode</strong></td>
<td>Set left mouse button to pan mode and middle mouse button to zoom mode.</td>
</tr>
<tr>
<td></td>
<td>Two Cursor Mode</td>
<td><strong>Menu: Wave &gt; Mouse Mode &gt; Two Cursor</strong></td>
<td>Sets two cursors in Wave window. First cursor moves with LMB, second cursor with MMB.</td>
</tr>
<tr>
<td></td>
<td>Edit Mode</td>
<td><strong>Menu: Wave or Dataflow &gt; Mouse Mode &gt; Edit Mode</strong></td>
<td>Set mouse to Edit Mode, where you drag the left mouse button to select a range and drag the middle mouse button to zoom.</td>
</tr>
<tr>
<td></td>
<td>Stop Drawing</td>
<td>None</td>
<td>Halt any drawing currently happening in the window.</td>
</tr>
</tbody>
</table>
Objectfilter Toolbar

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

**Figure 3-8. Objectfilter Toolbar**

![Objectfilter Toolbar](image)

**Table 3-8. Objectfilter Toolbar Buttons**

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="I" /></td>
<td>View Inputs Only</td>
<td>None</td>
<td>Changes the view of the Objects Window to show inputs.</td>
</tr>
<tr>
<td><img src="image" alt="O" /></td>
<td>View Outputs Only</td>
<td>None</td>
<td>Changes the view of the Objects Window to show outputs.</td>
</tr>
<tr>
<td><img src="image" alt="I/O" /></td>
<td>View Inouts Only</td>
<td>None</td>
<td>Changes the view of the Objects Window to show inouts.</td>
</tr>
<tr>
<td><img src="image" alt="i" /></td>
<td>View Internal Signals</td>
<td>None</td>
<td>Changes the view of the Objects Window to show Internal Signals.</td>
</tr>
<tr>
<td><img src="image" alt="ALL" /></td>
<td>Reset All Filters</td>
<td>None</td>
<td>Clears the filtering of Objects Window entries and displays all objects.</td>
</tr>
<tr>
<td><img src="image" alt="Change Filter" /></td>
<td>Change Filter</td>
<td>None</td>
<td>Opens the Filter Objects dialog box.</td>
</tr>
</tbody>
</table>

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 3-9. Process Toolbar**

![Process Toolbar](image)
Table 3-9. Process Toolbar Buttons

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>View Active Processes</td>
<td>Menu: Process &gt; Active</td>
<td>Changes the view of the Processes Window to only show active processes.</td>
</tr>
<tr>
<td></td>
<td>View Processes in Region</td>
<td>Menu: Process &gt; In Region</td>
<td>Changes the view of the Processes window to only show processes in the active region.</td>
</tr>
<tr>
<td></td>
<td>View Processes for the Design</td>
<td>Menu: Process &gt; Design</td>
<td>Changes the view of the Processes window to show processes in the design.</td>
</tr>
<tr>
<td></td>
<td>View Process hierarchy</td>
<td>Menu: Process &gt; Hierarchy</td>
<td>Changes the view of the Processes window to show process hierarchy.</td>
</tr>
</tbody>
</table>

Table 3-10. Schematic Toolbar Buttons

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
</table>
| ![Image](image)   | Remove All Highlights | Menu: Dataflow > Remove Highlight or Schematic > Edit > Remove Highlight | Clear the green highlighting identifying the path you’ve 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 |
Simulate Toolbar

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

**Figure 3-11. Simulate Toolbar**

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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. |

### Table 3-11. Simulate Toolbar Buttons

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="source_hyperlink.png" alt="Source" /></td>
<td>Source Hyperlinking</td>
<td>None</td>
<td>Toggles display of hyperlinks in design source files.</td>
</tr>
</tbody>
</table>
| ![Environment Up](environment_up.png) | Environment Up   | **Command:** env ..  
**Menu:** File > Environment | Changes your environment up one level of hierarchy.                           |
| ![Environment Back](environment_back.png) | Environment Back | **Command:** env -back  
**Menu:** File > Environment | Change your environment to its previous location.                           |
| ![Environment Forward](environment_forward.png) | Environment Forward | **Command:** env -forward  
**Menu:** File > Environment | Change your environment forward to a previously selected environment.         |
Table 3-11. Simulate Toolbar Buttons (cont.)

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Restart Image" /></td>
<td>Restart</td>
<td><strong>Command: restart</strong>&lt;br&gt;<strong>Menu: Simulate &gt; Run &gt; Restart</strong></td>
<td>Reload the design elements and reset the simulation time to zero, with the option of maintaining various settings and objects.</td>
</tr>
<tr>
<td><img src="image2" alt="Run Length Image" /></td>
<td>Run Length</td>
<td><strong>Command: run</strong>&lt;br&gt;<strong>Menu: Simulate &gt; Runtime Options</strong></td>
<td>Specify the run length for the current simulation.</td>
</tr>
<tr>
<td><img src="image3" alt="Run Image" /></td>
<td>Run</td>
<td><strong>Command: run</strong>&lt;br&gt;<strong>Menu: Simulate &gt; Run &gt; Run default_run_length</strong></td>
<td>Run the current simulation for the specified run length.</td>
</tr>
<tr>
<td><img src="image4" alt="Continue Run Image" /></td>
<td>Continue Run</td>
<td><strong>Command: run</strong>&lt;br&gt;-continue&lt;br&gt;<strong>Menu: Simulate &gt; Run &gt; Continue</strong></td>
<td>Continue the current simulation run until the end of the specified run length or until it hits a breakpoint or specified break event.</td>
</tr>
<tr>
<td><img src="image5" alt="Run All Image" /></td>
<td>Run All</td>
<td><strong>Command: run -all</strong>&lt;br&gt;<strong>Menu: Simulate &gt; Run &gt; Run -All</strong></td>
<td>Run the current simulation forever, or until it hits a breakpoint or specified break event.</td>
</tr>
<tr>
<td><img src="image6" alt="Break Image" /></td>
<td>Break</td>
<td><strong>Menu: Simulate &gt; Break</strong>&lt;br&gt;<strong>Hotkey: Break</strong></td>
<td>Immediate stop of a compilation, elaboration, or simulation run. Similar to hitting a breakpoint if the simulator is in the middle of a process.</td>
</tr>
<tr>
<td><img src="image7" alt="Stop -sync Image" /></td>
<td>Stop -sync</td>
<td>None</td>
<td>Stop simulation the next time time/delta is advanced.</td>
</tr>
<tr>
<td><img src="image8" alt="Performance Profiling Image" /></td>
<td>Performance Profiling</td>
<td><strong>Menu: Tools &gt; Profile &gt; Performance</strong></td>
<td>Enable collection of statistical performance data.</td>
</tr>
<tr>
<td><img src="image9" alt="Memory Profiling Image" /></td>
<td>Memory Profiling</td>
<td><strong>Menu: Tools &gt; Profile &gt; Memory</strong></td>
<td>Enable collection of memory usage data.</td>
</tr>
<tr>
<td><img src="image10" alt="Edit Breakpoints Image" /></td>
<td>Edit Breakpoints</td>
<td><strong>Menu: Tools &gt; Breakpoint</strong></td>
<td>Enable breakpoint editing, loading, and saving.</td>
</tr>
</tbody>
</table>

**Source Toolbar**

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

Standard Toolbar

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

Table 3-12. Source Toolbar Buttons

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Clear Bookmarks</td>
<td></td>
</tr>
</tbody>
</table>

Table 3-13. Standard Toolbar Buttons

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="New File Icon" /></td>
<td>New File</td>
<td>Menu: File &gt; New &gt; Source</td>
<td>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 • SystemVerilog • Do • Other • Set Default Action</td>
</tr>
<tr>
<td><img src="image" alt="Open Icon" /></td>
<td>Open</td>
<td>Menu: File &gt; Open</td>
<td>Opens the Open File dialog</td>
</tr>
</tbody>
</table>
### Table 3-13. Standard Toolbar Buttons (cont.)

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Save" /></td>
<td>Save</td>
<td><strong>Menu</strong>: File &gt; Save</td>
<td>Saves the contents of the active window or Saves the current wave window display and signal preferences to a DO file script.</td>
</tr>
</tbody>
</table>
| ![Reload](image) | Reload                | **Command**: Dataset Restart  
**Menu**: File > Datasets  | Reload the current dataset.                                               |
| ![Print](image) | Print                 | **Menu**: File > Print | Opens the Print dialog box.                                                |
| ![Cut](image)   | Cut                   | **Menu**: Edit > Cut   
**Hotkey**: Ctrl+x |                                                                                |
| ![Copy](image)  | Copy                  | **Menu**: Edit > Copy  
**Hotkey**: Ctrl+c |                                                                                |
| ![Paste](image) | Paste                 | **Menu**: Edit > Paste  
**Hotkey**: Ctrl+v |                                                                                |
| ![Undo](image)  | Undo                  | **Menu**: Edit > Undo   
**Hotkey**: Ctrl+z |                                                                                |
| ![Redo](image)  | Redo                  | **Menu**: Edit > Redo   
**Hotkey**: Ctrl+y |                                                                                |
| ![Add Selected to Window](image) | Add Selected to Window  
**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](image)  | Find                  | **Menu**: Edit > Find   
**Hotkey**: Ctrl+f (Windows) or Ctrl+s (UNIX) | Opens the Find dialog box.                                                |
| ![Collapse All](image) | 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. |
Toolbars

Step Toolbar

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, 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, as shown in Figure 3-14.

Figure 3-14. The 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 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 3-14.

- You can change the default

Step Toolbar

The Step toolbar allows you to step through your source code.
Figure 3-15. Step Toolbar

Table 3-14. Step Toolbar Buttons

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
</table>
| ![Button Icon](image1.png) | Step Into        | Command: step  
Menu: Simulate > Run > Step | Step the current simulation to the next statement. |
| ![Button Icon](image2.png) | 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. |
| ![Button Icon](image3.png) | Step Out         | Command: step -out | Step the current simulation out of the current function or procedure. |
| ![Button Icon](image4.png) | Step Into Current | Command: step -current | Step the simulation into the current instance, process, or thread. |
| ![Button Icon](image5.png) | Step Over Current | Command: step -over -current | Step the simulation over the current instance, process, or thread. |
| ![Button Icon](image6.png) | 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 3-16. Wave Toolbar
### Table 3-15. Wave Toolbar Buttons

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Show Drivers" /></td>
<td>Show Drivers</td>
<td>None</td>
<td>Display driver(s) of the selected signal, net, or register in the Dataflow and Source windows. Display drivers only in the Source window. The source window is not shown if there are no drivers.</td>
</tr>
<tr>
<td><img src="image" alt="Show Readers" /></td>
<td>Show Readers</td>
<td>None</td>
<td>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.</td>
</tr>
<tr>
<td><img src="image" alt="Add Contributing Signals" /></td>
<td>Add Contributing Signals</td>
<td><strong>Menu</strong>: Add &gt; To Wave &gt; Contributing Signals</td>
<td>Creates a group labeled <strong>Contributors: &lt;name&gt;</strong>, where &lt;name&gt; is the name of the currently selected signal. This group contains the inputs to the process driving &lt;name&gt;.</td>
</tr>
<tr>
<td><img src="image" alt="Wave Search Box" /></td>
<td>Wave Search Box</td>
<td>Click on the box when in transition mode (Falling Edge, Rising Edge, or Any Transition) to cycle through these options.</td>
<td>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.</td>
</tr>
<tr>
<td><img src="image" alt="Search Previous/Next" /></td>
<td>Search Previous/Next</td>
<td>Previous: Shift+Enter Next: enter</td>
<td>Searches for the next occurrence of the string, either backward or forward in time, from the cursor.</td>
</tr>
</tbody>
</table>
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 Cursor Toolbar

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

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Search Options]</td>
<td>Search Options</td>
<td></td>
<td>Dropdown button to:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Change Mode: value, rising edge, falling edge, or any transition.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Display the Wave Signal Search Dialog Box for advanced search behavior.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Clear the value history.</td>
</tr>
</tbody>
</table>

Figure 3-17. Wave Cursor Toolbar
Toolbars
Wave Edit Toolbar

Table 3-16. Wave Cursor Toolbar Buttons

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Insert Cursor]</td>
<td>Insert Cursor</td>
<td>None</td>
<td>Adds a new cursor to the active Wave window.</td>
</tr>
<tr>
<td>![Delete Cursor]</td>
<td>Delete Cursor</td>
<td>Menu: Wave &gt; Delete Cursor</td>
<td>Deletes the active cursor.</td>
</tr>
<tr>
<td>![Find Previous Transition]</td>
<td>Find Previous Transition</td>
<td>Menu: Edit &gt; Signal Search Hotkey: Shift + Tab</td>
<td>Moves the active cursor to the previous signal value change for the selected signal.</td>
</tr>
<tr>
<td>![Find Next Transition]</td>
<td>Find Next Transition</td>
<td>Menu: Edit &gt; Signal Search Hotkey: Tab</td>
<td>Moves the active cursor to the next signal value change for the selected signal.</td>
</tr>
<tr>
<td>![Find Previous Falling Edge]</td>
<td>Find Previous Falling Edge</td>
<td>Menu: Edit &gt; Signal Search</td>
<td>Moves the active cursor to the previous falling edge for the selected signal.</td>
</tr>
<tr>
<td>![Find Next Falling Edge]</td>
<td>Find Next Falling Edge</td>
<td>Menu: Edit &gt; Signal Search</td>
<td>Moves the active cursor to the next falling edge for the selected signal.</td>
</tr>
<tr>
<td>![Find Previous Rising Edge]</td>
<td>Find Previous Rising Edge</td>
<td>Menu: Edit &gt; Signal Search</td>
<td>Moves the active cursor to the previous rising edge for the selected signal.</td>
</tr>
<tr>
<td>![Find Next Rising Edge]</td>
<td>Find Next Rising Edge</td>
<td>Menu: Edit &gt; Signal Search</td>
<td>Moves the active cursor to the next rising edge for the selected signal.</td>
</tr>
</tbody>
</table>

Wave Edit Toolbar

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

Figure 3-18. Wave Edit Toolbar

![Wave Edit Toolbar Icon]
Table 3-17. Wave Edit Toolbar Buttons

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
</table>
| ![Insert Pulse](image) | Insert Pulse     | **Menu**: Wave > Wave Editor > Insert Pulse  
**Command**: wave edit insert_pulse                        | Insert a transition at the selected time.                       |
| ![Delete Edge](image) | Delete Edge      | **Menu**: Wave > Wave Editor > Delete Edge  
**Command**: wave edit delete                                    | Delete the selected transition.                                |
| ![Invert](image) | Invert           | **Menu**: Wave > Wave Editor > Invert  
**Command**: wave edit invert                                    | Invert the selected section of the waveform.                    |
| ![Mirror](image) | Mirror           | **Menu**: Wave > Wave Editor > Mirror  
**Command**: wave edit mirror                                    | Mirror the selected section of the waveform.                    |
| ![Change Value](image) | Change Value   | **Menu**: Wave > Wave Editor > Value  
**Command**: wave edit change_value                                 | Change the value of the selected section of the waveform.      |
| ![Stretch Edge](image) | Stretch Edge    | **Menu**: Wave > Wave Editor > Stretch Edge  
**Command**: wave edit stretch                                   | Move the selected edge by increasing/decreasing waveform duration. |
| ![Move Edge](image) | Move Edge        | **Menu**: Wave > Wave Editor > Move Edge  
**Command**: wave edit move                                       | Move the selected edge without increasing/decreasing waveform duration. |
| ![Extend All Waves](image) | 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 3-19. Wave Expand Time Toolbar
Table 3-18. Wave Expand Time Toolbar Buttons

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Expanded Time Off" /></td>
<td>Expanded Time Off</td>
<td>Menu: Wave &gt; Expanded Time &gt; Off</td>
<td>turns off the expanded time display (default mode)</td>
</tr>
<tr>
<td><img src="image" alt="Expanded Time Deltas Mode" /></td>
<td>Expanded Time Deltas Mode</td>
<td>Menu: Wave &gt; Expanded Time &gt; Deltas Mode</td>
<td>displays delta time steps</td>
</tr>
<tr>
<td><img src="image" alt="Expanded Time Events Mode" /></td>
<td>Expanded Time Events Mode</td>
<td>Menu: Wave &gt; Expanded Time &gt; Events Mode</td>
<td>displays event time steps</td>
</tr>
<tr>
<td><img src="image" alt="Expand All Time" /></td>
<td>Expand All Time</td>
<td>Menu: Wave &gt; Expanded Time &gt; Expand All</td>
<td>expands simulation time over the entire simulation time range, from 0 to current time</td>
</tr>
<tr>
<td><img src="image" alt="Expand Time at Active Cursor" /></td>
<td>Expand Time at Active Cursor</td>
<td>Menu: Wave &gt; Expanded Time &gt; Expand Cursor</td>
<td>expands simulation time at the simulation time of the active cursor</td>
</tr>
<tr>
<td><img src="image" alt="Collapse All Time" /></td>
<td>Collapse All Time</td>
<td>Menu: Wave &gt; Expanded Time &gt; Collapse All</td>
<td>collapses simulation time over entire simulation time range</td>
</tr>
<tr>
<td><img src="image" alt="Collapse Time at Active Cursor" /></td>
<td>Collapse Time at Active Cursor</td>
<td>Menu: Wave &gt; Expanded Time &gt; Collapse Cursor</td>
<td>collapses simulation time at the simulation time of the active cursor</td>
</tr>
</tbody>
</table>

**Zoom Toolbar**

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

Figure 3-20. Zoom Toolbar

![Zoom Toolbar](image)

Table 3-19. Zoom Toolbar Buttons

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Zoom In" /></td>
<td>Zoom In</td>
<td>Menu: Wave &gt; Zoom &gt; Zoom In Hotkey: i, I, or +</td>
<td>Zooms in by a factor of 2x.</td>
</tr>
</tbody>
</table>
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 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 3-21).

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
</table>
| ![Zoom Out Icon] | Zoom Out | **Menu**: Wave > Zoom > Zoom Out  
**Hotkey**: o, O, or - | Zooms out by a factor of 2x. |
| ![Zoom Full Icon] | 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 Icon] | 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 Icon] | Zoom between Cursors | | Zooms in or out to show the range between the last two selected cursors. |
| ![Zoom Other Window Icon] | Zoom Other Window | | Changes the view in additional instances of the Wave window to match the view of the active Wave window. |
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 3-21) 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**
The Compile Toolbar Tab provides access to compile actions.
Debug Toolbar Tab

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

Table 3-20. Compile Toolbar Tab Buttons

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Compilation Icon]</td>
<td>Compile</td>
<td>Command: vcom or vlog</td>
<td>Opens the Compile Source Files dialog box.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Menu: Compile &gt; Compile</td>
<td></td>
</tr>
<tr>
<td>![Compilation All Icon]</td>
<td>Compile All</td>
<td>Command: vcom or vlog</td>
<td>Compiles all files in the open project.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Menu: Compile &gt; Compile all</td>
<td></td>
</tr>
<tr>
<td>![Simulation Icon]</td>
<td>Simulate</td>
<td>Command: vsim</td>
<td>Opens the Start Simulation dialog box.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Menu: Simulate &gt; Start Simulation</td>
<td></td>
</tr>
<tr>
<td>![Break Icon]</td>
<td>Break</td>
<td>Menu: Simulate &gt; Break</td>
<td>Stop a compilation, elaboration, or the current simulation run.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hotkey: Break</td>
<td></td>
</tr>
<tr>
<td>![Compile Out of Date Icon]</td>
<td>Compile Out of Date</td>
<td></td>
<td>Recompile if changes have been made to any source files.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3-23. Debug Tab

![Debug Toolbar Image]
### Table 3-21. Debug Toolbar Tab Buttons

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Zoom In icon]</td>
<td>Zoom In</td>
<td><strong>Menu:</strong> Wave &gt; Zoom &gt;</td>
<td>Zooms in by a factor of 2x.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Zoom In</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Hotkey:</strong> i, I, or +</td>
<td></td>
</tr>
<tr>
<td>![Zoom Out icon]</td>
<td>Zoom Out</td>
<td><strong>Menu:</strong> Wave &gt; Zoom &gt;</td>
<td>Zooms out by a factor of 2x.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Zoom Out</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Hotkey:</strong> o, O, or -</td>
<td></td>
</tr>
<tr>
<td>![Zoom Full icon]</td>
<td>Zoom Full</td>
<td><strong>Menu:</strong> Wave &gt; Zoom &gt;</td>
<td>Zooms to show the full length of the simulation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Zoom Full</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Hotkey:</strong> f or F</td>
<td></td>
</tr>
<tr>
<td>![Zoom in on Active Cursor icon]</td>
<td>Zoom in on Active Cursor</td>
<td><strong>Menu:</strong> Wave &gt; Zoom &gt;</td>
<td>Zooms in by a factor of 2x, centered on the active cursor.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Zoom Cursor</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Hotkey:</strong> c or C</td>
<td></td>
</tr>
<tr>
<td>![Zoom between Cursors icon]</td>
<td>Zoom between Cursors</td>
<td></td>
<td>Zooms in or out to show the range between the last two selected cursors.</td>
</tr>
<tr>
<td>![Zoom Other Window icon]</td>
<td>Zoom Other Window</td>
<td></td>
<td>Changes the view in additional instances of the Wave window to match the view of the active Wave window.</td>
</tr>
<tr>
<td>![Source Hyperlinking icon]</td>
<td>Source Hyperlinking</td>
<td>None</td>
<td>Toggles display of hyperlinks in design source files.</td>
</tr>
<tr>
<td>![Insert Cursor icon]</td>
<td>Insert Cursor</td>
<td>None</td>
<td>Adds a new cursor to the active Wave window.</td>
</tr>
<tr>
<td>![Delete Cursor icon]</td>
<td>Delete Cursor</td>
<td><strong>Menu:</strong> Wave &gt; Delete Cursor</td>
<td>Deletes the active cursor.</td>
</tr>
<tr>
<td>![Find Previous Transition icon]</td>
<td>Find Previous Transition</td>
<td><strong>Menu:</strong> Edit &gt; Signal Search</td>
<td>Moves the active cursor to the previous signal value change for the selected signal.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Hotkey:</strong> Shift + Tab</td>
<td></td>
</tr>
<tr>
<td>![Find Next Transition icon]</td>
<td>Find Next Transition</td>
<td><strong>Menu:</strong> Edit &gt; Signal Search</td>
<td>Moves the active cursor to the next signal value change for the selected signal.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Hotkey:</strong> Tab</td>
<td></td>
</tr>
<tr>
<td>![Find Previous Falling Edge icon]</td>
<td>Find Previous Falling Edge</td>
<td><strong>Menu:</strong> Edit &gt; Signal Search</td>
<td>Moves the active cursor to the previous falling edge for the selected signal.</td>
</tr>
</tbody>
</table>
### Table 3-21. Debug Toolbar Tab Buttons (cont.)

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Find Next Falling Edge" /></td>
<td>Find Next Falling Edge</td>
<td><strong>Menu:</strong> Edit &gt; Signal Search</td>
<td>Moves the active cursor to the next falling edge for the selected signal.</td>
</tr>
<tr>
<td><img src="image" alt="Find Previous Rising Edge" /></td>
<td>Find Previous Rising Edge</td>
<td><strong>Menu:</strong> Edit &gt; Signal Search</td>
<td>Moves the active cursor to the previous rising edge for the selected signal.</td>
</tr>
<tr>
<td><img src="image" alt="Find Next Rising Edge" /></td>
<td>Find Next Rising Edge</td>
<td><strong>Menu:</strong> Edit &gt; Signal Search</td>
<td>Moves the active cursor to the next rising edge for the selected signal.</td>
</tr>
<tr>
<td><img src="image" alt="Select Mode" /></td>
<td>Select Mode</td>
<td><strong>Menu:</strong> Dataflow &gt; Mouse Mode &gt; Select Mode</td>
<td>Set the left mouse button to select mode and middle mouse button to zoom mode.</td>
</tr>
<tr>
<td><img src="image" alt="Zoom Mode" /></td>
<td>Zoom Mode</td>
<td><strong>Menu:</strong> Dataflow &gt; Mouse Mode &gt; Zoom Mode</td>
<td>Set left mouse button to zoom mode and middle mouse button to pan mode.</td>
</tr>
<tr>
<td><img src="image" alt="Pan Mode" /></td>
<td>Pan Mode</td>
<td><strong>Menu:</strong> Dataflow &gt; Mouse Mode &gt; Pan Mode</td>
<td>Set left mouse button to pan mode and middle mouse button to zoom mode.</td>
</tr>
<tr>
<td><img src="image" alt="Two Cursor Mode" /></td>
<td>Two Cursor Mode</td>
<td><strong>Menu:</strong> Wave &gt; Mouse Mode &gt; Two Cursor</td>
<td>Sets two cursors in Wave window. First cursor moves with LMB, second cursor with MMB.</td>
</tr>
<tr>
<td><img src="image" alt="Stop Drawing" /></td>
<td>Stop Drawing</td>
<td>None</td>
<td>Halt any drawing currently happening in the window.</td>
</tr>
<tr>
<td><img src="image" alt="Show Drivers" /></td>
<td>Show Drivers</td>
<td>None</td>
<td>Display driver(s) of the selected signal, net, or register in the Dataflow and Source windows.</td>
</tr>
<tr>
<td><img src="image" alt="Show Drivers" /></td>
<td>Show Drivers (source only)</td>
<td>None</td>
<td>Display drivers only in the Source window.</td>
</tr>
<tr>
<td><img src="image" alt="Show Drivers" /></td>
<td></td>
<td></td>
<td>The source window is not shown if there are no drivers.</td>
</tr>
</tbody>
</table>
### Table 3-21. Debug Toolbar Tab Buttons (cont.)

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Show Readers" /></td>
<td>Show Readers</td>
<td>None</td>
<td>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.</td>
</tr>
<tr>
<td><img src="image" alt="Add Contributing Signals" /></td>
<td>Add Contributing Signals</td>
<td><strong>Menu</strong>: Add &gt; To Wave &gt; Contributing Signals</td>
<td>Creates a group labeled <strong>Contributors: &lt;name&gt;</strong>, where <code>&lt;name&gt;</code> is the name of the currently selected signal. This group contains the inputs to the process driving <code>&lt;name&gt;</code>.</td>
</tr>
<tr>
<td><img src="image" alt="Expanded Time Off" /></td>
<td>Expanded Time Off</td>
<td><strong>Menu</strong>: Wave &gt; Expanded Time &gt; Off</td>
<td>Turns off the expanded time display (default mode)</td>
</tr>
<tr>
<td><img src="image" alt="Expanded Time Deltas Mode" /></td>
<td>Expanded Time Deltas Mode</td>
<td><strong>Menu</strong>: Wave &gt; Expanded Time &gt; Deltas Mode</td>
<td>Displays delta time steps</td>
</tr>
<tr>
<td><img src="image" alt="Expanded Time Events Mode" /></td>
<td>Expanded Time Events Mode</td>
<td><strong>Menu</strong>: Wave &gt; Expanded Time &gt; Events Mode</td>
<td>Displays event time steps</td>
</tr>
<tr>
<td><img src="image" alt="Expand All Time" /></td>
<td>Expand All Time</td>
<td><strong>Menu</strong>: Wave &gt; Expanded Time &gt; Expand All</td>
<td>Expands simulation time over the entire simulation time range, from 0 to current time</td>
</tr>
<tr>
<td><img src="image" alt="Expand Time at Active Cursor" /></td>
<td>Expand Time at Active Cursor</td>
<td><strong>Menu</strong>: Wave &gt; Expanded Time &gt; Expand Cursor</td>
<td>Expands simulation time at the simulation time of the active cursor</td>
</tr>
<tr>
<td><img src="image" alt="Collapse All Time" /></td>
<td>Collapse All Time</td>
<td><strong>Menu</strong>: Wave &gt; Expanded Time &gt; Collapse All</td>
<td>Collapses simulation time over entire simulation time range</td>
</tr>
<tr>
<td><img src="image" alt="Collapse Time at Active Cursor" /></td>
<td>Collapse Time at Active Cursor</td>
<td><strong>Menu</strong>: Wave &gt; Expanded Time &gt; Collapse Cursor</td>
<td>Collapses simulation time at the simulation time of the active cursor</td>
</tr>
<tr>
<td><img src="image" alt="Find First Difference" /></td>
<td>Find First Difference</td>
<td>None</td>
<td>Find the first difference in a waveform comparison</td>
</tr>
</tbody>
</table>
### Toolbars

#### Toolbar Tabs

**Edit Toolbar Tab**

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

---

### Table 3-21. Debug Toolbar Tab Buttons (cont.)

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Find Previous Annotated Difference" /></td>
<td>Find Previous Annotated Difference</td>
<td>None</td>
<td>Find the previous annotated difference in a waveform comparison</td>
</tr>
<tr>
<td><img src="image" alt="Find Previous Difference" /></td>
<td>Find Previous Difference</td>
<td>None</td>
<td>Find the previous difference in a waveform comparison</td>
</tr>
<tr>
<td><img src="image" alt="Find Next Difference" /></td>
<td>Find Next Difference</td>
<td>None</td>
<td>Find the next difference in a waveform comparison</td>
</tr>
<tr>
<td><img src="image" alt="Find Annotated Difference" /></td>
<td>Find Next Annotated Difference</td>
<td>None</td>
<td>Find the next annotated difference in a waveform comparison</td>
</tr>
<tr>
<td><img src="image" alt="Find Last Difference" /></td>
<td>Find Last Difference</td>
<td>None</td>
<td>Find the last difference in a waveform comparison</td>
</tr>
<tr>
<td><img src="image" alt="Wave Search Box" /></td>
<td>Wave Search Box</td>
<td>Click on the box when in transition mode (Falling Edge, Rising Edge, or Any Transition) to cycle through these options.</td>
<td>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.</td>
</tr>
<tr>
<td><img src="image" alt="Search Options" /></td>
<td>Search Previous/Next</td>
<td>Previous: Shift+Enter Next: enter</td>
<td>Searches for the next occurrence of the string, either backward or forward in time, from the cursor.</td>
</tr>
</tbody>
</table>
| ![Search Options](image) | Search Options               | Menu: Edit > Signal Search | Dropdown button to:  
  - Change Mode: value, rising edge, falling edge, or any transition.  
  - Display the Wave Signal Search Dialog Box for advanced search behavior.  
  - Clear the value history. |
Toolbars

Figure 3-24. Edit Tab

Table 3-22. Edit Toolbar Tab Buttons

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>New File</td>
<td>Menu: File &gt; New &gt; Source</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Click and hold the button to open a dropdown menu with the following options:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• VHDL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Verilog</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• SystemVerilog</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Do</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Other</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Set Default Action</td>
</tr>
<tr>
<td></td>
<td>Insert Pulse</td>
<td>Menu: Wave &gt;</td>
<td>Insert a transition at the selected time.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Wave Editor &gt; Insert Pulse</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Command: wave edit insert_pulse</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Delete Edge</td>
<td>Menu: Wave &gt;</td>
<td>Delete the selected transition.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Wave Editor &gt; Delete Edge</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Command: wave edit delete</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Invert</td>
<td>Menu: Wave &gt;</td>
<td>Invert the selected section of the waveform.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Wave Editor &gt; Invert</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Command: wave edit invert</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mirror</td>
<td>Menu: Wave &gt;</td>
<td>Mirror the selected section of the waveform.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Wave Editor &gt; Mirror</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Command: wave edit mirror</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Change Value</td>
<td>Menu: Wave &gt;</td>
<td>Change the value of the selected section of the waveform.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Wave Editor &gt; Value</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Command: wave edit change_value</td>
<td></td>
</tr>
</tbody>
</table>
Toolbars

Home Toolbar Tab

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

Figure 3-25. Home Tab

Table 3-23. Home Toolbar Tab Buttons

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Open</td>
<td>Menu: File &gt; Open</td>
<td>Opens the Open File dialog</td>
</tr>
<tr>
<td></td>
<td>Save</td>
<td>Menu: File &gt; Save</td>
<td>Saves the contents of the active window or Saves the current wave window display and signal preferences to a DO file script.</td>
</tr>
<tr>
<td>Button Name</td>
<td>Shortcuts</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-----------------------------</td>
<td>------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Reload</td>
<td><strong>Command</strong>: Dataset Restart</td>
<td>Reload the current dataset.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Menu</strong>: File &gt; Datasets</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Print</td>
<td><strong>Menu</strong>: File &gt; Print</td>
<td>Opens the Print dialog box.</td>
<td></td>
</tr>
<tr>
<td>Cut</td>
<td><strong>Menu</strong>: Edit &gt; Cut</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Hotkey</strong>: Ctrl+x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Copy</td>
<td><strong>Menu</strong>: Edit &gt; Copy</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Hotkey</strong>: Ctrl+c</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paste</td>
<td><strong>Menu</strong>: Edit &gt; Paste</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Hotkey</strong>: Ctrl+v</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Undo</td>
<td><strong>Menu</strong>: Edit &gt; Undo</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Hotkey</strong>: Ctrl+z</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Redo</td>
<td><strong>Menu</strong>: Edit &gt; Redo</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Hotkey</strong>: Ctrl+y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add Selected to Window</td>
<td><strong>Menu</strong>: Add &gt; to Wave</td>
<td>Clicking adds selected objects to the Wave window. Refer to “Add Selected to Window Button” for more information about the dropdown menu selections.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Hotkey</strong>: Ctrl+w</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Find</td>
<td><strong>Menu</strong>: Edit &gt; Find</td>
<td>Opens the Find dialog box.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Hotkey</strong>: Ctrl+f (Windows) or Ctrl+s (UNIX)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Environment Up</td>
<td><strong>Command</strong>: env..</td>
<td>Changes your environment up one level of hierarchy.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Menu</strong>: File &gt; Environment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Environment Back</td>
<td><strong>Command</strong>: env -back</td>
<td>Change your environment to its previous location.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Menu</strong>: File &gt; Environment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Environment Forward</td>
<td><strong>Command</strong>: env -forward</td>
<td>Change your environment forward to a previously selected environment.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Menu</strong>: File &gt; Environment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Button</td>
<td>Name</td>
<td>Shortcuts</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>----------------</td>
<td>-----------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td>Add Bookmark</td>
<td>Command Wave window only: bookmark add wave</td>
<td>Clicking this button bookmarks the current view of the active window.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Menu Wave window only: Add &gt; To Wave &gt; Bookmark</td>
<td>Click and hold the button to open a drop down menu with the following options:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Add Current View</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Add Custom ...</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Set Default Action</td>
</tr>
<tr>
<td><img src="image2.png" alt="Image" /></td>
<td>Delete All Bookmarks</td>
<td>CommandWave window only: bookmark delete wave-all</td>
<td>Removes all bookmarks, after prompting for your confirmation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Click and hold the button to open a drop down menu with the following options:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Active Window</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• All Windows</td>
</tr>
<tr>
<td><img src="image3.png" alt="Image" /></td>
<td>Manage Bookmarks</td>
<td>None</td>
<td>Displays the Manage Bookmarks dialog box.</td>
</tr>
<tr>
<td><img src="image4.png" alt="Image" /></td>
<td>Reload from File</td>
<td>None</td>
<td>Reloads bookmarks from the bookmarks.do file.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Set Default Action</td>
</tr>
<tr>
<td><img src="image5.png" alt="Image" /></td>
<td>Jump to Bookmark</td>
<td>CommandWave window only: bookmark goto wave</td>
<td>Displays bookmarks grouped by window. Select the bookmark you want to display.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;name&gt;</td>
<td></td>
</tr>
<tr>
<td><img src="image6.png" alt="Image" /></td>
<td>Search Documentation</td>
<td>None</td>
<td>A text entry box for your search string.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><img src="image7.png" alt="Image" /></td>
<td>Search Documentation</td>
<td>Hotkey: Enter</td>
<td>Activates the search for the term you entered into the text entry box.</td>
</tr>
</tbody>
</table>

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.
Toolbars

**Toolbar Tabs**

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

**Figure 3-26. Layout Tab**

![Layout Tab](image)

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="ColumnLayout" /></td>
<td>Column Layout</td>
<td><strong>Menu:</strong> Column Layout</td>
<td>A dropdown box that allows you to specify the column layout for the active window.</td>
</tr>
<tr>
<td><img src="image" alt="Simulate" /></td>
<td>Change Layout</td>
<td><strong>Menu:</strong> Layout &lt;layoutName&gt;</td>
<td>A dropdown box that allows you to select a GUI layout. - NoDesign - Simulate</td>
</tr>
</tbody>
</table>

**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 3-27. Schematic Tab**

![Schematic Tab](image)

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Trace Input" /></td>
<td>Trace Input Net to Event</td>
<td><strong>Menu:</strong> Tools &gt; Trace &gt; Trace next event</td>
<td>Move the next event cursor to the next input event driving the selected output.</td>
</tr>
</tbody>
</table>
### Table 3-25. Schematic Toolbar Tab Buttons (cont.)

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Trace Set</strong></td>
<td><strong>Menu: Tools &gt; Trace &gt; Trace event set</strong></td>
<td>Jump to the source of the selected input event.</td>
</tr>
<tr>
<td></td>
<td><strong>Trace Reset</strong></td>
<td><strong>Menu: Tools &gt; Trace &gt; Trace event reset</strong></td>
<td>Return the next event cursor to the selected output.</td>
</tr>
<tr>
<td></td>
<td><strong>Trace Net to Driver of X</strong></td>
<td><strong>Menu: Tools &gt; Trace &gt; TraceX</strong></td>
<td>Step back to the last driver of an unknown value.</td>
</tr>
<tr>
<td></td>
<td><strong>Expand Net to all Drivers</strong></td>
<td>None</td>
<td>Display driver(s) of the selected signal, net, or register.</td>
</tr>
<tr>
<td></td>
<td><strong>Expand Net to all Drivers and Readers</strong></td>
<td>None</td>
<td>Display driver(s) and reader(s) of the selected signal, net, or register.</td>
</tr>
<tr>
<td></td>
<td><strong>Expand Net to all Readers</strong></td>
<td>None</td>
<td>Display reader(s) of the selected signal, net, or register.</td>
</tr>
<tr>
<td></td>
<td><strong>Remove All Highlights</strong></td>
<td><strong>Menu: Dataflow &gt; Remove Highlight</strong></td>
<td>Clear the green highlighting identifying the path you’ve traversed through the design.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or <strong>Schematic &gt; Edit &gt; Remove Highlight</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Delete Content</strong></td>
<td><strong>Menu: Dataflow &gt; Delete</strong></td>
<td>Delete the selected signal.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or <strong>Schematic &gt; Edit &gt; Delete</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>or <strong>Schematic &gt; Edit &gt; Delete All</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Regenerate</strong></td>
<td><strong>Menu: Dataflow &gt; Regenerate</strong></td>
<td>Redraws the current schematic view to better take advantage of the available space. For example, after adding or removing elements.</td>
</tr>
</tbody>
</table>
Simulate Toolbar Tab

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

**Figure 3-28. Simulate Tab**

![Simulate Tab](image)

Table 3-25. Schematic Toolbar Tab Buttons (cont.)

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Show Wave" /></td>
<td>Show Wave</td>
<td>Menu: Dataflow &gt; Show Wave</td>
<td>Display the embedded wave viewer pane.</td>
</tr>
</tbody>
</table>

Table 3-26. Simulate Toolbar Tab Buttons

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Source Hyperlinking" /></td>
<td>Source Hyperlinking</td>
<td>None</td>
<td>Toggles display of hyperlinks in design source files.</td>
</tr>
<tr>
<td><img src="image" alt="Step Into" /></td>
<td>Step Into</td>
<td>Command: step Menu: Simulate &gt; Run &gt; Step</td>
<td>Step the current simulation to the next statement.</td>
</tr>
<tr>
<td><img src="image" alt="Step Over" /></td>
<td>Step Over</td>
<td>Command: step -over Menu: Simulate &gt; Run &gt; Step -Over</td>
<td>Execute HDL statements, treating them as simple statements instead of entered and traced line by line.</td>
</tr>
<tr>
<td><img src="image" alt="Step Out" /></td>
<td>Step Out</td>
<td>Command: step -out</td>
<td>Step the current simulation out of the current function or procedure.</td>
</tr>
<tr>
<td><img src="image" alt="Step Into Current" /></td>
<td>Step Into Current</td>
<td>Command: step -current</td>
<td>Step the simulation into the current instance, process, or thread.</td>
</tr>
<tr>
<td><img src="image" alt="Step Over Current" /></td>
<td>Step Over Current</td>
<td>Command: step -over -current</td>
<td>Step the simulation over the current instance, process, or thread.</td>
</tr>
<tr>
<td><img src="image" alt="Step Out Current" /></td>
<td>Step Out Current</td>
<td>Command: step -out -current</td>
<td>Step the simulation out of the current instance, process, or thread.</td>
</tr>
</tbody>
</table>
### Windows With Dedicated Toolbars

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

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Shortcuts</th>
<th>Description</th>
</tr>
</thead>
</table>
| ![Restart](image) | 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](image) | Run Length | **Command:** run  
**Menu:** Simulate > Run > Run `default_run_length` | Specify the run length for the current simulation. |
| ![Run](image) | Run | **Command:** run  
**Menu:** Simulate > Run > Run `default_run_length` | Run the current simulation for the specified run length. |
| ![Continue Run](image) | 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](image) | 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](image) | Simulate | **Command:** vsim  
**Menu:** Simulate > Start Simulation | Opens the Start Simulation dialog box. |
| ![Break](image) | 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](image) | Stop -sync | None | Stop simulation the next time time/delta is advanced. |
| ![Edit Breakpoints](image) | Edit Breakpoints | **Menu:** Tools > Breakpoint | Enable breakpoint editing, loading, and saving. |
Click the Show/Hide Toolbar button in the window title bar to display the window specific buttons (Figure 3-29). You can add buttons to any currently open window. Refer to Customizing Button and Tab Location for more information.

**Figure 3-29. Window Specific Buttons**

The following windows have default buttons or widgets:

- FSM Viewer Window
- Memory Data Window
- Objects Window
- Processes Window
- Source 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 3-27):
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.

---

**Table 3-27. Toolbar Tab Popup Menu**

<table>
<thead>
<tr>
<th>Popup Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hide Tabs</td>
<td>Hides all tabs except for the currently active tab.</td>
</tr>
<tr>
<td>Edit Toolbars</td>
<td>Opens the <strong>Toolbar Tab Widget Toolbox</strong> and locks the GUI into edit mode.</td>
</tr>
<tr>
<td>Remove &lt;name&gt; Tab</td>
<td>Removes the tab that is under the cursor.</td>
</tr>
<tr>
<td>Add Tab</td>
<td>Opens the Add/Create Tab dialog box. Use this dialog box to create a new user-defined tab or restore a previously defined tab.</td>
</tr>
<tr>
<td>Reset Tabs</td>
<td>Resets GUI to the default tabs and order of tabs. Removes user-defined tabs.</td>
</tr>
<tr>
<td>Reset &lt;name&gt; Toolbar</td>
<td>Resets the selected tab to the default buttons and widgets.</td>
</tr>
<tr>
<td>Reset All Toolbars</td>
<td>Resets all toolbar tabs to their default buttons and widgets. User-defined toolbar tabs default to an empty toolbar.</td>
</tr>
</tbody>
</table>
In addition you can change the order of the Toolbar Tabs. Open the **Toolbar Tab Widget Toolbox** to make modifications to Toolbar Tabs by right-clicking in the toolbar area of the main GUI and selecting **Edit Toolbars**.

### 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 3-30. 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 doesn’t 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>
<thead>
<tr>
<th>Default Toolbar</th>
<th>Toolbar Tab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bookmarks Toolbar</td>
<td>Home Toolbar Tab</td>
</tr>
<tr>
<td>Help Toolbar</td>
<td></td>
</tr>
<tr>
<td>Standard Toolbar</td>
<td></td>
</tr>
<tr>
<td>Wave Edit Toolbar</td>
<td>Edit Toolbar Tab</td>
</tr>
<tr>
<td>Compile Toolbar</td>
<td>Compile Toolbar Tab</td>
</tr>
<tr>
<td>Process Toolbar</td>
<td></td>
</tr>
<tr>
<td>Simulate Toolbar</td>
<td></td>
</tr>
<tr>
<td>Step Toolbar</td>
<td>Simulate Toolbar Tab</td>
</tr>
<tr>
<td>Mode Toolbar</td>
<td>Debug Toolbar Tab</td>
</tr>
<tr>
<td>Source Toolbar</td>
<td></td>
</tr>
<tr>
<td>Wave Toolbar</td>
<td></td>
</tr>
<tr>
<td>Wave Cursor Toolbar</td>
<td></td>
</tr>
<tr>
<td>Wave Expand Time Toolbar</td>
<td></td>
</tr>
<tr>
<td>Zoom Toolbar</td>
<td></td>
</tr>
<tr>
<td>Dataflow Toolbar</td>
<td>Schematic and Dataflow Toolbar Tab</td>
</tr>
<tr>
<td>Schematic Toolbar</td>
<td></td>
</tr>
<tr>
<td>Layout Toolbar</td>
<td>Layout Toolbar Tab</td>
</tr>
<tr>
<td>Memory Toolbar</td>
<td>Memory Data Window</td>
</tr>
<tr>
<td>Objectfilter Toolbar</td>
<td>Object Window Toolbar</td>
</tr>
<tr>
<td>Process Toolbar</td>
<td>Process Window Toolbar</td>
</tr>
<tr>
<td>Bookmarks Toolbar</td>
<td>Source Window Toolbar</td>
</tr>
</tbody>
</table>
## Chapter 4
Window Reference

The following table summarizes all of the available windows.

### Table 4-1. GUI Windows

<table>
<thead>
<tr>
<th>Window</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call Stack Window</td>
<td>displays the current call stack, allowing you to debug your design by analyzing the depth of function calls</td>
</tr>
<tr>
<td>Class Graph Window</td>
<td>displays interactive relationships of SystemVerilog classes in graphical form</td>
</tr>
<tr>
<td>Class Instances Window</td>
<td>displays class instances</td>
</tr>
<tr>
<td>Class Tree Window</td>
<td>displays interactive relationships of SystemVerilog classes in tabular form</td>
</tr>
<tr>
<td>Dataflow Window</td>
<td>displays &quot;physical&quot; connectivity and lets you trace events (causality)</td>
</tr>
<tr>
<td>Files Window</td>
<td>displays the source files and their locations for the loaded simulation</td>
</tr>
<tr>
<td>Library Window</td>
<td>lists design libraries and compiled design units</td>
</tr>
<tr>
<td>List Window</td>
<td>shows waveform data in a tabular format</td>
</tr>
<tr>
<td>Locals Window</td>
<td>displays data objects that are immediately visible at the current execution point of the selected process</td>
</tr>
<tr>
<td>Memory List Window</td>
<td>windows that show memories and their contents</td>
</tr>
<tr>
<td>Memory Data Window</td>
<td></td>
</tr>
<tr>
<td>Message Viewer Window</td>
<td>allows easy access, organization, and analysis of Note, Warning, Errors or other messages written to transcript during simulation</td>
</tr>
<tr>
<td>Objects Window</td>
<td>displays all declared data objects in the current scope</td>
</tr>
<tr>
<td>Processes Window</td>
<td>displays all processes that are scheduled to run during the current simulation cycle</td>
</tr>
<tr>
<td>Projects</td>
<td>provides access to information about Projects</td>
</tr>
<tr>
<td>Source Window</td>
<td>a text editor for viewing and editing files, such as Verilog, VHDL, and DO files</td>
</tr>
</tbody>
</table>
### Table 4-1. GUI Windows (cont.)

<table>
<thead>
<tr>
<th>Window</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Structure Window</strong></td>
<td>displays hierarchical view of active simulation. Name of window is either “sim” or “&lt;dataset_name&gt;”</td>
</tr>
<tr>
<td><strong>Transcript Window</strong></td>
<td>keeps a running history of commands and messages and provides a command-line interface</td>
</tr>
<tr>
<td><strong>Watch Window</strong></td>
<td>displays signal or variable values at the current simulation time</td>
</tr>
<tr>
<td><strong>Wave Window</strong></td>
<td>displays waveforms</td>
</tr>
</tbody>
</table>
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
- C/C++ functions

**Figure 4-1. Call Stack Window**

![](image)

**Fields**

**Table 4-2. Call Stack Window Columns**

<table>
<thead>
<tr>
<th>Column Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>indicates the depth of the function call, with the most recent at the top.</td>
</tr>
<tr>
<td>In</td>
<td>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.</td>
</tr>
<tr>
<td>Line</td>
<td>indicates the line number containing the function call.</td>
</tr>
<tr>
<td>File</td>
<td>indicates the location of the file containing the function call.</td>
</tr>
</tbody>
</table>

**Usage Notes**

**Call Stack Window Tasks**

This window allows you to perform the following actions:
• Double-click the line of any function call:
  o Displays the local variables at that level in the **Locals Window**.
  o Displays the corresponding source code in the **Source Window**.

**Related Commands of the Call Stack Window**

<table>
<thead>
<tr>
<th>Command Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stack down</td>
<td>This command moves down the call stack.</td>
</tr>
<tr>
<td>stack frame</td>
<td>This command selects the specified call frame.</td>
</tr>
<tr>
<td>stack level</td>
<td>This command reports the current call frame number.</td>
</tr>
<tr>
<td>stack tb</td>
<td>This command is an alias for the <strong>tb</strong> command.</td>
</tr>
<tr>
<td>stack up</td>
<td>This command moves up the call stack.</td>
</tr>
</tbody>
</table>
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 4-2. Class Graph Window](image)
Fields

### Table 4-4. Class Graph Window Popup Menu

<table>
<thead>
<tr>
<th>Popup Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter</td>
<td>Controls the display of methods and properties from the class boxes.</td>
</tr>
<tr>
<td>Zoom Full</td>
<td></td>
</tr>
<tr>
<td>View Entire Design</td>
<td>Reloads the view to show the class hierarchy of the complete design.</td>
</tr>
<tr>
<td>Print to Postscript</td>
<td></td>
</tr>
<tr>
<td>Organize by Base/Extended Class</td>
<td>Reorganizes the window so that the base or extended (default) classes are at the top of the hierarchy.</td>
</tr>
</tbody>
</table>

Usage Notes

Class Graph Window Tasks

This section describes tasks for using the Cover Directives window.

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 4-3. Class Instances Window
## Fields

### Table 4-5. Class Instances Window Popup Menu

<table>
<thead>
<tr>
<th>Popup Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Declaration</td>
<td>Highlights the line of code where the type of the instance is declared, opening the source file if necessary.</td>
</tr>
<tr>
<td>Add Wave</td>
<td>Adds the selected class instance to the Wave window.</td>
</tr>
<tr>
<td>Add to</td>
<td>Allows you to log the selected class instance, or add it to the Wave or List windows.</td>
</tr>
</tbody>
</table>

## 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 4-4. Class Tree Window
## Fields

### Table 4-6. Class Tree Window Icons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="icon" alt="Class Icon" /></td>
<td>Class</td>
</tr>
<tr>
<td><img src="icon" alt="Parameterized Class Icon" /></td>
<td>Parameterized Class</td>
</tr>
<tr>
<td><img src="icon" alt="Function Icon" /></td>
<td>Function</td>
</tr>
<tr>
<td><img src="icon" alt="Task Icon" /></td>
<td>Task</td>
</tr>
<tr>
<td><img src="icon" alt="Variable Icon" /></td>
<td>Variable</td>
</tr>
<tr>
<td><img src="icon" alt="Virtual Interface Icon" /></td>
<td>Virtual Interface</td>
</tr>
<tr>
<td><img src="icon" alt="Covergroup Icon" /></td>
<td>Covergroup</td>
</tr>
<tr>
<td><img src="icon" alt="Structure Icon" /></td>
<td>Structure</td>
</tr>
</tbody>
</table>

### Table 4-7. Class Tree Window Columns

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>The name of the item</td>
</tr>
<tr>
<td>Type</td>
<td>The type of item</td>
</tr>
<tr>
<td>File</td>
<td>The source location of the item</td>
</tr>
<tr>
<td>Unique Id</td>
<td>The internal name of the parameterized class (only available with parameterized classes)</td>
</tr>
<tr>
<td>Scope</td>
<td>The scope of the covergroup (only available with covergroups)</td>
</tr>
</tbody>
</table>

### Table 4-8. Class Tree Window Popup Menu

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

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. See **Symbol Mapping** for details.

---

**Note**

This version of ModelSim has limited Dataflow functionality resulting in many of the features described in this chapter operating differently. The window will show only one process and its attached signals or one signal and its attached processes, as displayed in **Figure 4-5**.
Window Reference

Dataflow Window Tasks

Figure 4-5. Dataflow Window - ModelSim

Fields

- Dataflow view

Dataflow Window Tasks

This section describes tasks for using the Dataflow window.

You can interact with the Dataflow in one of three different Mouse modes, which you can change through the DataFlow menu or the Zoom Toolbar:

- **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
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:

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Do the following:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select a single object</td>
<td>Single click</td>
</tr>
<tr>
<td>Select multiple objects</td>
<td>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.</td>
</tr>
</tbody>
</table>

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.
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:
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.

Refer to the section “Exploring Designs with the Embedded Wave Viewer” for more information.
Figure 4-6. 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 4-7. Files Window](image)

Right-click anywhere in the window to display the popup menu and select one of the options in Table 4-21.

The Files menu (see Table 4-22) becomes available in the **Main** menu when the Files window is active.
Fields

Table 4-9. Files Window Columns

<table>
<thead>
<tr>
<th>Column Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the file</td>
</tr>
<tr>
<td>Specified Path</td>
<td>The location of the file as specified in the design files.</td>
</tr>
<tr>
<td>Full Path</td>
<td>The full-path location of the design files.</td>
</tr>
<tr>
<td>Type</td>
<td>The file type.</td>
</tr>
</tbody>
</table>

Table 4-10. Files Window Popup Menu

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Source</td>
<td>Opens the selected file in a Source window</td>
</tr>
</tbody>
</table>
| 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.                           |
| Properties               | Displays the File Properties dialog box, containing information about the selected file. |

Table 4-11. Files Menu

<table>
<thead>
<tr>
<th>Files Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Source</td>
<td>Opens the selected file in a Source window</td>
</tr>
</tbody>
</table>
| 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. |
Library Window

To access:

- View > Library
- view library command

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

Description

![Library Window](image)

Figure 4-8. Library Window

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>work</td>
<td>Library</td>
<td>C:/questasim_5.6/examples/tutorials/...</td>
</tr>
<tr>
<td>and2</td>
<td>Module</td>
<td>C:/questasim_5.6/examples/tutorials/...</td>
</tr>
<tr>
<td>cache</td>
<td>Module</td>
<td>C:/questasim_5.6/examples/tutorials/...</td>
</tr>
<tr>
<td>cache_set</td>
<td>Module</td>
<td>C:/questasim_5.6/examples/tutorials/...</td>
</tr>
<tr>
<td>memory</td>
<td>Module</td>
<td>C:/questasim_5.6/examples/tutorials/...</td>
</tr>
<tr>
<td>or2</td>
<td>Module</td>
<td>C:/questasim_5.6/examples/tutorials/...</td>
</tr>
<tr>
<td>proc</td>
<td>Module</td>
<td>C:/questasim_5.6/examples/tutorials/...</td>
</tr>
<tr>
<td>top</td>
<td>Module</td>
<td>C:/questasim_5.6/examples/tutorials/...</td>
</tr>
<tr>
<td>tcp_opt</td>
<td>Optimized</td>
<td>C:/questasim_5.6/examples/tutorials/...</td>
</tr>
<tr>
<td>v_and2</td>
<td>Module</td>
<td>C:/questasim_5.6/examples/tutorials/...</td>
</tr>
<tr>
<td>floatfxlib</td>
<td>Library</td>
<td>$MODEL_Tech/../floatfxlib</td>
</tr>
</tbody>
</table>

Fields

This section describes GUI elements specific to this Window.

<table>
<thead>
<tr>
<th>Table 4-12. Library Window Columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column Title</td>
</tr>
<tr>
<td>-------------</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Path</td>
</tr>
<tr>
<td>Type</td>
</tr>
</tbody>
</table>

Right-click anywhere in the window to display the popup menu and select one of the following options:

<table>
<thead>
<tr>
<th>Table 4-13. Library Window Popup Menu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Popup Menu Item</td>
</tr>
<tr>
<td>Simulate</td>
</tr>
</tbody>
</table>
## Table 4-13. Library Window Popup Menu (cont.)

<table>
<thead>
<tr>
<th>Popup Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Edit</strong></td>
<td>Opens the selected file in your editor window.</td>
</tr>
<tr>
<td><strong>Refresh</strong></td>
<td>Reloads the contents of the window.</td>
</tr>
<tr>
<td><strong>Recompile</strong></td>
<td>Compiles the selected file.</td>
</tr>
<tr>
<td><strong>Update</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Create Wave</strong></td>
<td>Runs the wave create command for any ports in the selected design unit.</td>
</tr>
<tr>
<td><strong>Delete</strong></td>
<td>Removes a design unit from the library or runs the vdel command on a selected library.</td>
</tr>
<tr>
<td><strong>Copy</strong></td>
<td>Copies the directory location of libraries or the library location of design units within the library.</td>
</tr>
<tr>
<td><strong>New</strong></td>
<td>Allows you to create a new library with the Create a New Library dialog box.</td>
</tr>
<tr>
<td><strong>Properties</strong></td>
<td>Displays information about the selected library or design unit.</td>
</tr>
</tbody>
</table>
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. See Delta Delays 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 4-9. Tabular Format of the List Window](image)

You can view the following object types in the List window:

- **VHDL** — Signals, aliases, process variables, and shared variables
- **Verilog** — Nets, registers, and variables
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 of more than 10 characters is replaced with the following:

\[\text{too narrow}\]

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.

Right-click in the right-hand pane to display the popup menu and select an option in Table 4-33. Several menu items (See Table ??) are available when the List window is active:

### Fields

**Table 4-14. List Window Popup Menu**

<table>
<thead>
<tr>
<th>Popup Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examine</td>
<td>Displays the value of the signal over which you used the right mouse button, at the time selected with the Active Marker</td>
</tr>
<tr>
<td>Add Marker</td>
<td>Adds a marker at the location of the Active Marker</td>
</tr>
<tr>
<td>Delete Marker</td>
<td>Deletes the closest marker to your mouse location</td>
</tr>
</tbody>
</table>

#### 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>
```
Window Reference
List Window Tasks

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

```plaintext
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 .................................................. 114
Selecting Multiple Signals ....................................................... 115
Setting Time Markers in the List Window .................................. 115
Searching in the List Window ..................................................... 118
Searching for Values or Transitions ......................................... 119
Using the Expression Builder for Expression Searches ............... 120
Saving an Expression to a Tcl Variable ................................. 122
Formatting the List Window ...................................................... 123
Saving the Window Format ....................................................... 125
Combining Signals into Buses ................................................... 126
Configuring New Line Triggering .............................................. 127

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:

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Do the following...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use menu to add data</td>
<td>Right-click signals and objects in the Objects window or the Structure window and select Add &gt; to List.</td>
</tr>
<tr>
<td>Use the command line to add data</td>
<td>Use the add list command.</td>
</tr>
</tbody>
</table>
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.

This process creates virtual signals. For more information, refer to the section Virtual Signals.

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.
Working with Markers

The table below summarizes actions you can take with markers.

**Table 4-15. Actions for Time Markers**

<table>
<thead>
<tr>
<th>Action</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add marker</td>
<td>Select a line and then choose <strong>List &gt; Add Marker</strong></td>
</tr>
<tr>
<td>Delete marker</td>
<td>Select a tagged line and then choose <strong>List &gt; Delete Marker</strong></td>
</tr>
<tr>
<td>Goto marker</td>
<td>Choose <strong>View &gt; Goto &gt; &lt;time&gt;</strong> (only available when undocked)</td>
</tr>
</tbody>
</table>

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 4-12** 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 4-12. List Window After configure list -delta none Option is Used**

![Image of List Window](image)

**Figure 4-13** 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 4-13. List Window After configure list -delta collapse Option is Used**

![Image of List Window](image)

**Figure 4-14** 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 4-14. List Window After configure list -delta all Option is Used**

![Image of List Window](image)
Searching in the List Window

Use the List window to locate objects.

Procedure

Search the List window using one of the following methods:
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.
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** 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 ModelSim.

### 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 4-17.

Figure 4-17. Expression Builder Dialog Box

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. In addition, you can use the **Selected Signal** button to create an expression from signals you select from the List window.

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 4-18.
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.

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:
  ```tcl
  set foo
  ```
- Put `$foo` in the Expression: entry box for the Search for Expression selection.
- Issue a `searchlog` command using `foo`:
  ```tcl
  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 4-19. Modifying List Window Display Properties](image)

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 4-20** shows the list of radix types you can select in this dialog box.
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 (with symbolic values).
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.

### 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:
Window Reference
List Window Tasks

- 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 (see Saving and Restoring Breakpoints) when invoked with the do command in subsequent simulation runs. The syntax is:

```
write format restart <filename>
```

If the ShutdownFile modelsim.ini variable is set to this .do filename, it will call the write format restart command upon exit.

**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:

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Do the following...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use the GUI</td>
<td>Select two or more signals in the Wave or List window and then choose List &gt; 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.</td>
</tr>
<tr>
<td>Use the Command Line</td>
<td>Use the virtual signal command at the Main window command prompt.</td>
</tr>
</tbody>
</table>
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 ModelSim 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.

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:
The following table summarizes the triggering options:

**Table 4-16. Triggering Options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deltas</td>
<td>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.</td>
</tr>
<tr>
<td>Strobe trigger</td>
<td>Specify an interval at which you want to trigger data display</td>
</tr>
<tr>
<td>Trigger gating</td>
<td>Use a gating expression to control triggering; see Using Gating Expressions to Control Triggering for more details.</td>
</tr>
</tbody>
</table>
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 ModelSim 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.
5. Click the **Selected Signal** button to open the “Select Signal for Expression” dialog box.

**Figure 4-25. 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 isn't 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:

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

See the configure command for more 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:

```plaintext
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.
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

![Locals Window](Figure 4-26. Locals Window)

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

![Change Selected Variable Dialog Box](Figure 4-27. Change Selected Variable Dialog Box)
**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.

Right-click anywhere in the Locals window to open a popup menu (See Table 4-37).

**Fields**

**Table 4-17. Locals Window Columns**

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>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</td>
</tr>
<tr>
<td>Value</td>
<td>lists the current value(s) associated with each name</td>
</tr>
</tbody>
</table>

**Table 4-18. Locals Window Popup Menu**

<table>
<thead>
<tr>
<th>Popup Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Declaration</td>
<td>Displays, in the Source window, the declaration of the object</td>
</tr>
<tr>
<td>Add</td>
<td>Adds the selected object(s) to the specified window</td>
</tr>
<tr>
<td>Copy</td>
<td>Copies selected item to clipboard</td>
</tr>
<tr>
<td>Find</td>
<td>Opens the Find toolbar at the bottom of the window</td>
</tr>
<tr>
<td>Expand/Collapse</td>
<td>Expands or collapses data in the window</td>
</tr>
<tr>
<td>Global Signal Radix</td>
<td>Sets radix for selected signal(s) in all windows</td>
</tr>
<tr>
<td>Change</td>
<td>Displays the Usage Notes, which allows you to alter the value of the object</td>
</tr>
</tbody>
</table>

**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.
Window Reference

Locals Window

- 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

This section describes GUI elements specific to this Window.

Right-click in the window to display the popup menu and select one of the options in Table 4-38. The Memory Data menu (see Table 4-40) becomes available in the Main menu when the Memory Data window is active.

Fields

Table 4-19. Memory Data Popup Menu — Address Pane

<table>
<thead>
<tr>
<th>Popup Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goto</td>
<td>Allows you to go to a specific address</td>
</tr>
<tr>
<td>Split Screen</td>
<td>Splits the Memory Data window to allow you to view different parts of the memory simultaneously.</td>
</tr>
</tbody>
</table>
### Table 4-19. Memory Data Popup Menu — Address Pane (cont.)

<table>
<thead>
<tr>
<th>Popup Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>Allows you to set various properties for the Memory Data window.</td>
</tr>
<tr>
<td>Close Instance</td>
<td>Closes the active Memory Data window.</td>
</tr>
<tr>
<td>Close All</td>
<td>Closes all Memory Data windows.</td>
</tr>
</tbody>
</table>

### Table 4-20. Memory Data Popup Menu — Data Pane

<table>
<thead>
<tr>
<th>Popup Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit</td>
<td>Allows you to edit the value of the selected data.</td>
</tr>
<tr>
<td>Change</td>
<td>Allows you to change data within the memory through the use of the Change Memory dialog box.</td>
</tr>
<tr>
<td>Import Data Patterns</td>
<td>Allows you to import data patterns into the selected memory through the Import Memory dialog box.</td>
</tr>
<tr>
<td>Export Data Patterns</td>
<td>Allows you to export data patterns from the selected memory through the Export Memory dialog box.</td>
</tr>
<tr>
<td>Split Screen</td>
<td>Refer to items in the Memory Data Popup Menu — Address Pane</td>
</tr>
<tr>
<td>Properties</td>
<td></td>
</tr>
<tr>
<td>Close Instance</td>
<td></td>
</tr>
<tr>
<td>Close All</td>
<td></td>
</tr>
</tbody>
</table>

### Table 4-21. Memory Data Menu

<table>
<thead>
<tr>
<th>Popup Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Declaration</td>
<td>Opens a Source window to the file and line number where the memory is declared.</td>
</tr>
<tr>
<td>Compare Contents</td>
<td>Allows you to compare the selected memory against another memory in the design or an external file.</td>
</tr>
<tr>
<td>Import Data Patterns</td>
<td>Refer to items in the Memory Data Popup Menu — Data Pane</td>
</tr>
<tr>
<td>Export Data Patterns</td>
<td></td>
</tr>
<tr>
<td>Expand Packed Memories</td>
<td>Toggle the expansion of packed memories.</td>
</tr>
<tr>
<td>Identify Memories Within Cells</td>
<td>Toggle the identification of memories within Verilog cells.</td>
</tr>
<tr>
<td>Show VHDL String as Memory</td>
<td>Toggle the identification of VHDL strings as memories.</td>
</tr>
</tbody>
</table>
Usage Notes
This section describes tasks for using the Memory Data window.

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 4-29. 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 on 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 4-22. Memory Identification — ModelSim

<table>
<thead>
<tr>
<th>Element Kind&lt;sup&gt;1&lt;/sup&gt;</th>
<th>VHDL</th>
<th>Verilog/SystemVerilog</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• enum&lt;sup&gt;2&lt;/sup&gt;</td>
<td>any integral type (that is, integer_type):</td>
</tr>
<tr>
<td></td>
<td>• bit_vector</td>
<td>• shortint</td>
</tr>
<tr>
<td></td>
<td>• floating point type</td>
<td>• int</td>
</tr>
<tr>
<td></td>
<td>• std_logic_vector</td>
<td>• longint</td>
</tr>
<tr>
<td></td>
<td>• std_ulogic_vector</td>
<td>• byte</td>
</tr>
<tr>
<td></td>
<td>• integer type</td>
<td>• bit (2 state)</td>
</tr>
<tr>
<td></td>
<td>• floating point type</td>
<td>• logic</td>
</tr>
<tr>
<td></td>
<td>• std_logic_vector</td>
<td>• reg</td>
</tr>
<tr>
<td></td>
<td>• std_ulogic_vector</td>
<td>• integer</td>
</tr>
<tr>
<td></td>
<td>• integer type</td>
<td>• time (4 state)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• packed_struct/packed_union (2 state)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• packed_struct/packed_union (4 state)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• packed_array (single-Dim, multi-D, 2 state and 4 state)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• enum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• string</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scope: Recognizable in</th>
<th>VHDL</th>
<th>Verilog/SystemVerilog</th>
</tr>
</thead>
<tbody>
<tr>
<td>• architecture</td>
<td>• module</td>
<td></td>
</tr>
<tr>
<td>• process</td>
<td>• interface</td>
<td></td>
</tr>
<tr>
<td>• record</td>
<td>• package</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Array Kind</th>
<th>VHDL</th>
<th>Verilog/SystemVerilog</th>
</tr>
</thead>
<tbody>
<tr>
<td>• single-dimensional</td>
<td>• any combination of unpacked, dynamic and associative arrays&lt;sup&gt;3&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>• multi-dimensional</td>
<td>• real/shortreal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• float</td>
<td></td>
</tr>
</tbody>
</table>

---

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.
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 4-30. Memory List Window**

<table>
<thead>
<tr>
<th>Instance</th>
<th>Range</th>
<th>Depth</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>/top/c/mru_mem</td>
<td>[0:31]</td>
<td>32</td>
<td>3</td>
</tr>
<tr>
<td>/top/c/s0/data_mem...</td>
<td>[0:31]</td>
<td>32</td>
<td>16</td>
</tr>
<tr>
<td>/top/c/s0/tag_mem...</td>
<td>[0:31]</td>
<td>32</td>
<td>3</td>
</tr>
<tr>
<td>/top/c/s1/data_mem...</td>
<td>[0:31]</td>
<td>32</td>
<td>16</td>
</tr>
<tr>
<td>/top/c/s1/tag_mem...</td>
<td>[0:31]</td>
<td>32</td>
<td>3</td>
</tr>
<tr>
<td>/top/c/s2/data_mem...</td>
<td>[0:31]</td>
<td>32</td>
<td>16</td>
</tr>
<tr>
<td>/top/c/s2/tag_mem...</td>
<td>[0:31]</td>
<td>32</td>
<td>3</td>
</tr>
<tr>
<td>/top/c/s3/data_mem...</td>
<td>[0:31]</td>
<td>32</td>
<td>16</td>
</tr>
<tr>
<td>/top/c/s3/tag_mem...</td>
<td>[0:31]</td>
<td>32</td>
<td>3</td>
</tr>
<tr>
<td>/top/m/mem</td>
<td>[0:255]</td>
<td>256</td>
<td>16</td>
</tr>
</tbody>
</table>

Right-click anywhere in the window to display the popup menu and select one of the options (See Table 4-43).

The **Memories** menu (See Table 4-44) becomes available in the **Main** menu when the Memory List window is active.
### Fields

**Table 4-23. Memory List Window Columns**

<table>
<thead>
<tr>
<th>Column Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance</td>
<td>Hierarchical name of the memory</td>
</tr>
<tr>
<td>Range</td>
<td>Memory range</td>
</tr>
<tr>
<td>Depth</td>
<td>Memory depth</td>
</tr>
<tr>
<td>Width</td>
<td>Word width</td>
</tr>
</tbody>
</table>

**Table 4-24. Memory List Popup Menu**

<table>
<thead>
<tr>
<th>Popup Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Contents</td>
<td>Opens a Memory Data window for the selected memory.</td>
</tr>
<tr>
<td>Memory Declaration</td>
<td>Opens a Source window to the file and line number where the memory is declared.</td>
</tr>
<tr>
<td>Compare Contents</td>
<td>Allows you to compare the selected memory against another memory in the design or an external file.</td>
</tr>
<tr>
<td>Import Data Patterns</td>
<td>Allows you to import data patterns into the selected memory through the Import Memory dialog box.</td>
</tr>
<tr>
<td>Export Data Patterns</td>
<td>Allows you to export data patterns from the selected memory through the Export Memory dialog box.</td>
</tr>
</tbody>
</table>

**Table 4-25. Memories Menu**

<table>
<thead>
<tr>
<th>Popup Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Contents</td>
<td>Refer to items in the Memory List Popup Menu</td>
</tr>
<tr>
<td>Memory Declaration</td>
<td></td>
</tr>
<tr>
<td>Compare Contents</td>
<td></td>
</tr>
<tr>
<td>Import Data Patterns</td>
<td></td>
</tr>
<tr>
<td>Export Data Patterns</td>
<td></td>
</tr>
<tr>
<td>Expand Packed Memories</td>
<td>Toggle the expansion of packed memories.</td>
</tr>
<tr>
<td>Identify Memories Within Cells</td>
<td>Toggle the identification of memories within Verilog cells.</td>
</tr>
<tr>
<td>Show VHDL String as Memory</td>
<td>Toggle the identification of VHDL strings as memories.</td>
</tr>
</tbody>
</table>

### Usage Notes

This section describes tasks for using the Memory List window.
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, ModelSim 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." For details, see Using the WildcardFilter Preference Variable.

**Note**

For post-process debug, you can add the memories into the Wave or List windows but the Memory List window is not available.
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 transcripted 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 `vsim` 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. 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 transcripted 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`.

- **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 in the `modelsim.ini` file.

**Figure 4-31. Message Viewer Window**

Right-click anywhere in the window to open a popup menu (see Table 4-46) that contains multiple selections.

**Note**

The Messages Bar in the Wave window provides indicators as to when a message occurred.

**Message Viewer Window Tasks**

Figure 4-32 and Table 4-26 provide an overview of the Message Viewer and several tasks you can perform.
Figure 4-32. Message Viewer Window — Tasks

Table 4-26. Message Viewer Tasks

<table>
<thead>
<tr>
<th>Icon</th>
<th>Task</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Display a detailed description of the message.</td>
<td>Right-click the message text then choose View Verbose Message.</td>
</tr>
<tr>
<td></td>
<td>Open message in Source window</td>
<td>Double-click the message in the Message Column.</td>
</tr>
<tr>
<td></td>
<td>Expand a hierarchical node</td>
<td>Double-click on a non-leaf node in the Messages column.</td>
</tr>
<tr>
<td>2</td>
<td>Open the Configuration Options for Message Viewer dialog. Provides access to Analysis Questions, Column Layout, Filter Expression, Hierarchy Configuration and Sort Configuration.</td>
<td>Left-click the down arrow to toggle the “drawer” open and closed.</td>
</tr>
<tr>
<td>3</td>
<td>Open the source file and add a bookmark to the location of the object(s).</td>
<td>Double-click the object name(s).</td>
</tr>
<tr>
<td>3</td>
<td>Change the focus of the Structure and Objects windows.</td>
<td>Double-click the hierarchical reference.</td>
</tr>
<tr>
<td>4</td>
<td>Open the source file and set a marker at the line number.</td>
<td>Double-click the file name.</td>
</tr>
</tbody>
</table>
## Table 4-27. Message Viewer Window Columns

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>Keyword for the various categories of messages:</td>
</tr>
<tr>
<td></td>
<td>• DISPLAY</td>
</tr>
<tr>
<td></td>
<td>• SDF</td>
</tr>
<tr>
<td></td>
<td>• TCHK</td>
</tr>
<tr>
<td></td>
<td>• VCD</td>
</tr>
<tr>
<td></td>
<td>• VITAL</td>
</tr>
<tr>
<td></td>
<td>• WLF</td>
</tr>
<tr>
<td></td>
<td>• MISC</td>
</tr>
<tr>
<td></td>
<td>• &lt;user-defined&gt;</td>
</tr>
<tr>
<td>Comment</td>
<td>User comment</td>
</tr>
<tr>
<td>Compulsory</td>
<td>Whether an item was in a compulsory (required) test for ranking</td>
</tr>
<tr>
<td>CPU Time</td>
<td>Total CPU time consumed</td>
</tr>
<tr>
<td>Date</td>
<td>Date the test was run</td>
</tr>
<tr>
<td>File Info</td>
<td>Filename related to the cause of the message, and in some cases</td>
</tr>
<tr>
<td></td>
<td>the line number in parentheses</td>
</tr>
<tr>
<td>Host OS</td>
<td>Operating system in use by the host on which the test was run</td>
</tr>
<tr>
<td>Hostname</td>
<td>Name of host (server) on which the test was run</td>
</tr>
<tr>
<td>instance</td>
<td>Instance or region associated with the message</td>
</tr>
<tr>
<td>Iteration/Delta</td>
<td>Iteration (delta) in which the message occurs</td>
</tr>
<tr>
<td>LOG name</td>
<td>Name (path) to the generated log/transcript file</td>
</tr>
<tr>
<td>MEMUSAGE</td>
<td>Total memory used by the simulator for the test</td>
</tr>
<tr>
<td>Message</td>
<td>Organized tree-structure of the sorted messages, as well as, when</td>
</tr>
<tr>
<td></td>
<td>expanded, the text of the messages.</td>
</tr>
<tr>
<td>Message ID</td>
<td>Message ID</td>
</tr>
<tr>
<td>Message ID Name</td>
<td>Message ID name</td>
</tr>
<tr>
<td>Objects</td>
<td>Object(s) related to the message, if any.</td>
</tr>
<tr>
<td>Process</td>
<td>Process or leaf associated with the message</td>
</tr>
<tr>
<td>Region</td>
<td>Hierarchical region related to the message, if any</td>
</tr>
<tr>
<td>run CWD</td>
<td>Directory in which the test was run</td>
</tr>
<tr>
<td>Seed</td>
<td>Random seed</td>
</tr>
<tr>
<td>Severity</td>
<td>Message severity, such as Warning, Note or Error.</td>
</tr>
<tr>
<td>Sim Time</td>
<td>Total simulation time</td>
</tr>
</tbody>
</table>
Table 4-27. Message Viewer Window Columns (cont.)

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sim Timeunits</td>
<td>Timeunit used by the simulation</td>
</tr>
<tr>
<td>Source File Name</td>
<td>Name of the file where the message originated</td>
</tr>
<tr>
<td>Source File Number</td>
<td>Declaration number of the file associated with the message</td>
</tr>
<tr>
<td>Source Line Number</td>
<td>Line number within &quot;filename&quot; where the message originated</td>
</tr>
<tr>
<td>Test Name</td>
<td>Name of the test</td>
</tr>
<tr>
<td>Test Status</td>
<td>Completion status (OK, Error, etc.)</td>
</tr>
<tr>
<td>Time</td>
<td>Time of simulation when the message was issued.</td>
</tr>
<tr>
<td>Timing Check Kind</td>
<td>Information about timing checks</td>
</tr>
<tr>
<td>User ID</td>
<td>Username under which the test was run</td>
</tr>
<tr>
<td>VRM Context</td>
<td>Username under which the test was run</td>
</tr>
<tr>
<td>Vsim Args</td>
<td>Arguments passed to vsim command</td>
</tr>
<tr>
<td>WLF Filename</td>
<td>Name of WLF file from which message was imported</td>
</tr>
<tr>
<td>WLF Name</td>
<td>Name (path) to the generated WLF file</td>
</tr>
<tr>
<td>WLF Raw Time</td>
<td>Simulation time (in ticks) associated with the message</td>
</tr>
<tr>
<td>WLF Time Unit</td>
<td>Simulation time unit</td>
</tr>
</tbody>
</table>

Table 4-28. Message Viewer Window Popup Menu

<table>
<thead>
<tr>
<th>Popup Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reload Viewer Data</td>
<td>Opens a Source window for the file, and in some cases takes you to</td>
</tr>
<tr>
<td></td>
<td>the associated line number.</td>
</tr>
<tr>
<td>View</td>
<td>Opens selected item:</td>
</tr>
<tr>
<td></td>
<td>* Verbose Message</td>
</tr>
<tr>
<td></td>
<td>* Message Source</td>
</tr>
<tr>
<td></td>
<td>* Log File</td>
</tr>
<tr>
<td></td>
<td>* Object Declarations</td>
</tr>
<tr>
<td></td>
<td>* Change Environment</td>
</tr>
<tr>
<td></td>
<td>* Waveform:</td>
</tr>
<tr>
<td></td>
<td>* Go to Time in Wave</td>
</tr>
<tr>
<td></td>
<td>* Add Objs to Wave</td>
</tr>
<tr>
<td></td>
<td>Verbose Message dialog box with details about message</td>
</tr>
<tr>
<td></td>
<td>Source code at line number where message is</td>
</tr>
<tr>
<td></td>
<td>Log file, in a Source window</td>
</tr>
<tr>
<td></td>
<td>Object window, to view declarations</td>
</tr>
<tr>
<td></td>
<td>Change environment</td>
</tr>
<tr>
<td></td>
<td>Waveform window, opens:</td>
</tr>
<tr>
<td></td>
<td>at time of selected message</td>
</tr>
<tr>
<td></td>
<td>adds objects associated with selected message</td>
</tr>
<tr>
<td>Analysis Questions</td>
<td>Opens Analysis Questions dialog box; used for saving and managing specific</td>
</tr>
<tr>
<td></td>
<td>queries of the data.</td>
</tr>
<tr>
<td>Filter Expressions</td>
<td>Opens Filter Expressions dialog; used for saving and managing filters.</td>
</tr>
</tbody>
</table>
Table 4-28. Message Viewer Window Popup Menu (cont.)

<table>
<thead>
<tr>
<th>Popup Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierarchy Configurations</td>
<td>Opens Hierarchy Configurations dialog box; used for saving and managing particular hierarchy configurations of the data.</td>
</tr>
<tr>
<td>Column Layouts</td>
<td>Opens Configure Column Layout dialog; used for creating, editing and managing the configuration of columns.</td>
</tr>
<tr>
<td>Show Titles in Hier Column</td>
<td>Toggles on and off showing the titles within the hierarchy column</td>
</tr>
<tr>
<td>Global Options</td>
<td>Configures how/when Message Viewer opens.</td>
</tr>
<tr>
<td>Edit Transforms...</td>
<td>Opens a dialog which will open a transform rules file for editing with the Transform Rule File Editor.</td>
</tr>
<tr>
<td>Load/Save Setup File</td>
<td>Loads/Saves a particular setup to a name you specify.</td>
</tr>
<tr>
<td>Expand/Collapse Selected/All</td>
<td>Manipulates the expansion of the Messages column.</td>
</tr>
</tbody>
</table>

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, as highlighted in Figure 4-33.

Figure 4-33. 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

To access: Filter Expressions > [Configure Filter] > Filter Expressions dialog box
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 Edit Filter Expression dialog box in Figure 4-34 shows an example where you want to show all messages, either errors or warnings, that reference the 15th line of the file `cells.v`.

![Figure 4-34. Edit Filter Expression Dialog Box](image)

**Fields**

- **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, NAND, and NOR.
  - **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.
  - **Inclusion field** — specifies whether the Column field should or should not contain a given value.
  - For text-based filter values your choices are: Contains, Doesn’t Contain, or Exact.
  - For numeric- and time-based filter values your choices are: ==, !=, <, <=, >, and >=.
  - **Case Sensitivity field** — specifies whether your filter rule should treat your filter value as Case Sensitive or Case Insensitive. This field only applies to text-based filter values.
  - **Filter Value field** — specifies the filter value associated with your filter rule.
  - **Time Unit field** — specifies the time unit. Your choices are: fs, ps, ns, us, ms. This field only applies to the Time selection from the Column field.
- **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, see Current Time Label.)

Viewable data objects include:

- signals
- nets
- registers
- constants and variables not declared in a process
- generics
- parameters

![Figure 4-35. Objects Window](image)

Right-click anywhere in the window to display the popup menu and an option displayed in Table 4-49:
The Object filter toolbar provides filtering of design objects appearing in the Objects window.

**Figure 4-36. Object Window Toolbar**

| Fields |

**Table 4-29. Columns in the Objects Window**

<table>
<thead>
<tr>
<th>Column name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>the name of each object in the current region</td>
</tr>
<tr>
<td>Value</td>
<td>the current value of each object</td>
</tr>
<tr>
<td>Kind</td>
<td>the object type</td>
</tr>
<tr>
<td>Mode</td>
<td>the object mode (internal, in, out, and so forth.)</td>
</tr>
</tbody>
</table>

**Table 4-30. Objects Window Popup Menu**

<table>
<thead>
<tr>
<th>Popup Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Declaration</td>
<td>Opens a Source window to the declaration of the object</td>
</tr>
<tr>
<td>View Memory Contents</td>
<td></td>
</tr>
<tr>
<td>Add Wave</td>
<td>Adds the selected object(s) to the Wave window</td>
</tr>
<tr>
<td>Add Wave New</td>
<td>Creates a new instance of the Wave window and adds the selected object(s) to that window.</td>
</tr>
<tr>
<td>Add Wave To</td>
<td>Opens a drop down list of Wave windows when multiple windows exist. Adds the selected object(s) to the selected Wave window.</td>
</tr>
<tr>
<td>Add Dataflow</td>
<td>Adds the selected object(s) to a Dataflow window</td>
</tr>
<tr>
<td>Add to</td>
<td>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.</td>
</tr>
<tr>
<td>Copy</td>
<td>Copies information about the object to the clipboard</td>
</tr>
<tr>
<td>Find</td>
<td>Opens the Find box</td>
</tr>
<tr>
<td>Insert Breakpoint</td>
<td>Adds a breakpoint for the selected object</td>
</tr>
</tbody>
</table>
Objects Window Tasks

This section describes tasks for using the Objects window.

Interacting with Other Windows ................................................. 157
Setting Signal Radix. .............................................................. 157
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, 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 4-37), 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.
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.
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.
Processes Window

To access:

- View > Process
- view process command

Use this window to view a list of HDL 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 4-38. Processes Window**

![Processes Window Screenshot]

**Fields**

**Table 4-32. Processes Window Column Descriptions**

<table>
<thead>
<tr>
<th>Column Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the process.</td>
</tr>
<tr>
<td>Class Info</td>
<td>SystemVerilog class object id or UVM component name.</td>
</tr>
<tr>
<td>Order</td>
<td>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.</td>
</tr>
</tbody>
</table>
### Processes Window

<table>
<thead>
<tr>
<th>Column Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Path</td>
<td>Hierarchical parent pathname of the process</td>
</tr>
<tr>
<td>State</td>
<td>Process state.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Wait</strong> — Indicates the process is waiting for a wake up trigger (change in VHDL signal, Verilog net, or a time period).</td>
</tr>
<tr>
<td></td>
<td>• <strong>Ready</strong> — Indicates the process is scheduled to be executed in current simulation phase (or in active simulation queue) of current delta cycle.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Active</strong> — Indicates the process is currently active and being executed.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Queued</strong> — Indicates the process is scheduled to be executed in current delta cycle, but not in current simulation phase (or in active simulation queue).</td>
</tr>
<tr>
<td></td>
<td>• <strong>Done</strong> — Indicates the process has been terminated, and will never restart during current simulation run.</td>
</tr>
<tr>
<td>Processes in the Idle and Wait states are distinguished as follows. Idle processes 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.</td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>Process type, according to the language, including the following types:</td>
</tr>
<tr>
<td></td>
<td>• Always</td>
</tr>
<tr>
<td></td>
<td>• Assign</td>
</tr>
<tr>
<td></td>
<td>• Final</td>
</tr>
<tr>
<td></td>
<td>• Fork-Join (dynamic process like fork-join, sc_spawn, and so forth.)</td>
</tr>
<tr>
<td></td>
<td>• Initial</td>
</tr>
<tr>
<td></td>
<td>• Implicit (internal processes created by simulator like Implicit wires, and so forth.)</td>
</tr>
<tr>
<td></td>
<td>• Primitive (UDP, Gates, and so forth.)</td>
</tr>
<tr>
<td></td>
<td>• VHDL Process</td>
</tr>
</tbody>
</table>
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.

![Figure 4-39. Source Window](image)

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.
Fields

- **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. (For details, refer to **Current Time Label**.)
- **Code Folding Indicators** — denotes sections of code that can be folded or expanded

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.
Using the Source Window

Use the Source window to view and analyze your data.

Dragging and Dropping Objects into the Wave and List Windows .................. 166
Setting your Context by Navigating Source Files ........................................ 167
Setting File-Line Breakpoints with the GUI .................................................. 169
Adding File-Line Breakpoints with the bp Command .................................. 170
Editing File-Line Breakpoints ................................................................. 170
Setting Conditional Breakpoints ............................................................. 172
Checking Object Values and Descriptions ................................................... 174
Marking Lines with Bookmarks ............................................................... 175
Performing Incremental Search for Specific Code ....................................... 175
Customizing the Source Window .............................................................. 176

Dragging and Dropping Objects into the Wave and List Windows

ModelSim 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.

When an object is dragged and dropped into the Wave window, the `add wave` command will be reflected in the Transcript window.

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 4-41 shows the pop-up menu the tool displays after you select then right-click an instance name in a source file.

Figure 4-41. 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:
Window Reference

Using the Source Window

- **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 exist, 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 aCn 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

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 on 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 on 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.

**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 4-42**).
Figure 4-42. 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.

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 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 4-43**.

**Figure 4-43. 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 4-43.

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 *Questa SV/AFV 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 4-43) 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 4-44. Source Code for source.sv**

```verilog
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
```

**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 4-44) 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 4-44), 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 on 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 4-43) 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, 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 4-45. 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 4-46) 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.
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 `DefaultRadix` 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 `Source > 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
- Button: View Objects Window Button

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.

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.
Right-click an object in the Structure window to display the popup menu and select an option in Table 4-57.
### Table 4-33. Structure Window Popup Menu

<table>
<thead>
<tr>
<th>Popup Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Declaration</td>
<td>Opens the source file and bookmarks the object.</td>
</tr>
<tr>
<td>View Instantiation</td>
<td>Opens the source file and bookmarks the object.</td>
</tr>
<tr>
<td>Add Wave</td>
<td>Adds the selected object(s) to the Wave window.</td>
</tr>
<tr>
<td>Add Wave New</td>
<td>Creates a new instance of the Wave window and adds the selected object(s) to that window.</td>
</tr>
<tr>
<td>Add Wave To</td>
<td>Opens a drop down list of Wave windows when multiple windows exist. Adds the selected object(s) to the selected Wave window.</td>
</tr>
<tr>
<td>Add Dataflow</td>
<td>Adds the selected object(s) to a Dataflow window.</td>
</tr>
<tr>
<td>Add to</td>
<td>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.</td>
</tr>
<tr>
<td>Copy</td>
<td>Copies the object instance path to the clipboard</td>
</tr>
<tr>
<td>Find</td>
<td>Opens the Search Bar (at bottom of window) in the Find mode to make searching for objects easier, especially with large designs.</td>
</tr>
<tr>
<td>Save Selected</td>
<td>Saves all hierarchy under the selected instance.</td>
</tr>
<tr>
<td>Expand Selected</td>
<td>Displays the hierarchy of the object recursively.</td>
</tr>
<tr>
<td>Collapse Selected</td>
<td>Closes the hierarchy of the object.</td>
</tr>
<tr>
<td>Collapse All</td>
<td>Collapses the hierarchy to the top instance.</td>
</tr>
<tr>
<td>XML Import Hint</td>
<td>Displays the XML Import Hint dialog box with information about the Link Type and Name</td>
</tr>
</tbody>
</table>
Table 4-33. Structure Window Popup Menu (cont.)

<table>
<thead>
<tr>
<th>Popup Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show</td>
<td>Lists the design unit types that are currently displayed.</td>
</tr>
<tr>
<td></td>
<td>• Processes</td>
</tr>
<tr>
<td></td>
<td>• Functions</td>
</tr>
<tr>
<td></td>
<td>• Packages</td>
</tr>
<tr>
<td></td>
<td>• Tasks</td>
</tr>
<tr>
<td></td>
<td>• Statement</td>
</tr>
<tr>
<td></td>
<td>• VPackages</td>
</tr>
<tr>
<td></td>
<td>• VITypedef</td>
</tr>
<tr>
<td></td>
<td>• SVClass</td>
</tr>
<tr>
<td></td>
<td>• Class Instances</td>
</tr>
<tr>
<td></td>
<td>• Capacity</td>
</tr>
<tr>
<td></td>
<td>• Change Filter</td>
</tr>
</tbody>
</table>

Table 4-34. Columns in the Structure Window

<table>
<thead>
<tr>
<th>Column name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design Unit</td>
<td>The name of the design unit</td>
</tr>
<tr>
<td>Design Unit Type</td>
<td>The type of design unit</td>
</tr>
</tbody>
</table>

Structure Window Tasks

This section describes tasks for using the Structure window.

Display Source Code of a Structure Window Object ........................................... 180
Add Structure Window Objects to Other Windows .................................................. 181
Finding Items in the Structure Window ............................................................... 181
Filtering Structure Window Objects ................................................................. 182

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, 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 (respectively) 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 4-48). With ‘Search While Typing’ enabled (the default) each keypress changes the pattern and restarts the search immediately.

![Figure 4-48. Find Mode Popup Displays Matches](image)

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:
Window Reference
Structure Window Tasks

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
- **VlPackage** — Verilog Packages
- **VITypedef** — Verilog Type Definitions
- **Cell Instances** — Verilog cell instances or VHDL architecture instance.
- **Capacity** — Memory capacity design unit
- **Subprogram** — VHDL procedures and functions; Verilog functions and tasks
Window Reference

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 ModelSim. 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 Transcript tab contains the command line interface, identified by the ModelSim prompt, and the simulation interface, identified by the VSIM prompt.

When the Transcript window is active, a "Transcript" menu selection appears in the Main window menu bar.

When undocked, the Transcript window allows access to the following toolbars:

- Standard Toolbar
- Help Toolbar

Fields

- **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.

- **ModelSim 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.

- **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** 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.
Window Reference

Transcript Window Tasks

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 4-49).
Disabling Creation of the Transcript File

You can disable the creation of the transcript file by using a ModelSim command immediately after ModelSim 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.
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.
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.
- **Virtual objects** — virtual signals and virtual functions.

**Figure 4-52. Watch Window**

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.
**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 4-53 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 4-53. 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 (see Table 4-74).
### Table 4-35. Watch Window Popup Menu

<table>
<thead>
<tr>
<th>Popup menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Add the selected item or items to the desired window</td>
</tr>
<tr>
<td>Force</td>
<td>Opens the Force Selected Signal dialog box, which allows you to force the signal to given value. Refer to the <code>force</code> command for details about the options.</td>
</tr>
<tr>
<td>NoForce</td>
<td>Removes the force on the selected object. Refer to the <code>noforce</code> command.</td>
</tr>
<tr>
<td>Clock</td>
<td>Performs actions related to the <code>force</code> command with the <code>-repeat</code> argument.</td>
</tr>
<tr>
<td>Change</td>
<td>Performs actions related to the <code>change</code> command.</td>
</tr>
<tr>
<td>Follow Selection Context</td>
<td>Changes the current context in the Structure window.</td>
</tr>
<tr>
<td>Save Format</td>
<td>Saves a new (or loads an existing) .do file containing <code>add watch</code> commands to recreate the Watch window.</td>
</tr>
<tr>
<td>Load Format</td>
<td></td>
</tr>
<tr>
<td>Group</td>
<td>Groups several selected signals into a single item.</td>
</tr>
<tr>
<td>UnGroup</td>
<td>Breaks a previously created group back into its individual signals.</td>
</tr>
<tr>
<td>Properties</td>
<td>Opens the Properties dialog box, which allows you to alter the properties of the selected signal or group, including:</td>
</tr>
<tr>
<td></td>
<td>* Header name</td>
</tr>
<tr>
<td></td>
<td>* Radix type</td>
</tr>
<tr>
<td></td>
<td>This option is not available when multiple signals are selected.</td>
</tr>
<tr>
<td>Delete Item</td>
<td>Removes the selected signal from the window. You can alternatively use the delete key.</td>
</tr>
<tr>
<td>Clear All</td>
<td>Removes all signals from the window.</td>
</tr>
</tbody>
</table>
### Table 4-36. Watch Window Menu

<table>
<thead>
<tr>
<th>Popup menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Force</td>
<td>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.</td>
</tr>
<tr>
<td>NoForce</td>
<td>Removes the force on the selected object. Refer to the noforce command.</td>
</tr>
<tr>
<td>Clock</td>
<td>Performs actions related to the force command with the -repeat argument.</td>
</tr>
<tr>
<td>Change</td>
<td>Performs actions related to the change command.</td>
</tr>
<tr>
<td>Follow Selection Context</td>
<td>Changes the current context in the Structure window.</td>
</tr>
<tr>
<td>Save Format</td>
<td>Saves a new (or loads an existing) .do file containing add watch commands to recreate the Watch window.</td>
</tr>
<tr>
<td>Load Format</td>
<td></td>
</tr>
<tr>
<td>Group</td>
<td>Groups several selected signals into a single item.</td>
</tr>
<tr>
<td>UnGroup</td>
<td>Breaks a previously created group back into its individual signals.</td>
</tr>
<tr>
<td>Tile</td>
<td>Reorganizes the items in the Watch window into different tiled formats.</td>
</tr>
<tr>
<td>Delete Item</td>
<td>Removes the selected signal from the window. You can alternatively use the delete key.</td>
</tr>
<tr>
<td>Clear All</td>
<td>Removes all signals from the window.</td>
</tr>
</tbody>
</table>

### 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 4-54, /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 4-54. 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.

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

When you drag and drop objects into the Wave window, the add wave command is reflected in the Transcript window.
Refer to Adding Objects to the Wave Window for more information about adding objects to the Wave 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.

The white bar along the left margin indicates the selected Wave window or pane of a split wave window (see Splitting Wave Window Panes).

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.

Values Pane

The values pane displays the values of the displayed signals. You can resize the values pane by clicking on 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 (Figure 4-57), 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 4-57. Setting the Global Signal Radix from the Wave Window](image)

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.
Waveform Pane

Figure 4-59 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.

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 (see Figure 4-60). This
toolbox shows a group of icons that gives you quick access to actions you can perform on the waveform display, as described in Table 4-37.

**Figure 4-60. Analog Sidebar Toolbox**

Cursor Pane
Figure 4-61 shows the Cursor Pane, which displays cursor names, cursor values and the cursor locations on the timeline. You can link cursors so that they move across the timeline together. See Linking Cursors in the Waveform Analysis chapter.

**Figure 4-61. Cursor Pane**

On the left side of this pane is a group of icons called the Cursor and Timeline Toolbox (see Working with Cursors). This toolbox gives you quick access to cursor and timeline features and configurations. See Measuring Time with Cursors in the Wave Window 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 4-62. Wave Window - Message Bar**

The message indicators (the down-pointing arrows) are color-coded as follows:

- **Red** — Indicates an error.
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 on 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 on 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**
  
  b. 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, as shown in **Figure 4-63**

**Figure 4-63. View Objects Window Dropdown Menu**

![View Objects Window Dropdown Menu]

**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
Wave Window

- 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 transactions (indicated by a blue four point star) — see for more information

- Virtual objects (indicated by an orange diamond) — virtual signals, buses, and functions, see; Virtual Objects for more information

  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 on 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 (see Saving the Window Format).

Wave Window Toolbars

  The Wave window (in the undocked Wave window) gives you quick access to the following toolbars:

  - Standard Toolbar
  - Compile Toolbar
  - Simulate Toolbar
  - Step Toolbar
  - Wave Cursor Toolbar
  - Wave Edit Toolbar
  - Wave Toolbar
  - Zoom Toolbar
  - Wave Expand Time Toolbar
Fields

Table 4-37. Analog Sidebar Icons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Icon]</td>
<td>Open Wave Properties</td>
<td>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 displayed by choosing Format &gt; Format... &gt; Analog (custom) from the main menu.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Toggle Row Height</td>
<td>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.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Rescale to fit Y data</td>
<td>Changes the waveform height so that it fits top-to-bottom within the current height of the row.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Show menu of other actions</td>
<td>Displays • View Min Y • View Max Y • Overlay Above • Overlay Below • Colorize All • Colorize Selected</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Drag to resize waveform height</td>
<td>Creates an up/down dragging arrow that you can use to temporarily change the height of the row containing the analog waveform.</td>
</tr>
</tbody>
</table>

Table 4-38. Window Icons

<table>
<thead>
<tr>
<th>Icon shape</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSM button</td>
<td>![Example]</td>
<td>opens the FSM Viewer window</td>
</tr>
<tr>
<td>Null</td>
<td>![Example]</td>
<td>verilog/system verilog name event</td>
</tr>
</tbody>
</table>
You can manipulate the user interface with various keyboard and mouse actions.

**Window-Specific Keyboard Shortcuts**

You can open a dynamic list of common (predefined) and user-defined keyboard shortcuts for many ModelSim windows by entering "Ctrl+/", on your keyboard.

For example, Figure 5-1 shows the list of keyboard shortcuts provided for the Source window.

![Figure 5-1. Keyboard Shortcuts for Source Window](image)

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.

**ModelSim Windows That Have Keyboard Shortcuts**

The following windows have keyboard shortcuts assigned:

- Dataflow Window
- Library Window
- Objects Window
- Source Window
- Structure Window
- Transcript Window
- Wave Window
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 ModelSim 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 Mentor product that is based upon the ModelSim GUI, such as ADMS, 0-In, MVC, and Codelink.

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 ModelSim simulator.

Figure 5-2 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 5-2. 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 Shortcuts 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.
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.
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 ModelSim to open the Notepad editor).

### Table 5-1. Mouse Shortcuts

<table>
<thead>
<tr>
<th>Mouse - UNIX and Windows</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Click the left mouse button</td>
<td>relocate the cursor</td>
</tr>
<tr>
<td>Click and drag the left mouse button</td>
<td>select an area</td>
</tr>
<tr>
<td>Shift-click the left mouse button</td>
<td>extend selection</td>
</tr>
<tr>
<td>Double-click the left mouse button</td>
<td>select a word</td>
</tr>
<tr>
<td>Double-click and drag the left mouse button</td>
<td>select a group of words</td>
</tr>
<tr>
<td>Ctrl-click the left mouse button</td>
<td>move insertion cursor without changing the selection</td>
</tr>
<tr>
<td>Click the left mouse button on a previous ModelSim or VSIM prompt</td>
<td>copy and paste previous command string to current prompt</td>
</tr>
<tr>
<td>Click the middle mouse button</td>
<td>paste selection to the clipboard</td>
</tr>
<tr>
<td>Click and drag the middle mouse button</td>
<td>scroll the window</td>
</tr>
</tbody>
</table>
### Table 5-2. Keyboard Shortcuts

<table>
<thead>
<tr>
<th>Keystrokes - UNIX and Windows</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left Arrow</td>
<td>move cursor left or right one character</td>
</tr>
<tr>
<td>Right Arrow</td>
<td></td>
</tr>
<tr>
<td>Ctrl + Left Arrow</td>
<td>move cursor left or right one word</td>
</tr>
<tr>
<td>Ctrl + Right Arrow</td>
<td></td>
</tr>
<tr>
<td>Shift + Any Arrow</td>
<td>extend text selection</td>
</tr>
<tr>
<td>Ctrl + Shift + Left Arrow</td>
<td>extend text selection by one word</td>
</tr>
<tr>
<td>Ctrl + Shift + Right Arrow</td>
<td></td>
</tr>
<tr>
<td>Up Arrow</td>
<td>Transcript window: scroll through command history</td>
</tr>
<tr>
<td>Down Arrow</td>
<td>Source window: move cursor one line up or down</td>
</tr>
<tr>
<td>Ctrl + Up Arrow</td>
<td>Transcript window: moves cursor to first or last line</td>
</tr>
<tr>
<td>Ctrl + Down Arrow</td>
<td>Source window: moves cursor up or down one paragraph</td>
</tr>
<tr>
<td>Alt + /</td>
<td>Open a pop-up command prompt for entering commands.</td>
</tr>
<tr>
<td>Ctrl + Home</td>
<td>move cursor to the beginning of the text</td>
</tr>
<tr>
<td>Ctrl + End</td>
<td>move cursor to the end of the text</td>
</tr>
<tr>
<td>Backspace</td>
<td>delete character to the left</td>
</tr>
<tr>
<td>Ctrl + h (UNIX only)</td>
<td>delete character to the right</td>
</tr>
<tr>
<td>Delete</td>
<td></td>
</tr>
<tr>
<td>Ctrl + d (UNIX only)</td>
<td></td>
</tr>
<tr>
<td>Esc (Windows only)</td>
<td>cancel</td>
</tr>
<tr>
<td>Alt</td>
<td>activate or inactivate menu bar mode</td>
</tr>
<tr>
<td>Alt-F4</td>
<td>close active window</td>
</tr>
<tr>
<td>Home</td>
<td>move cursor to the beginning of the line</td>
</tr>
<tr>
<td>Ctrl + a</td>
<td></td>
</tr>
<tr>
<td>Ctrl + Shift + a</td>
<td>select all contents of active window</td>
</tr>
<tr>
<td>Ctrl + b</td>
<td>move cursor left</td>
</tr>
<tr>
<td>Ctrl + d</td>
<td>delete character to the right</td>
</tr>
<tr>
<td>End</td>
<td>move cursor to the end of the line</td>
</tr>
<tr>
<td>Ctrl + e</td>
<td></td>
</tr>
<tr>
<td>Ctrl + f (UNIX)</td>
<td>move cursor right one character</td>
</tr>
<tr>
<td>Right Arrow (Windows)</td>
<td></td>
</tr>
<tr>
<td>Ctrl + k</td>
<td>delete to the end of line</td>
</tr>
</tbody>
</table>
### Table 5-2. Keyboard Shortcuts (cont.)

<table>
<thead>
<tr>
<th>Keystrokes - UNIX and Windows</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ctrl + n</td>
<td>move cursor one line down (Source window only under Windows)</td>
</tr>
<tr>
<td>Ctrl + o (UNIX only)</td>
<td>insert a new line character at the cursor</td>
</tr>
<tr>
<td>Ctrl + p</td>
<td>move cursor one line up (Source window only under Windows)</td>
</tr>
<tr>
<td>Ctrl + s (UNIX) Ctrl + f (Windows)</td>
<td>find</td>
</tr>
<tr>
<td>Ctrl + t</td>
<td>reverse the order of the two characters on either side of the cursor</td>
</tr>
<tr>
<td>Ctrl + u</td>
<td>delete line</td>
</tr>
<tr>
<td>Page Down Ctrl + v (UNIX only)</td>
<td>move cursor down one screen</td>
</tr>
<tr>
<td>Ctrl + x</td>
<td>cut the selection</td>
</tr>
<tr>
<td>Ctrl + s Ctrl + x (UNIX Only)</td>
<td>save</td>
</tr>
<tr>
<td>Ctrl + v</td>
<td>paste the selection</td>
</tr>
<tr>
<td>Ctrl + a (Windows Only)</td>
<td>select the entire contents of the widget</td>
</tr>
<tr>
<td>Ctrl + \</td>
<td>clear any selection in the widget</td>
</tr>
<tr>
<td>Ctrl + - (UNIX) Ctrl + / (UNIX) Ctrl + z (Windows)</td>
<td>undoes previous edits in the Source window</td>
</tr>
<tr>
<td>Meta + &lt; (UNIX only)</td>
<td>move cursor to the beginning of the file</td>
</tr>
<tr>
<td>Meta + &gt; (UNIX only)</td>
<td>move cursor to the end of the file</td>
</tr>
<tr>
<td>Page Up Meta + v (UNIX only)</td>
<td>move cursor up one screen</td>
</tr>
<tr>
<td>Ctrl + c</td>
<td>copy selection</td>
</tr>
<tr>
<td>F3</td>
<td>Performs a Find Next action in the Source window.</td>
</tr>
<tr>
<td>F4 Shift+F4</td>
<td>Change focus to next pane in main window Change focus to previous pane in main window</td>
</tr>
<tr>
<td>F5 Shift+F5</td>
<td>Toggle between expanding and restoring size of pane to fit the entire main window Toggle on/off the pane headers.</td>
</tr>
<tr>
<td>F8</td>
<td>search for the most recent command that matches the characters typed (Main window only)</td>
</tr>
</tbody>
</table>
The Main window allows insertions or pastes only after the prompt; therefore, you don’t 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 5-4. 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 5-3. List Window Keyboard Shortcuts

<table>
<thead>
<tr>
<th>Key - UNIX and Windows</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left Arrow</td>
<td>scroll listing left (selects and highlights the item to the left of the currently selected item)</td>
</tr>
<tr>
<td>Right Arrow</td>
<td>scroll listing right (selects and highlights the item to the right of the currently selected item)</td>
</tr>
<tr>
<td>Up Arrow</td>
<td>scroll listing up</td>
</tr>
<tr>
<td>Down Arrow</td>
<td>scroll listing down</td>
</tr>
<tr>
<td>Page Up</td>
<td>scroll listing up by page</td>
</tr>
<tr>
<td>Ctrl + Up Arrow</td>
<td>scroll listing up by page</td>
</tr>
<tr>
<td>Page Down</td>
<td>scroll listing down by page</td>
</tr>
<tr>
<td>Ctrl + Down Arrow</td>
<td>scroll listing down by page</td>
</tr>
<tr>
<td>Tab</td>
<td>searches forward (down) to the next transition on the selected signal</td>
</tr>
<tr>
<td>Shift + Tab</td>
<td>searches backward (up) to the previous transition on the selected signal</td>
</tr>
<tr>
<td>Shift + Left Arrow</td>
<td>extends selection left/right</td>
</tr>
<tr>
<td>Shift + Right Arrow</td>
<td>extends selection left/right</td>
</tr>
<tr>
<td>Ctrl + f (Windows)</td>
<td>opens the Find dialog box to find the specified item label within the list display</td>
</tr>
<tr>
<td>Ctrl + s (UNIX)</td>
<td>opens the Find dialog box to find the specified item label within the list display</td>
</tr>
</tbody>
</table>

Wave Window Mouse and Keyboard Shortcuts

The following mouse actions and keystrokes can be used in the Wave window.

Table 5-4. Wave Window Mouse Shortcuts

<table>
<thead>
<tr>
<th>Mouse action</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ctrl + Click left mouse button and drag</td>
<td>zoom area (in)</td>
</tr>
<tr>
<td>Ctrl + Click left mouse button and drag</td>
<td>zoom out</td>
</tr>
</tbody>
</table>
Table 5-4. Wave Window Mouse Shortcuts (cont.)

<table>
<thead>
<tr>
<th>Mouse action</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ctrl + Click left mouse button and drag</td>
<td>zoom fit</td>
</tr>
<tr>
<td>Click left mouse button and drag</td>
<td>moves closest cursor</td>
</tr>
<tr>
<td>Ctrl + Click left mouse button on a scroll bar arrow</td>
<td>scrolls window to very top or bottom (vertical scroll) or far left or right (horizontal scroll)</td>
</tr>
<tr>
<td>Click middle mouse button in scroll bar (UNIX only)</td>
<td>scrolls window to position of click</td>
</tr>
<tr>
<td>Shift + scroll with middle mouse button</td>
<td>scrolls window</td>
</tr>
</tbody>
</table>

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

Table 5-5. Wave Window Keyboard Shortcuts

<table>
<thead>
<tr>
<th>Keystroke</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>bring into view and center the currently active cursor</td>
</tr>
<tr>
<td>i</td>
<td>zoom in (mouse pointer must be over the cursor or waveform panes)</td>
</tr>
<tr>
<td>Shift + i</td>
<td>zoom in (mouse pointer must be over the cursor or waveform panes)</td>
</tr>
<tr>
<td>o</td>
<td>zoom out (mouse pointer must be over the cursor or waveform panes)</td>
</tr>
<tr>
<td>Shift + o</td>
<td>zoom out (mouse pointer must be over the cursor or waveform panes)</td>
</tr>
<tr>
<td>f</td>
<td>zoom full (mouse pointer must be over the cursor or waveform panes)</td>
</tr>
<tr>
<td>Shift + f</td>
<td>zoom full (mouse pointer must be over the cursor or waveform panes)</td>
</tr>
<tr>
<td>l</td>
<td>zoom last (mouse pointer must be over the cursor or waveform panes)</td>
</tr>
<tr>
<td>Shift + l</td>
<td>zoom last (mouse pointer must be over the cursor or waveform panes)</td>
</tr>
<tr>
<td>r</td>
<td>zoom range (mouse pointer must be over the cursor or waveform panes)</td>
</tr>
<tr>
<td>Shift + r</td>
<td>zoom range (mouse pointer must be over the cursor or waveform panes)</td>
</tr>
<tr>
<td>m</td>
<td>zooms all open Wave windows to the zoom range of the active window.</td>
</tr>
<tr>
<td>Up Arrow</td>
<td>scrolls entire window up or down one line, when mouse pointer is over waveform pane</td>
</tr>
<tr>
<td>Down Arrow</td>
<td>scrolls highlight up or down one line, when mouse pointer is over pathname or values pane</td>
</tr>
<tr>
<td>Left Arrow</td>
<td>scroll pathname, values, or waveform pane left</td>
</tr>
</tbody>
</table>
Table 5-5. Wave Window Keyboard Shortcuts (cont.)

<table>
<thead>
<tr>
<th>Keystroke</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Right Arrow</td>
<td>scroll pathname, values, or waveform pane right</td>
</tr>
<tr>
<td>Page Up</td>
<td>scroll waveform pane up by a page</td>
</tr>
<tr>
<td>Page Down</td>
<td>scroll waveform pane down by a page</td>
</tr>
<tr>
<td>Tab</td>
<td>search forward (right) to the next transition on the selected signal - finds the next edge</td>
</tr>
<tr>
<td>Shift + Tab</td>
<td>search backward (left) to the previous transition on the selected signal - finds the previous edge</td>
</tr>
<tr>
<td>Ctrl+G</td>
<td>automatically create a group for the selected signals by region with the name Group&lt;n&gt;. If you use this shortcut on signals for which there is already a “Group&lt;n&gt;” they will be placed in that region’s group rather than creating a new one.</td>
</tr>
<tr>
<td>Ctrl + F (Windows)</td>
<td>open the find dialog box; searches within the specified field in the pathname pane for text strings</td>
</tr>
<tr>
<td>Ctrl + S (UNIX)</td>
<td>open the find dialog box; searches within the specified field in the pathname pane for text strings</td>
</tr>
<tr>
<td>Ctrl + Left Arrow</td>
<td>scroll pathname, values, or waveform pane left or right by a page</td>
</tr>
<tr>
<td>Ctrl + Right Arrow</td>
<td>scroll pathname, values, or waveform pane left or right by a page</td>
</tr>
</tbody>
</table>
Keyboard Shortcuts and Mouse Actions

Wave Window Mouse and Keyboard Shortcuts
The ModelSim 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 platforms. The variable values save automatically when you exit ModelSim. 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**

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.

These layout modes are fully customizable and the GUI stores your manipulations in 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.

**Result**

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.

Result

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>.

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.

User-Defined Buttons and Menus

You can create Tcl processes (procs) that add user-defined buttons and menus to the main window of the ModelSim 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 6-1).

Figure 6-1. 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"
```
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 ModelSim 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.

After defining a new radix, the radix will be available for use in the List, Watch, and Wave windows or with the `examine` command.
GUI Customization

User-Defined Radices

There are four commands used to manage user defined radices:

- `radix define`
- `radix names`
- `radix list`
- `radix delete`

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 6-2).

```
{<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 6-1 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 6-1. Using the `radix define` Command**

```text
radix define States {
  11'b000000000001 "IDLE",
  11'b000000000010 "CTRL",
  11'b000000000100 "WT_WD_1",
  11'b000000001000 "WT_WD_2",
  11'b000000100000 "WT_BLK_1",
  11'b000001000000 "WT_BLK_2",
}
GUI Customization
User-Defined Radices

Figure 6-2 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 6-1).

Figure 6-3 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 6-1).

Using radix define to Specify Radix Color

The following example illustrates how to use the radix define command to specify the radix color:

```
Example 6-2. Using radix define to Specify Color

radix define States {
    11'b00000000001 "IDLE" -color yellow,
```
User-Defined Radices

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'b00010000000 "WT_BLK_3",
11'b00100000000 "WT_BLK_4",
11'b01000000000 "WT_BLK_5",
11'b000010000000 "RD_WD_1" -color green,
11'b000100000000 "RD_WD_2" -color green,
-default hex
-default color 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 -default color 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'b0??00 "Write" -color orange,
  6'b1??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

The Global Signal Radix can be set from the Locals, Objects, Schematic, or Wave windows as follows:

1. Select a signal or group of signals.
2. 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 (Figure 6-4), 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 6-4. Setting the Global Signal Radix](image)

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.
GUI Customization

User-Defined Radices

Setting a Fixed Point Radix

Fixed point types are used in VHDL to represent non-integer numbers without using a floating point format. ModelSim automatically recognizes VHDL sfixed and ufixeda 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 6-4.
3. Click the **Custom Fixed/Float** selection to open the Fixed Point Radix dialog box (Figure 6-5).

![Figure 6-5. Fixed Point Radix Dialog Box](image)

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.
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 ModelSim GUI with the Preferences dialog box.

**Procedure**

1. Invoke the ModelSim 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, either in the registry of your Windows machine.

**Preferences Dialog Box**

**Tools > Edit Preferences**

Dialog box for controlling user-defined preferences of the ModelSim GUI.
GUI Preferences

Preferences Dialog Box

Figure 7-1. Preferences Dialog Box

 Fields

- **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
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 7-1. Default ListTranslateTable Values

<table>
<thead>
<tr>
<th>Mapped State</th>
<th>Bit Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOGIC_U</td>
<td>'U'</td>
</tr>
<tr>
<td>LOGIC_X</td>
<td>'X' 'x'</td>
</tr>
<tr>
<td>LOGIC_0</td>
<td>'0' FALSE</td>
</tr>
<tr>
<td>LOGIC_1</td>
<td>'1' TRUE</td>
</tr>
<tr>
<td>LOGIC_Z</td>
<td>'Z' 'z'</td>
</tr>
<tr>
<td>LOGIC_W</td>
<td>'W'</td>
</tr>
<tr>
<td>LOGIC_L</td>
<td>'L'</td>
</tr>
<tr>
<td>LOGIC_H</td>
<td>'H'</td>
</tr>
<tr>
<td>LOGIC_DC</td>
<td>'-'</td>
</tr>
</tbody>
</table>

Table 7-2. Default LogicStyleTable Values

<table>
<thead>
<tr>
<th>Mapped State</th>
<th>Line Style</th>
<th>Color</th>
<th>Row Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOGIC_U</td>
<td>Solid</td>
<td>red</td>
<td>1</td>
</tr>
<tr>
<td>LOGIC_X</td>
<td>Solid</td>
<td>red</td>
<td>1</td>
</tr>
<tr>
<td>LOGIC_0</td>
<td>Solid</td>
<td>green</td>
<td>0</td>
</tr>
<tr>
<td>LOGIC_1</td>
<td>Solid</td>
<td>green</td>
<td>2</td>
</tr>
<tr>
<td>LOGIC_Z</td>
<td>Solid</td>
<td>blue</td>
<td>1</td>
</tr>
<tr>
<td>LOGIC_W</td>
<td>DoubleDash</td>
<td>red</td>
<td>1</td>
</tr>
<tr>
<td>LOGIC_L</td>
<td>DoubleDash</td>
<td>grey90</td>
<td>0</td>
</tr>
<tr>
<td>LOGIC_H</td>
<td>DoubleDash</td>
<td>grey90</td>
<td>2</td>
</tr>
</tbody>
</table>
### Table 7-2. Default LogicStyleTable Values (cont.)

<table>
<thead>
<tr>
<th>Mapped State</th>
<th>Line Style</th>
<th>Color</th>
<th>Row Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOGIC_DC</td>
<td>DoubleDash</td>
<td>blue</td>
<td>1</td>
</tr>
</tbody>
</table>

**Figure 7-2. Modifying Signal Display Attributes in the Wave Window**
Index

— A —
Active Processes pane, 161
  <Italic>see also windows, Active Processes pane
Active window, selecting, 16
Add bookmark
  source window, 175
Analog sidebar, 197
Autofill text entry
  find, 23
Automatic command help, 188

— B —
base (radix)
  List window, 123
Bookmarks
  clear all in Source window, 175
  Source window, 175
break
  stop simulation run, 48, 55, 67, 79
Breakpoints
  command execution, 172
  conditional, 172
    use of SystemVerilog keyword this, 173
deleting, 170
edit, 170
in SystemVerilog class methods, 172
set with GUI, 169
Source window, viewing in, 165
Buttons
  user-defined, 218

— C —
Call Stack pane, 88
Clear bookmarks
  source window, 175
Clock change
  sampling signals at, 131
Clocking block inout display, 200

— D —
Dataflow window, 99
  <Italic>see also windows, Dataflow window
Deltas
  in List window, 128
descriptions of HDL items, 174
Design object icons, described, 33
DO file scripts
  creating from a saved transcript, 185
DO files
  creating from a saved transcript, 185

— E —
Edit
  breakpoints, 170
Editing
  in notepad windows, 207
  in the Main window, 207
  in the Source window, 207

Color
  radix
    example, 221
colorization, in Source window, 176
Colorize
  Transcript window, 186
Combine Selected Signals dialog box, 115
Combine signals, 126
Command completion, 188
compare, 126
Compare signal
  virtual, 126
Conditional breakpoints, 172
  use of SystemVerilog keyword this, 173
conditional breakpoints
  use of keyword this, 173
Contains, 22, 25, 26
Creating do file, 22, 126
Cursor linking, 198
Expanded Time
   viewing in List window, 116
Expression Builder
   configuring a List trigger with, 129
   saving expressions to Tcl variable, 122

— F —
F8 function key, 209
File-line breakpoints, 169
   edit, 170
Files window, 106
Filter, 25
Filtering
   Contains field, 22, 26
   signals in Objects window, 158
Find, 22
   in Structure window, 181
   inline search bar, 175, 187
   prefill text entry field, 23
Fixed point radix, 224
FocusFollowsMouse, 16
Fonts
   scaling, 22
Format
   saving/restoring, 22, 126
   signal
      Wave window, 197
Format file
   Wave window, 125
Function call, debugging, 88

— G —
Global signal radix, 157, 196, 222
Glob-style, 25

— H —
Help
   command help, 188
   highlighting, in Source window, 176
Highlights
   in Source window, 168

— I —
Icons
   shapes and meanings, 33
   inline search bar, 175, 187
   <Italic>see also pane, List pane

— K —
Keyboard shortcuts
   user defined, 204
keyboard shortcuts
   List window, 211
   Main window, 207
   Source window, 207
   Wave window, 211

— L —
Link cursors, 198
List pane
   <Italic>see also pane, List pane
List window, 111
   expanded time viewing, 116
   setting triggers, 129
Locals window, 132
   <Italic>see also windows, Locals window

— M —
Memories
   navigation, 138
   save to WLF file, 144
   saving formats, 143
   selecting memory instances, 143
   viewing contents, 143
   viewing multiple instances, 143
Memory tab
   memories you can view, 139
Menus
   user-defined, 218
Message Viewer tab, 146
Messages, 146
modelsim.tcl
   user_defined menus, 218
   user-defined buttons, 218
mouse shortcuts
   Main window, 207
   Source window, 207
   Wave window, 211
msgmode variable, 146

— N —
Nets
   Dataflow window, displaying in, 99
   values of
displaying in Objects window, 154
waveforms, viewing, 194
Notepad windows, text editing, 207
-notrigger argument, 131

— O —
Objects window, 154

— P —
Prefill text entry
find, 23
PrefMemory(ExpandPackedMem) variable, 143
PrefSource(OpenOnBreak) variable, 165
PrefSource(OpenOnFinish) variable, 165
PrefSource(OpenOnStep) variable, 165

— R —
Radix
change in Watch pane, 191
color
example, 221
List window, 123
set globally, 157, 196, 222
setting fixed point, 224
setting for Objects window, 157, 222
user-defined, 219
definition body, 220
radix
SystemVerilog types, 196
Radix define command
setting radix color, 221
Registers
values of
displaying in Objects window, 154
waveforms, viewing, 194
Regular-expression, 26
restart command
in GUI, 41
Restoring
window format, 22, 126

— S —
saveLines preference variable, 186
Saving
window format, 22, 126
Scaling fonts, 22
Search
in Structure window, 181
inline search bar
Source window, 175
Transcript, 187
prefill text entry field, 23
Search bar, 22
Setting radix
fixed point, 224
Shortcuts
text editing, 207
shortcuts
List window, 211
Main window, 207
Source window, 207
Wave window, 211
Signal format
Wave window, 197
Signal radix
for Objects window
Objects window
setting signal radix, 157, 222
set globally, 157, 196, 222
Signals
combine into bus, 126
Dataflow window, displaying in, 99
sampling at clock change, 131
types, selecting which to view, 158
values of
displaying in Objects window, 154
waveforms, viewing, 194
signals
Filtering in the Objects window, 158
Simulating
viewing results in List window, 111
source highlighting, customizing, 176
Source window, 164
clear highlights, 168
colorization, 176
disable automatic opening
$finish call, 165
on break, 165
single stepping, 165
inline search bar, 175
tab stops in, 176
see also windows, Source window
Status bar
Main window, 32
Structure window
find item, 181
syntax highlighting, 176
SystemVerilog, 173
class methods, conditional breakpoints, 172
SystemVerilog types
radix, 196
— T —
tab stops
Source window, 176
tabbed toolbars, 65
Text
filtering, 25
Text editing, 207
Toolbar
filter, 25
toolbar
Compile, 48
Dataflow, 48
Help, 49
Layout, 50
Memory, 50
Mode, 51
Objectfilter, 52
Process, 52
Schematic, 53
Simulate, 54
Source, 55
Standard, 56
Step, 58
tabs, 65
Wave, 59
Wave Cursor, 61
Wave Edit, 62
Wave Expand Time, 63
Zoom, 64
toolbar tabs, 65
Compile tab, 66
Debug tab, 67
Edit tab, 71
enabling, 65
Home tab, 73
Layout tab, 75
mapping to default toolbars, 83
Schematic and Dataflow tab, 76
Simulate tab, 78
user modification, 81
Transcript
colorize, 186
command help, 188
disable file creation, 187
inline search bar, 187
saving, 185
saving as a DO file, 185
Transcript window
changing buffer size, 186
changing line count, 186
Triggers
in List window, 127
Triggers, in the List window, 129
— U —
user_hook variable, 218
User-defined bus, 126
User-defined radix, 219
definition body, 220
— V —
Values
of HDL items, 174
Variables
displaying in Objects window, 154
Verilog
source code viewing, 164
VHDL
source code viewing, 164
viewing, 146
Viewing files for the simulation, 106
Virtual signal, 126
— W —
Wave window, 194
format signal, 197
saving layout, 125
see also windows, Wave window
Waveforms
viewing, 194
Window format
  saving/restoring, 22, 126

Windows
  Active Processes pane, 161
  Dataflow window, 99
  List window, 111
    display properties of, 123
    formatting HDL items, 123
    setting triggers, 127, 129
  Locals window, 132
  Main window
    status bar, 32
    time and delta display, 32
  Objects window, 154
  Signals window
    VHDL and Verilog items viewed in, 154
  Source window, 164
    viewing HDL source code, 164
  Variables window
    VHDL and Verilog items viewed in, 132
  Wave window, 194
    save format file, 125

windows
  Main window
    text editing, 207
  Source window
    text editing, 207

WLF file
  saving memories to, 144
  write format restart, 22, 126
END-USER LICENSE AGREEMENT ("Agreement")

This is a legal agreement concerning the use of Software (as defined in Section 2) and hardware (collectively “Products”) between the company acquiring the Products (“Customer”), and the Mentor Graphics entity that issued the corresponding quotation or, if no quotation was issued, the applicable local Mentor Graphics entity (“Mentor Graphics”). Except for license agreements related to the subject matter of this license agreement which are physically signed by Customer and an authorized representative of Mentor Graphics, this Agreement and the applicable quotation contain the parties’ entire understanding relating to the subject matter and supersede all prior or contemporaneous agreements. If Customer does not agree to these terms and conditions, promptly return or, in the case of Software received electronically, certify destruction of Software and all accompanying items within five days after receipt of Software and receive a full refund of any license fee paid.

1. ORDERS, FEES AND PAYMENT.

1.1. To the extent Customer (or if agreed by Mentor Graphics, Customer’s appointed third party buying agent) places and Mentor Graphics accepts purchase orders pursuant to this Agreement (each an “Order”), each Order will constitute a contract between Customer and Mentor Graphics, which shall be governed solely and exclusively by the terms and conditions of this Agreement, any applicable addenda and the applicable quotation, whether or not those documents are referenced on the Order. Any additional or conflicting terms and conditions appearing on an Order or presented in any electronic portal or automated order management system, whether or not required to be electronically accepted, will not be effective unless agreed in writing and physically signed by an authorized representative of Customer and Mentor Graphics.

1.2. Amounts invoiced will be paid, in the currency specified on the applicable invoice, within 30 days from the date of such invoice. Any past due invoices will be subject to the imposition of interest charges in the amount of one and one-half percent per month or the applicable legal rate currently in effect, whichever is lower. Prices do not include freight, insurance, customs duties, taxes or other similar charges, which Mentor Graphics will state separately in the applicable invoice. Unless timely provided with a valid certificate of exemption or other evidence that items are not taxable, Mentor Graphics will invoice Customer for all applicable taxes including, but not limited to, VAT, GST, sales tax, consumption tax and service tax. Customer will make all payments free and clear of, and without reduction for, any withholding or other taxes; any such taxes imposed on payments by Customer hereunder will be Customer’s sole responsibility. If Customer appoints a third party to place purchase orders and/or make payments on Customer’s behalf, Customer shall be liable for payment under Orders placed by such third party in the event of default.

1.3. All Products are delivered FCA factory (Incoterms 2010), freight prepaid and invoiced to Customer, except Software delivered electronically, which shall be deemed delivered when made available to Customer for download. Mentor Graphics retains a security interest in all Products delivered under this Agreement, to secure payment of the purchase price of such Products, and Customer agrees to sign any documents that Mentor Graphics determines to be necessary or convenient for use in filing or perfecting such security interest. Mentor Graphics’ delivery of Software by electronic means is subject to Customer’s provision of both a primary and an alternate e-mail address.

2. GRANT OF LICENSE. The software installed, downloaded, or otherwise acquired by Customer under this Agreement, including any updates, modifications, revisions, copies, documentation and design data (“Software”) are copyrighted, trade secret and confidential information of Mentor Graphics or its licensors, who maintain exclusive title to all Software and retain all rights not expressly granted by this Agreement. Mentor Graphics grants to Customer, subject to payment of applicable license fees, a nontransferable, nonexclusive license to use Software solely: (a) in machine-readable, object-code form (except as provided in Subsection 5.2); (b) for Customer’s internal business purposes; (c) for the term of the license; and (d) on the computer hardware and at the site authorized by Mentor Graphics. A site is restricted to a one-half mile (800 meter) radius. Customer may have Software temporarily used by an employee for telecommuting purposes from locations other than a Customer office, such as the employee’s residence, an airport or hotel, provided that such employee’s primary place of employment is the site where the Software is authorized for use. Mentor Graphics’ standard policies and programs, which vary depending on Software, license fees paid or services purchased, apply to the following: (a) relocation of Software; (b) use of Software, which may be limited, for example, to execution of a single session by a single user on the authorized hardware or for a restricted period of time (such limitations may be technically implemented through the use of authorization codes or similar devices); and (c) support services provided, including eligibility to receive telephone support, updates, modifications, and revisions. For the avoidance of doubt, if Customer provides any feedback or requests any change or enhancement to Products, whether in the course of receiving support or consulting services, evaluating Products, performing beta testing or otherwise, any inventions, product improvements, modifications or developments made by Mentor Graphics (at Mentor Graphics’ sole discretion) will be the exclusive property of Mentor Graphics.

3. ESC SOFTWARE. If Customer purchases a license to use development or prototyping tools of Mentor Graphics’ Embedded Software Channel (“ESC”), Mentor Graphics grants to Customer a nontransferable, nonexclusive license to reproduce and distribute executable
files created using ESC compilers, including the ESC run-time libraries distributed with ESC C and C++ compiler Software that are linked into a composite program as an integral part of Customer’s compiled computer program, provided that Customer distributes these files only in conjunction with Customer’s compiled computer program. Mentor Graphics does NOT grant Customer any right to duplicate, incorporate or embed copies of Mentor Graphics’ real-time operating systems or other embedded software products into Customer’s products or applications without first signing or otherwise agreeing to a separate agreement with Mentor Graphics for such purpose.

4. BETA CODE.

4.1. Portions or all of certain Software may contain code for experimental testing and evaluation (which may be either alpha or beta, collectively “Beta Code”), which may not be used without Mentor Graphics’ explicit authorization. Upon Mentor Graphics’ authorization, Mentor Graphics grants to Customer a temporary, nontransferable, nonexclusive license for experimental use to test and evaluate the Beta Code without charge for a limited period of time specified by Mentor Graphics. Mentor Graphics may choose, at its sole discretion, not to release Beta Code commercially in any form.

4.2. If Mentor Graphics authorizes Customer to use the Beta Code, Customer agrees to evaluate and test the Beta Code under normal conditions as directed by Mentor Graphics. Customer will contact Mentor Graphics periodically during Customer’s use of the Beta Code to discuss any malfunctions or suggested improvements. Upon completion of Customer’s evaluation and testing, Customer will send to Mentor Graphics a written evaluation of the Beta Code, including its strengths, weaknesses and recommended improvements.

4.3. Customer must maintain Beta Code in confidence and shall restrict access to the Beta Code, including the methods and concepts utilized therein, solely to those employees and Customer location(s) authorized by Mentor Graphics to perform beta testing. Customer agrees that any written evaluations and all inventions, product improvements, modifications or developments that Mentor Graphics conceived or made during or subsequent to this Agreement, including those based partly or wholly on Customer’s feedback, will be the exclusive property of Mentor Graphics. Mentor Graphics will have exclusive rights, title and interest in all such property. The provisions of this Subsection 4.3 shall survive termination of this Agreement.

5. RESTRICTIONS ON USE.

5.1. Customer may copy Software only as reasonably necessary to support the authorized use. Each copy must include all notices and legends embedded in Software and affixed to its medium and container as received from Mentor Graphics. All copies shall remain the property of Mentor Graphics or its licensors. Customer shall maintain a record of the number and primary location of all copies of Software, including copies merged with other software, and shall make those records available to Mentor Graphics upon request. Customer shall not make Products available in any form to any person other than Customer’s employees and on-site contractors, excluding Mentor Graphics contractors, whose job performance requires access and who are under obligations of confidentiality. Customer shall take appropriate action to protect the confidentiality of Products and ensure that any person permitted access does not disclose or use Products except as permitted by this Agreement. Customer shall give Mentor Graphics written notice of any unauthorized disclosure or use of the Products as soon as Customer becomes aware of such unauthorized disclosure or use. Except as otherwise permitted for purposes of interoperability as specified by applicable and mandatory local law, Customer shall not reverse-assemble, reverse-compile, reverse-engineer or in any way derive any source code from Software. Log files, data files, rule files and script files generated by or for the Software (collectively “Files”), including without limitation files containing Standard Verification Rule Format (“SVRF”) and Tcl Verification Format (“TVF”) which are Mentor Graphics’ trade secret and proprietary syntaxes for expressing process rules, constitute or include confidential information of Mentor Graphics. Customer may share Files with third parties, excluding Mentor Graphics competitors, provided that the confidentiality of such Files is protected by written agreement at least as well as Customer protects other information of a similar nature or importance, but in any case with at least reasonable care. Customer may use Files containing SVRF or TVF only with Mentor Graphics products. Under no circumstances shall Customer use Products or Files or allow their use for the purpose of developing, enhancing or marketing any product that is in any way competitive with Products, or disclose to any third party the results of, or information pertaining to, any benchmark.

5.2. If any Software or portions thereof are provided in source code form, Customer will use the source code only to correct software errors and enhance or modify the Software for the authorized use. Customer shall not disclose or permit disclosure of source code, in whole or in part, including any of its methods or concepts, to anyone except Customer’s employees or on-site contractors, excluding Mentor Graphics competitors, with a need to know. Customer shall not copy or compile source code in any manner except to support this authorized use.

5.3. Customer may not assign this Agreement or the rights and duties under it, or relocate, sublicense, or otherwise transfer the Products, whether by operation of law or otherwise (“Attempted Transfer”), without Mentor Graphics’ prior written consent and payment of Mentor Graphics’ then-current applicable relocation and/or transfer fees. Any Attempted Transfer without Mentor Graphics’ prior written consent shall be a material breach of this Agreement and may, at Mentor Graphics’ option, result in the immediate termination of the Agreement and/or the licenses granted under this Agreement. The terms of this Agreement, including without limitation the licensing and assignment provisions, shall be binding upon Customer’s permitted successors in interest and assigns.

5.4. The provisions of this Section 5 shall survive the termination of this Agreement.

6. SUPPORT SERVICES. To the extent Customer purchases support services, Mentor Graphics will provide Customer with updates and technical support for the Products, at the Customer site(s) for which support is purchased, in accordance with Mentor Graphics’ then current End-User Support Terms located at http://supportnet.mentor.com/supportterms.

7. LIMITED WARRANTY.

7.1. Mentor Graphics warrants that during the warranty period its standard, generally supported Products, when properly installed, will substantially conform to the functional specifications set forth in the applicable user manual. Mentor Graphics does not warrant that Products will meet Customer’s requirements or that operation of Products will be uninterrupted or error free. The
warranty period is 90 days starting on the 15th day after delivery or upon installation, whichever first occurs. Customer must notify Mentor Graphics in writing of any nonconformity within the warranty period. For the avoidance of doubt, this warranty applies only to the initial shipment of Software under an Order and does not renew or reset, for example, with the delivery of (a) Software updates and (b) authorization codes or alternate Software under a transaction involving Software re-mix. This warranty shall not be valid if Products have been subject to misuse, unauthorized modification, improper installation or Customer is not in compliance with this Agreement. MENTOR GRAPHICS’ ENTIRE LIABILITY AND CUSTOMER’S EXCLUSIVE REMEDY SHALL BE, AT MENTOR GRAPHICS’ OPTION, EITHER (A) REFUND OF THE PRICE PAID UPON RETURN OF THE PRODUCTS TO MENTOR GRAPHICS OR (B) MODIFICATION OR REPLACEMENT OF THE PRODUCTS THAT DO NOT MEET THIS LIMITED WARRANTY. MENTOR GRAPHICS MAKES NO WARRANTIES WITH RESPECT TO: (A) SERVICES; (B) PRODUCTS PROVIDED AT NO CHARGE; OR (C) BETA CODE; ALL OF WHICH ARE PROVIDED “AS IS.”

7. LIMITATION OF LIABILITY. EXCEPT WHERE THIS EXCLUSION OR RESTRICTION OF LIABILITY WOULD BE VOID OR INEFFECTIVE UNDER APPLICABLE LAW, IN NO EVENT SHALL MENTOR GRAPHICS OR ITS LICENSORS BE LIABLE FOR INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES (INCLUDING LOST PROFITS OR SAVINGS) WHETHER BASED ON CONTRACT, TORT OR ANY OTHER LEGAL THEORY, EVEN IF MENTOR GRAPHICS OR ITS LICENSORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL MENTOR GRAPHICS’ OR ITS LICENSORS’ LIABILITY UNDER THIS AGREEMENT EXCEED THE AMOUNT RECEIVED FROM CUSTOMER FOR THE HARDWARE, SOFTWARE LICENSE OR SERVICE GIVING RISE TO THE CLAIM. IN THE CASE WHERE NO AMOUNT WAS PAID, MENTOR GRAPHICS AND ITS LICENSORS SHALL HAVE NO LIABILITY FOR ANY DAMAGES WHATSOEVER. THE PROVISIONS OF THIS SECTION 7 SHALL SURVIVE THE TERMINATION OF THIS AGREEMENT.

8. HAZARDOUS APPLICATIONS. CUSTOMER ACKNOWLEDGES IT IS SOLELY RESPONSIBLE FOR TESTING ITS PRODUCTS USED IN APPLICATIONS WHERE THE FAILURE OR INACCURACY OF ITS PRODUCTS MIGHT RESULT IN DEATH OR PERSONAL INJURY (“HAZARDOUS APPLICATIONS”). EXCEPT TO THE EXTENT THIS EXCLUSION OR RESTRICTION OF LIABILITY WOULD BE VOID OR INEFFECTIVE UNDER APPLICABLE LAW, IN NO EVENT SHALL MENTOR GRAPHICS OR ITS LICENSORS BE LIABLE FOR INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES (INCLUDING LOST PROFITS OR SAVINGS) WHETHER BASED ON CONTRACT, TORT OR ANY OTHER LEGAL THEORY, EVEN IF MENTOR GRAPHICS OR ITS LICENSORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL MENTOR GRAPHICS’ OR ITS LICENSORS’ LIABILITY UNDER THIS AGREEMENT EXCEED THE AMOUNT RECEIVED FROM CUSTOMER FOR THE HARDWARE, SOFTWARE LICENSE OR SERVICE GIVING RISE TO THE CLAIM. IN THE CASE WHERE NO AMOUNT WAS PAID, MENTOR GRAPHICS AND ITS LICENSORS SHALL HAVE NO LIABILITY FOR ANY DAMAGES WHATSOEVER. THE PROVISIONS OF THIS SECTION 8 SHALL SURVIVE THE TERMINATION OF THIS AGREEMENT.

9. INDEMNIFICATION. CUSTOMER AGREES TO INDEMNIFY AND HOLD HARMLESS MENTOR GRAPHICS AND ITS LICENSORS FROM ANY CLAIMS, LOSS, COST, DAMAGE, EXPENSE OR LIABILITY, INCLUDING ATTORNEYS’ FEES, ARISING OUT OF OR IN CONNECTION WITH THE USE OF MENTOR GRAPHICS PRODUCTS IN OR FOR HAZARDOUS APPLICATIONS. THE PROVISIONS OF THIS SECTION 9 SHALL SURVIVE THE TERMINATION OF THIS AGREEMENT.

10. INFRINGEMENT. CUSTOMER AGREES TO INDEMNIFY AND HOLD HARMLESS MENTOR GRAPHICS AND ITS LICENSORS FROM ANY CLAIMS, LOSS, COST, DAMAGE, EXPENSE OR LIABILITY, INCLUDING ATTORNEYS’ FEES, ARISING OUT OF OR IN CONNECTION WITH THE USE OF MENTOR GRAPHICS PRODUCTS IN OR FOR HAZARDOUS APPLICATIONS. THE PROVISIONS OF THIS SECTION 10 SHALL SURVIVE THE TERMINATION OF THIS AGREEMENT.

11. TERMINATION AND EFFECT OF TERMINATION.

11.1. Mentor Graphics may terminate this Agreement immediately upon written notice if Customer: (a) exceeds the scope of the license or otherwise fails to comply with the licensing or confidentiality provisions of this Agreement, or (b) becomes insolvent, files a bankruptcy petition, institutes proceedings for liquidation or winding up or enters into an agreement to assign its assets for the benefit of creditors. For any other material breach of any
provision of this Agreement, Mentor Graphics may terminate this Agreement and/or any license granted under this Agreement upon 30 days written notice if Customer fails to cure the breach within the 30 day notice period. Termination of this Agreement or any license granted hereunder will not affect Customer’s obligation to pay for Products shipped or licenses granted prior to the termination, which amounts shall be payable immediately upon the date of termination.

12.2. Upon termination of this Agreement, the rights and obligations of the parties shall cease except as expressly set forth in this Agreement. Upon termination, Customer shall ensure that all use of the affected Products ceases, and shall return hardware and either return to Mentor Graphics or destroy Software in Customer’s possession, including all copies and documentation, and certify in writing to Mentor Graphics within ten business days of the termination date that Customer no longer possesses any of the affected Products or copies of Software in any form.

13. EXPORT. The Products provided hereunder are subject to regulation by local laws and United States (“U.S.”) government agencies, which prohibit export, re-export or diversion of certain products, information about the products, and direct or indirect products thereof, to certain countries and certain persons. Customer agrees that it will not export or re-export Products in any manner without first obtaining all necessary approval from appropriate local and U.S. government agencies. If Customer wishes to disclose any information to Mentor Graphics that is subject to any U.S. or other applicable export restrictions, including without limitation the U.S. International Traffic in Arms Regulations (ITAR) or special controls under the Export Administration Regulations (EAR), Customer will notify Mentor Graphics personnel, in advance of each instance of disclosure, that such information is subject to such export restrictions.

14. U.S. GOVERNMENT LICENSE RIGHTS. Software was developed entirely at private expense. The parties agree that all Software is commercial computer software within the meaning of the applicable acquisition regulations. Accordingly, pursuant to U.S. FAR 48 CFR 12.212 and DFAR 48 CFR 227.7202, use, duplication and disclosure of the Software by or for the U.S. government or a U.S. government subcontractor is subject solely to the terms and conditions set forth in this Agreement, which shall supersede any conflicting terms or conditions in any government order document, except for provisions which are contrary to applicable mandatory federal laws.

15. THIRD PARTY BENEFICIARY. Mentor Graphics Corporation, Mentor Graphics (Ireland) Limited, Microsoft Corporation and other licensors may be third party beneficiaries of this Agreement with the right to enforce the obligations set forth herein.

16. REVIEW OF LICENSE USAGE. Customer will monitor the access to and use of Software. With prior written notice and during Customer’s normal business hours, Mentor Graphics may engage an internationally recognized accounting firm to review Customer’s software monitoring system and records deemed relevant by the internationally recognized accounting firm to confirm Customer’s compliance with the terms of this Agreement or U.S. or other local export laws. Such review may include FlexNet (or successor product) report log files that Customer shall capture and provide at Mentor Graphics’ request. Customer shall make records available in electronic format and shall fully cooperate with data gathering to support the license review. Mentor Graphics shall bear the expense of any such review unless a material non-compliance is revealed. Mentor Graphics shall treat as confidential information all information gained as a result of any request or review and shall only use or disclose such information as required by law or to enforce its rights under this Agreement. The provisions of this Section 16 shall survive the termination of this Agreement.

17. CONTROLLING LAW, JURISDICTION AND DISPUTE RESOLUTION. The owners of certain Mentor Graphics intellectual property licensed under this Agreement are located in Ireland and the U.S. To promote consistency around the world, disputes shall be resolved as follows: excluding conflict of laws rules, this Agreement shall be governed by and construed under the laws of the State of Oregon, U.S., if Customer is located in North or South America, and the laws of Ireland if Customer is located outside of North or South America. All disputes arising out of or in relation to this Agreement shall be submitted to the exclusive jurisdiction of the courts of Portland, Oregon when the laws of Oregon apply, or Dublin, Ireland when the laws of Ireland apply. Notwithstanding the foregoing, all disputes in Asia arising out of or in relation to this Agreement shall be resolved by arbitration in Singapore before a single arbitrator to be appointed by the chairman of the Singapore International Arbitration Centre (“SIAC”) to be conducted in the English language, in accordance with the Arbitration Rules of the SIAC in effect at the time of the dispute, which rules are deemed to be incorporated by reference in this section. Nothing in this section shall restrict Mentor Graphics’ right to bring an action (including for example a motion for injunctive relief) against Customer in the jurisdiction where Customer’s place of business is located. The United Nations Convention on Contracts for the International Sale of Goods does not apply to this Agreement.

18. SEVERABILITY. If any provision of this Agreement is held by a court of competent jurisdiction to be void, invalid, unenforceable or illegal, such provision shall be severed from this Agreement and the remaining provisions will remain in full force and effect.

19. MISCELLANEOUS. This Agreement contains the parties’ entire understanding relating to its subject matter and supersedes all prior or contemporaneous agreements. Some Software may contain code distributed under a third party license agreement that may provide additional rights to Customer. Please see the applicable Software documentation for details. This Agreement may only be modified in writing, signed by an authorized representative of each party. Waiver of terms or excuse of breach must be in writing and shall not constitute subsequent consent, waiver or excuse.

Rev. 140201, Part No. 258976