

Introduction

PatternSequenceTool defines the execution sequence of a pattern or set of patterns and identifies which `WaveformTable` objects are used with the executed patterns. This tool lets you group the `Pattern` objects, waveform definitions, and timing expressions into a single, executable pattern set; refer to [Features](#). Thus, you can run a pattern with one set of timing or waveforms in one test and then re-run the pattern with a different set of timing or waveforms in another test.

At the pattern language level, this tool calls the `PatternSequence` object out of a `Test` object, which defines the pattern execution for a Microflow Method or Test; refer to [PatternSequence Object](#).

This enVision release also supports the `DomainSet` object.

Features

You can change the sequence of patterns without editing the pattern file by using this tool for the following tasks:

- Define or modify the execution sequence of a single test pattern for debugging, characterizing, and developing test programs.
- Control the pattern execution mode.
- Mask fails, generate triggers, and control special pattern parameters.
- Insert an initialization sequence before each pattern in the sequence.

- Define the waveforms used with the pattern set.

PatternSequence Object

Patterns in a test program are run from one pattern directly into the next pattern, or the patterns are linked by a control program generated by the Pattern Loader. A `PatternSequence` object specifies the execution sequence of patterns and the type of `WaveformTables` associated with them. With this tool, you can run a pattern with one set of timing or waveforms in one `Test` object and then re-run it with a modified set of timing or waveforms in another `Test` object.

The `PatternSequence` object manages two distinct entities:

- Thread—a collection of patterns and other threads executed sequentially as specified by the user. [Figure 29.1](#) shows the relationship between the Threads and the enVision software tools. A pattern sequence may have one or more threads of execution. Their order of definition is the order of their execution on the tester. With `PatternSequenceTool`, you can define the execution control properties of the threads:
 - Specific [entry](#) and [exit](#) points in the patterns, such as the first and last vector in the pattern.
 - Portions of the patterns to [ignore failures](#) upon execution.
 - Specific points in the patterns to [trigger](#) external events.

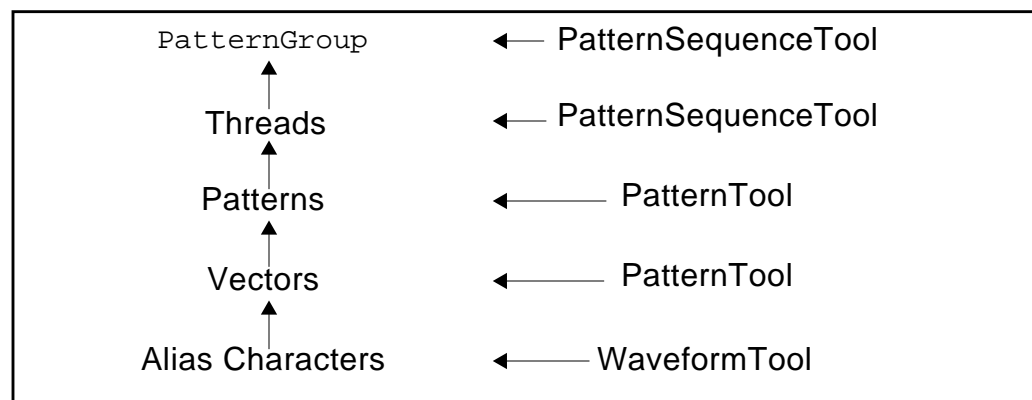


Figure 29.1: Hierarchy of Alias Characters, Vectors, Patterns, Threads and PatternGroup

- **WaveformTable timing**—part of the `PatternSequence` object that lists the `WaveformTables` that define all aliases referenced in the patterns as `WaveformTable References`.

A `WaveformTable` object may contain the definitions for all pins and all alias characters, or a subset of them. LTX recommends the partitioning of the pins and aliases into subsets, so you can reuse common waveforms across multiple `WaveformTable References`, thereby reducing code redundancy and the time to define the waveforms.

The second major function of the `PatternSequence` object is the `Zipper Table`, which resolves the list of `WaveformTable References` from each vector of all patterns in the pattern sequence to a `WaveformTable` object; refer to the [WaveformTables Pane](#).

For more information about this object, refer to the *Patterns, Waveforms, and Timing* chapter in the *enVision Digital Programming* manual.

PatternSequenceTool Windows, Menus, and Dialogs

The main window of `PatternSequenceTool` contains [menus](#) and [panes with controls](#) for viewing and modifying the `Threads` of a pattern and the `WaveformTables` and `WaveformTable References` associated with a pattern; see [Figure 29.2](#) and refer to [Table 29.1](#). It includes a [Thread Editor](#).

Certain menu items use [dialogs](#).

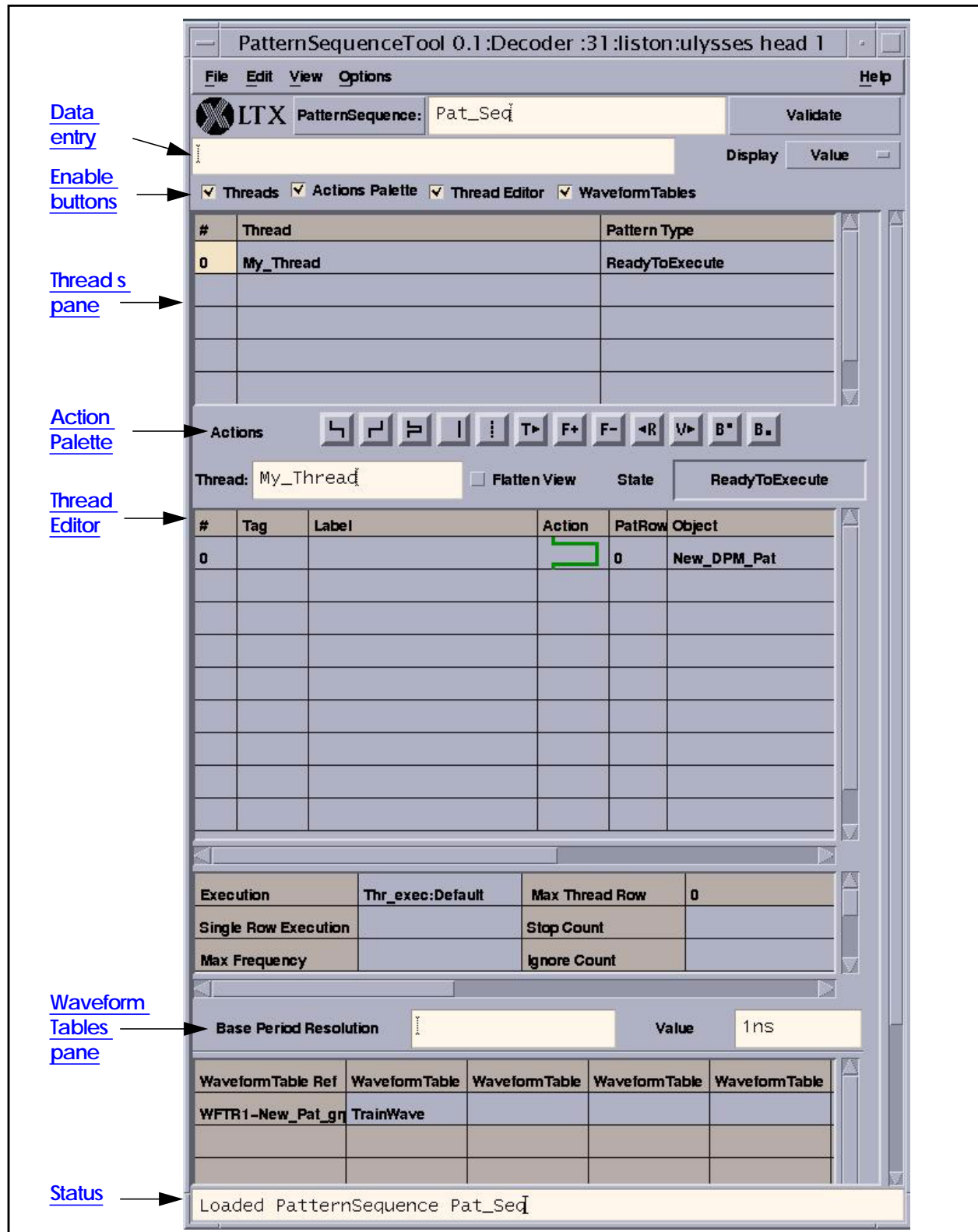


Figure 29.2: PatternSequenceTool—Main Window

Table 29.1: PatternSequenceTool Panes and Text Fields, Main Window

Item	Meaning
Threads (enable button)	Click to display the Threads Pane .
Actions palette (enable button)	Click to display the Actions Palette .
Thread Editor (enable button)	Click to display the Thread Editor Pane .
WaveformTables (enable button)	Click to display the WaveformTables Pane .
PatternSequence	Same as Find in File menu; refer to Finder Object Dialog .
Validate	Click to validate certain aspects of the PatternSequence object and to resolve the relevant expression in the timing specified in the WaveformTables. Validation errors are output to an ASCII file in the directory you launched enVision. If no validation errors, no error file is created.
Display	Selects how the contents of each cell in the tool working area is displayed: Value—resolved value for the expressions in each cell of the selected site. Expression—actual expression in each cell for the selected site. MultiValue—all values in each cell for all enabled sites, independent of the selected site. MultiRange—lowest and highest values in each cell, separated by ellipses, for all enabled sites, independent of the selected site
Threads pane	Shows available Threads; refer to Threads Pane .
Actions palette	Refer to Actions Palette .
Thread Editor	Refer to Thread Editor Pane .
WaveformTables pane	Refer to WaveformTables Pane .
Status	Displays tool and command status and error messages.

PatternSequenceTool Menus

The PatternSequenceTool toolbar consists of the following menus:

- File—[Table 29.2](#)
- Edit—[Table 29.3](#)
- View—[Site Selection Dialog](#) (one option in menu)
- Options—[Table 29.4](#)
- Help—[Table 29.5](#)

Table 29.2: PatternSequenceTool—File Menu

Selection	Action
Find	Same as pressing PatternSequence button on main window; refer to Finder Object Dialog .
Delete	Deletes current PatternSequence object. Dialog appears, prompting you to confirm the deletion.
Clone	Clears the name of PatternSequence object in the text field next to the LTX logo and copies the current PatternSequence object. Enter a new name in the PatternSequence text field for the new object and press Return.
Print	Prints the window contents; refer to the <i>User Interface Tools</i> chapter in this manual for printer options.
Close	Saves any changes and closes PatternSequenceTool.

Table 29.3: PatternSequenceTool—Edit Menu

Selection	Action
Pattern Sequence	<p>Comments Opens dialog for entering comments about the <code>PatternSequence</code> object. After entering all comments, click OK to save the comments.</p>
Base Period Resolution	<p>Sets the tester period resolution by changing the base period of the system.</p> <p>Base period is from 1 ns to 1.142857 ns on the VX IV tester. Default base period is 1 ns, meaning the tester is accurate to the integer period values.</p> <p>Another way to specify the base period is to enter a period value equal to the accuracy the tester must be accurate to. For example, you can have multiple <code>WaveformTables</code>, each with a different period value greater than the minimum programmable device cycle.</p>
Reload Waveforms	<p>Opens dialog prompting you to reload the waveforms (timing) at the next Test execution or Validate command. Click OK to reload in hardware the <code>WaveformTables</code> and waveform cells associated with the current <code>PatternSequence</code> object.</p> <p>Reloading may not be necessary; however, if you want to make sure the timing and waveform definitions are loaded, use this command.</p>
Tester Resource Report	<p>Opens Tester Resource Report dialog.</p>
Force Waveform Compile	<p>Sets an internal flag informing the tester to recompile the patterns invoked by the current <code>PatternSequence</code> object the next time the patterns are executed.</p>
Pattern Reload	<p>Opens a dialog, prompting you to reload the patterns at next Test execution or Validate command. Click OK to reload in hardware the patterns referenced in the current <code>PatternSequence</code> object.</p> <p>Reloading may not be necessary; however, if you want to make sure the patterns are loaded, use this command.</p>

Table 29.3: PatternSequenceTool—Edit Menu (Continued)

Selection	Action
Thread	Comments Opens a dialog for entering Thread comments. After entering all comments, click OK to save the comments.
	Pattern Reload Opens Pattern Reload dialog, prompting you to reload only the patterns in the Thread at the next Test execution or Validate command. Click OK to reload in hardware the patterns in the Thread referenced in current PatternSequence object. Reloading may not be necessary; however, if you want to make sure the patterns are loaded, use this command.

Table 29.4: PatternSequenceTool—Options Menu

Selection	Action
Xdefaults	Changes fonts and colors in PatternSequenceTool display; refer to Setting the PatternSequenceTool .Xdefaults . Note <code>cellColor1</code> through <code>cellColor10</code> , which define color of alias characters, are based on the color set in tool. By default, these colors should match. If you change fonts, the <code>patternFont</code> and <code>headerSignal</code> fonts have matching bold versions. They must match in both height and width, or bolding will not function properly.
Save Geometry	Saves screen position and file size of tool to a file in user’s home directory: <code>\$HOME/app-defaults/PatternSequenceTool</code> . File is read each time the tool starts.

Table 29.5: PatternSequenceTool—Help Menu

Selection	Action
On PatternSequenceTool	Opens PDF file of this chapter. Chapter is part of the manual listed in the top right-hand corner of the page; refer to Index option under Help.
Index	Opens linked PDF file that lists all customer documentation/manuals for this tester. Open a manual by clicking on its name.
On Version	Opens Tool Version pop-up that lists information about the installed enVision software release.

PatternSequenceTool Dialogs

PatternSequenceTool has the following dialogs:

- [Finder Object Dialog](#)
- [Tester Resource Report](#)
- [Site Selection Dialog](#)
- [Label Expression Builder](#)
- [Resync WaveformTables Dialog](#)
- [Mark Ranges Dialog](#)

Finder Object Dialog

1. Selecting Find in the File menu or selecting Find from a context-sensitive menu opens a dialog for choosing a PatternSequence object from a list of PatternSequence objects already created. It is the same as pressing the PatternSequence button next to the LTX logo; see [Figure 29.3](#).
2. To load an object, double click on it or type in a new name for this object in the Selection field and click OK.

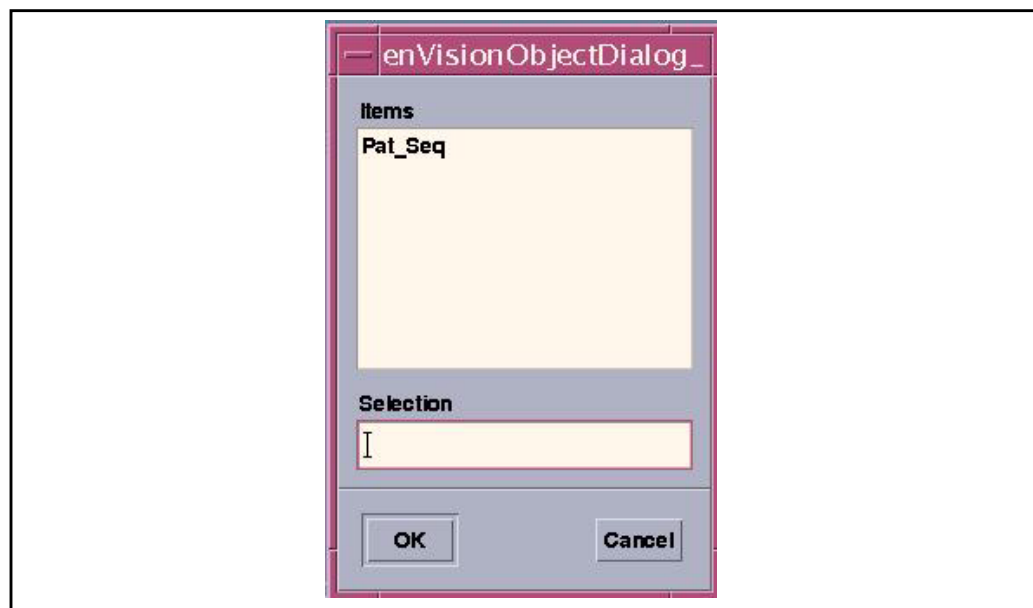


Figure 29.3: Find Object Dialog

Tester Resource Report

1. Selecting Tester Resource Report from the Edit menu opens a Tester Resource Report dialog.
2. In the Tester Resource Report dialog, click Resource Pins to open the Pin Finder dialog.
3. In the Pin Finder dialog:
 - a. Select the pin groups or pins used by the current `PatternSequence` object to appear in the report. You can also enter this information in the text field next to the Report Pins button in the Tester Resource Report dialog. If you do not select any pins, the report includes all pins referenced in the WaveformTables and the patterns.
 - b. Click OK to create a report file in the directory in which you launched enVision.
4. ErrorTool appears, stating an Advisory message with the report status or path of the generated file, or both. Generated file is ASCII; refer to [Sample Tester Resource Report](#).

Sample Tester Resource Report

The report includes the following information:

- Pin or Pin Group, or both
- WaveformTable references
- PatternGroup
- Alias
- Waveform cells (`WFCells`)
- Waveform sets (drive and compare)
- Marker values for Waveform sets

Report includes information about the hardware resources at 3 levels:

- Thread—provides the highest level of detail, showing the attributes of each pattern, followed by a Thread summary.

- **Pattern Sequence**—summarizes how the current PatternSequence object is used.
- **Test Program**—shows a test program summary.

Definitions of some terms in the three reports:

- **Pattern Type**—Cpm, Dpm, Apg, KAL (Keep Alive) and Base.
- **Mode**—is blank for normal mode, x2 for DVM mode, and FullMux for Mux mode patterns.
- **Cycle Count**—only shown for DPM pattern types. Due to the nature of CPM (looping, conditional jumps), CPM counts are not calculated. These counts are displayed so you can use this information for writing control patterns.

The following listing is a partial report:

```
Waveform Compile Report for Pattern Sequence DC_pats.
Tue Dec 10 13:40:28 1996

Timing Report for Pin/s Clk
Timeset Assignments for pin Clk

  WFT Ref      Pattern Group      Alias      WFCell      LTS
  -----      -
  _0           Z86DC_pats           c/C        DC.DC_6     0

Timeset marker values for pin Clk

LTS   DT1   DT2           DT3   DT4           CT1   CT2   DPeriod   CPeriod
----   ---   ---           --    ---           ---   ---   -
0 RZ  60.62nS 107.24nS ANRZ  14nS           18nS  AF 74.62nS 78.62nS 140nS 140nS

** Hardware Stop on Vector is installed and will work for all pattern types! **
Thread           __EV_Thread_72 - Type = UserControlPatterns
Thread           Simple_functional - Type = UserControlPatterns
Thread           Bus_tristated - Type = UserControlPatterns
Thread           Bus_low - Type = UserControlPatterns
Thread           Bus_high - Type = UserControlPatterns
Thread           Dummy_pat - Type = UserControlPatterns
Thread           Continuous_mode - Type = UserControlPatterns

Waveform Memory Information for Pattern Sequence DC_pats

  Number of WTM locations used is 7, total for program = 13
  Number of PLTS locations used is 1, total for program = 1

Pattern Usage Report for Pattern Sequence DC_pats

THREAD REPORT - Name: "Bus_HiZ", Execution Type: "UserControlPatterns"
```

Pattern Name	Memory Type / Mode	Num Of PLTS, WTM	Cycle Count	Vector Count	Frm Load Address	To Load Address
Z86DC	Cpm	1, 7	* 0	452	4	455

* Due to the nature of CPM actual cycle count are not calculated.

Total Cpm vectors used: 452
 Total Dpm vectors used: 0
 Number of WTM locations used: 7
 Number of PLTS used: 1

Site Selection Dialog

1. Selecting the only option in the View menu opens the Site Selection dialog.
2. In the Site Selection dialog, select a particular site as the active site to display or as the locked site to display; see [Figure 29.4](#).

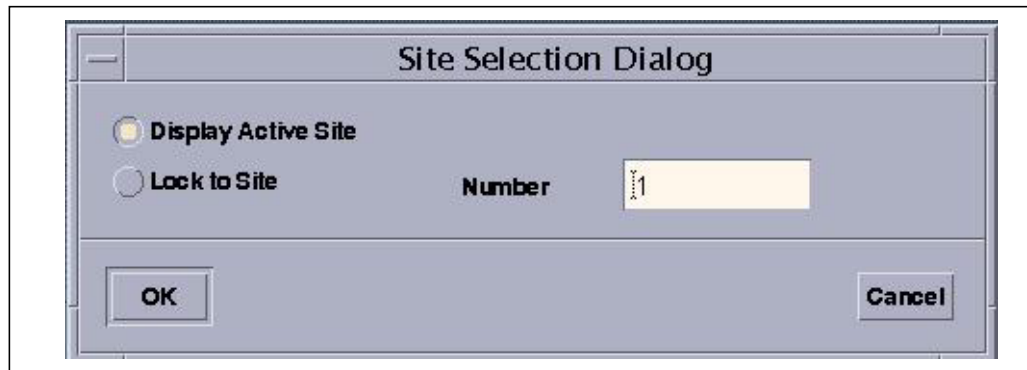


Figure 29.4: Site Selection Dialog

3. If your application has more than one site, use BinTool to select one of them as the active site because only one site can be displayed at a time; refer to the Show Selected Site and Show Displayable options in the *BinTool* chapter in this manual.
4. Choose one of the following:
 - a. Display the active site in PatternSequenceTool by clicking Display Active Site and then clicking OK.
 - b. Lock the site entered in Number field as the site to display in PatternSequenceTool, independent of Show Displayable setting in BinTool, by clicking Lock to Site and then clicking OK.

Label Expression Builder

Use this dialog to create a label expression; see [Figure 29.5](#). Form a label expression by specifying the Pattern Group, Pattern and Label names in the text fields listed in [Table 29.6](#). Because these fields act like filters you must enter information in the following order: PatternGroup, Pattern, and Label.

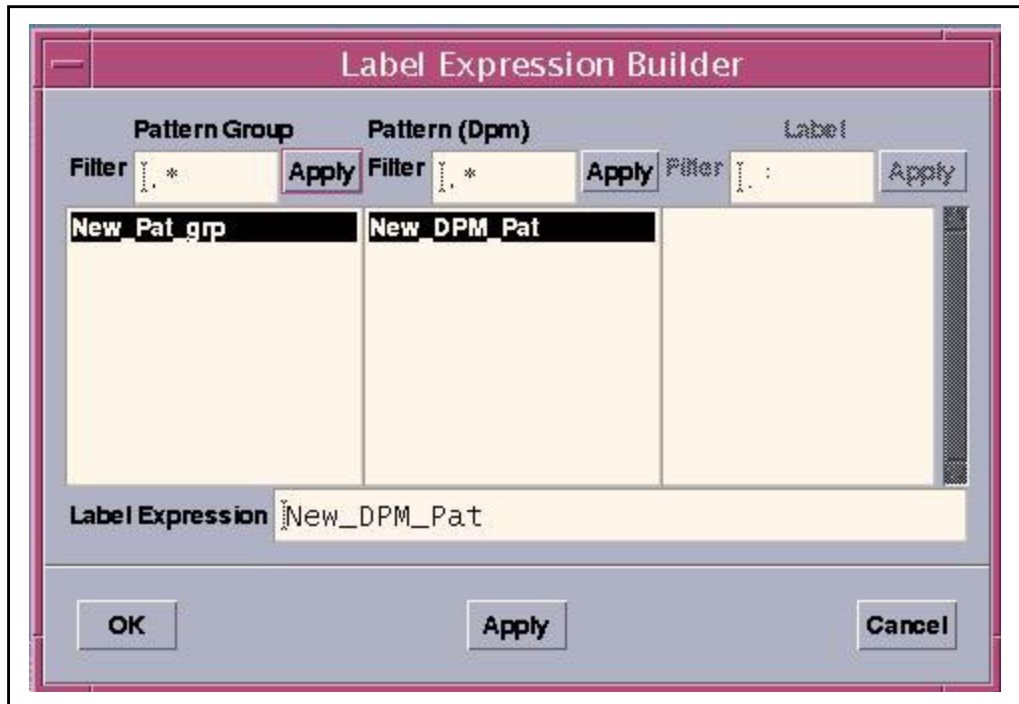


Figure 29.5: Label Expression Builder Dialog

Table 29.6: Label Expression Builder

Selection	Action
PatternGroup	Select one of the pattern set defined in the PatternGroup statement.
Pattern	Select one of the patterns defined in the pattern files and grouped under the selected PatternGroup.
Label	Select a label included in one of the selected patterns.

Resync WaveformTables Dialog

Use this dialog to resynchronize new aliases for a specified PinGroup with the associated WaveformTable. This dialog will cause the alias for the specified pins, which do not exist in the WaveformTable, to be added with the default values suggested by the PatternGroup:

1. Clicking mouse button M3 in the Waveform Table column in the Waveform Table pane opens the Resync WaveformTables dialog; see [Figure 29.6](#) and refer to [Table 29.7](#).
2. Enter the required PinGroup and Aliases. After you complete this dialog, enVision reprocesses the selected WaveformTable and creates any alias data added to the new pattern file for the specified pins and aliases.
3. In WaveformTool, add the waveform and timing definitions to complete this operation.

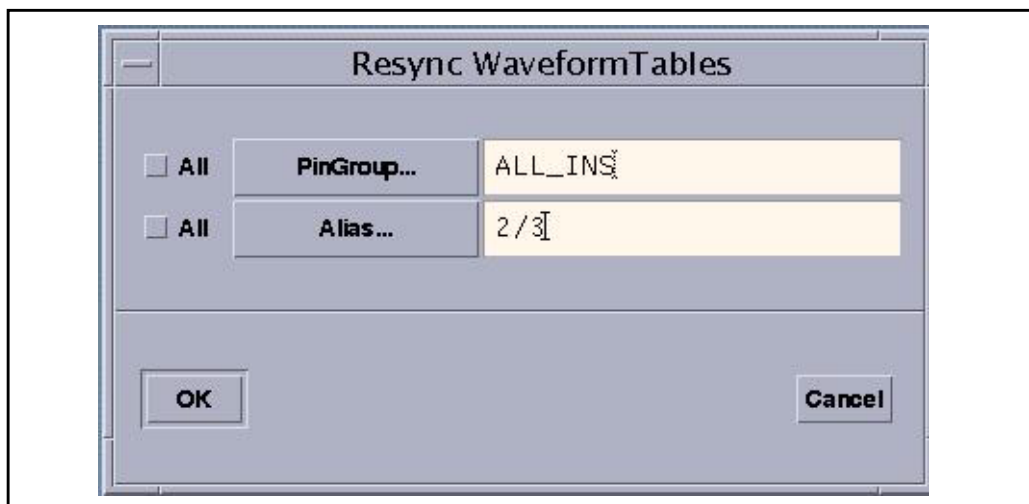


Figure 29.6: Resync WaveformTables Dialog

Table 29.7: Resync WaveformTables Dialog

Item	Selection
All (PinGroup)	Selects all pins used by the PatternGroup to be included in the Waveform Table being re-synchronized; otherwise, use PinGroup to select individual PinGroups. Selecting All de-selects PinGroup.
PinGroup	Click to open a Pin Finder dialog. From this dialog, select the pins to be included in the WaveformTable.
All (Alias)	Selects all aliases associated with the selected pins to be included in the WaveformTable being re-synchronized; otherwise, use PinGroup to select individual aliases. Selecting All de-selects Alias.
Alias	Opens Alias Characters dialog. From this dialog, select the aliases to be included in the WaveformTable.

Table 29.7: Resync WaveformTables Dialog (Continued)

Item	Selection
OK	Exits this dialog and instructs enVision to reprocess the listed Waveform Table, which creates any new alias data added to a new pattern file for the specified pins and aliases.

Mark Ranges Dialog

Selecting View Marks from the context-sensitive Rows menu in the Thread Editor opens the Mark Ranges dialog; see [Figure 29.7](#) and refer to [Table 29.8](#).



Figure 29.7: Marks Range Dialog

Table 29.8: Marks Range Dialog

Item	Selection
Begin Mark	Enter the starting Thread index number for a marked range.
End Mark	Enter the ending Thread index number for a marked range.
OK	Clicking button cause PatternSequenceTool to insert begin (<) and end (>) marks in the # columns of the Active Thread in the Thread Editor.

PatternSequenceTool Panes

PatternSequenceTool panes consists of text fields and windows for [entering](#) and displaying the data for Threads and WaveformTable References. The following panes are displayed or hidden by the [enable](#) buttons:

- [Threads Pane](#)
- [Actions Palette](#)
- [Thread Editor Pane](#)
- [WaveformTables Pane](#)

Entering Data in PatternSequenceTool

In PatternSequenceTool you enter all data in the text field below the LTX logo except for the PatternSequence object name, Base Period Resolution and Value, and Thread name. Be sure to press the Return key after typing in the data; otherwise, PatternSequenceTool, like the other tools, will not store the information.

Most columns in the PatternSequenceTool panes have context-sensitive pop-ups that are opened by clicking the mouse button M3; refer to the individual panes for details.

Threads Pane

This pane shows the available Threads, including the Thread number (`Pattern_index` argument to several Methods), Thread name, keyword description of the Thread's execution status, and pattern type. You can also use this pane to create or delete Threads; see [Figure 29.8](#) and refer to [Table 29.9](#).

#	Thread	Pattern Type
0	My_Thread	ReadyToExecute
1	Simple_functional	ReadyToExecute
2	Bus_tristated	ReadyToExecute
3	Bus_low	ReadyToExecute
4	Bus_high	ReadyToExecute

Figure 29.8: Pattern Sequence Thread Block

Table 29.9: PatternSequenceTool—Thread Pane

Item	Meaning
# (Thread Number)	<p>Displays the number of the Threads in the loaded PatternSequence object. In this read-only column the Active Thread is in reverse video. It is the thread last selected in a pattern execution; also refer to Externally Referencing the Active Thread.</p> <p>You can select the Thread to edit in the Thread Editor by double clicking its name.</p>
Thread	<p>Displays the Thread name in the loaded PatternSequence object.</p> <p>Double clicking on its name makes it the Active Thread and causes it to appear in the Thread Editor Pane. The selected Thread and its associated Thread number will be highlighted.</p> <p>If an icon is displayed in this column, the Thread is anonymous, otherwise the name of the Thread is shown.</p> <p>If the cell is empty, it contains no Thread object exists.</p> <p>In this column mouse button M3 supports various tool operations; refer Context-Sensitive Active Menu in Thread Pane.</p>

Table 29.9: PatternSequenceTool—Thread Pane (Continued)

Item	Meaning
Pattern Type	<p>Read-only column displays the Thread type:</p> <p>DataPatterns—Thread contains only in-line Data Pattern Type (DPM), meaning the EnterExit Action is not used on any thread row. No Base pattern is used.</p> <p>UserControlPatterns—Thread is controlled by a Control Pattern Type (CPM) pattern specified by the programmer. DPMs are specified as Ref Action.</p> <p>MaskedDataPatterns—Thread contains DPM patterns with Enable Fails and Disable Fails Actions, which forces the pattern system to process the patterns uniquely.</p> <p>GeneratedControlPattern—Thread is controlled by a CPM pattern generated by the system. In this mode, you may specify one or more Base Type (Base) patterns to keep the DUT in the proper state, while these auto-generated vectors are being executed.</p> <p>NotExecutable—Thread cannot be executed because it is anonymous or was not correctly defined.</p> <p>ReadyToExecute—Thread not yet processed, but can be executed.</p>

Context-Sensitive Active Menu in Thread Pane

With the cursor in the Thread column, clicking mouse button M3 opens the context-sensitive Object menu of PatternSequenceTool operations you can perform on the Active Thread; refer to [Table 29.10](#).

Table 29.10: Thread Pane—Context-Sensitive Object Menu

Item	Selection
Find	Same as Find in File menu; refer to Table 29.2 .
Examine	Objects only: spawns appropriate tool for viewing or editing selected object. Same as double-clicking the icon representing the object.
Delete	Same as Delete in File menu except that it applies to selected Thread object.
Clone	Same as Clone in File menu except that it applies to selected Thread object.
Cut	Copies selected cell contents to paste buffer and clears cell.
Copy	Copies selected cell contents to paste buffer.
Paste	Copies data from paste buffer into selected cell.

Table 29.10: Thread Pane—Context-Sensitive Object Menu (Continued)

Item	Selection
Append	Appends a row below the selected Thread row.
Usage	Opens Threads Ancestor dialog, reporting direct Ancestors to selected object. If a Thread has multiple ancestors, a triangle appears in top left corner of row number cell.
Comment	Same as Comment in Edit menu; refer to Table 29.3 .
Pattern Reload	Same as Pattern Reload in Edit menu, Pattern section; refer to Table 29.3 .

Actions Palette

This section of PatternSequenceTool shows the Actions you can assign to the Threads; see [Figure 29.9](#) and refer to [Table 29.11](#):

1. To select the desired Action, click on the desired radio button in the Actions palette. If no buttons in the Action palette are selected, this feature is turned off.

NOTE If you select an invalid action, no icon appears and the background color of the column changes to indicate a warning state.

2. In the Thread Editor, position the cursor over Action column for the desired Thread and click mouse button M1. Pressing M1 causes the Action to change into the selected icon.

NOTE You can also change the Actions in the Thread Editor by using the Action item; refer to [Table 29.12](#).

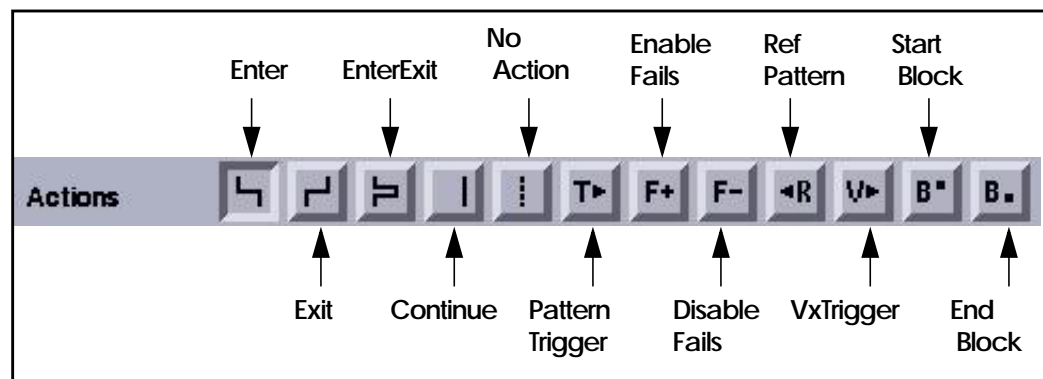


Figure 29.9: Action Palette

Table 29.11: Action Palette Icons













Action	Meaning
Enter 	Identifies entry point into pattern Thread: where vector execution starts. It is defined as the entry point in the flattened view of a Thread, which shows any nested Thread expanded into its components. By default, first pattern vector is entry point into pattern, but any location can be specified with a Label .
Exit 	Identifies exit point for pattern Thread: where vector execution stops. It is defined as exit point in the flattened view of a Thread, which shows any other nested Thread expanded into its components. By default, last vector in pattern is the pattern exit, but any location can be specified with a Label .
EnterExit 	Identifies entry and exit points of a pattern execution, mainly for hierarchical Threads. Entry and exit points are assumed to be the beginning and ending of a pattern if used with a <code>Pattern</code> object. When specifying this action with a label, a single vector execution is resolved. Or, if this action is specified with a <code>Pattern</code> object, it specifies the execution of the entire <code>Pattern</code> object defined in the pattern file.
Continue 	Executes contiguous vectors in the <code>Pattern</code> object for the entire Thread row. Default for every <code>Pattern</code> object.
No Action 	Forces no execution of the vectors in the <code>Pattern</code> object referenced by the Thread row. Useful for debugging, so execution of failing patterns can be turned off.
Pattern Trigger 	Identifies point in Thread row to generate a Pattern Trigger, which may be used in measuring frequency and CTMU timing.
Enable Fails 	Identifies point in Thread row to enable pattern fail in constant, runtime invariant, or runtime variant applications. It is the inverse of Ignore Fail. Since multiple enables or disables can be applied to a single Thread, this action can mask pattern fails on any desired segments.
Disable Fails 	Identifies point in Thread row to disable pattern fail in constant, runtime invariant, or runtime variant applications. It is the same as Ignore Fail.

Table 29.11: Action Palette Icons (Continued)

Action	Meaning
Reference Pattern 	Identifies a pattern used by the Thread-specified patterns executed only by micro-instructions. PatternSequence object requires this information to load and resolve all references from Pattern objects called by micro-instructions. With this Action, you can use a CPM pattern to control the execution of DPM patterns; refer to Base pattern type in the <i>Patterns, Waveforms, and Timing</i> chapter in the <i>enVision Digital Programming</i> manual.
VxTrigger 	Identifies point in Thread row to generate a trigger sent to the Interface module. Can trigger the Device Power Supply current monitor, which characterizes fluctuations in a dynamic power supply over a long pattern execution. If this Action is not used, the specified trigger is at the end of the pattern, so power supply current measurements are backward compatible with other LTX testers.
Start Block 	Identifies point in a pattern where a block starting point is defined. Use this Action to define Serial blocking when executing Serial mode patterns. These blocks are effectively Continue actions, but they instruct the interpreter to support the Serial Mode. These blocks create a Pat Row for a Thread row.
End Block 	Identifies point in a pattern as a block ending point.

Thread Editor Pane

This section of the tool is for editing the active Thread; see [Figure 29.10](#) and refer to [Table 29.12](#).

Certain items have a context-sensitive menu; refer to [Table 29.12](#) for more information.

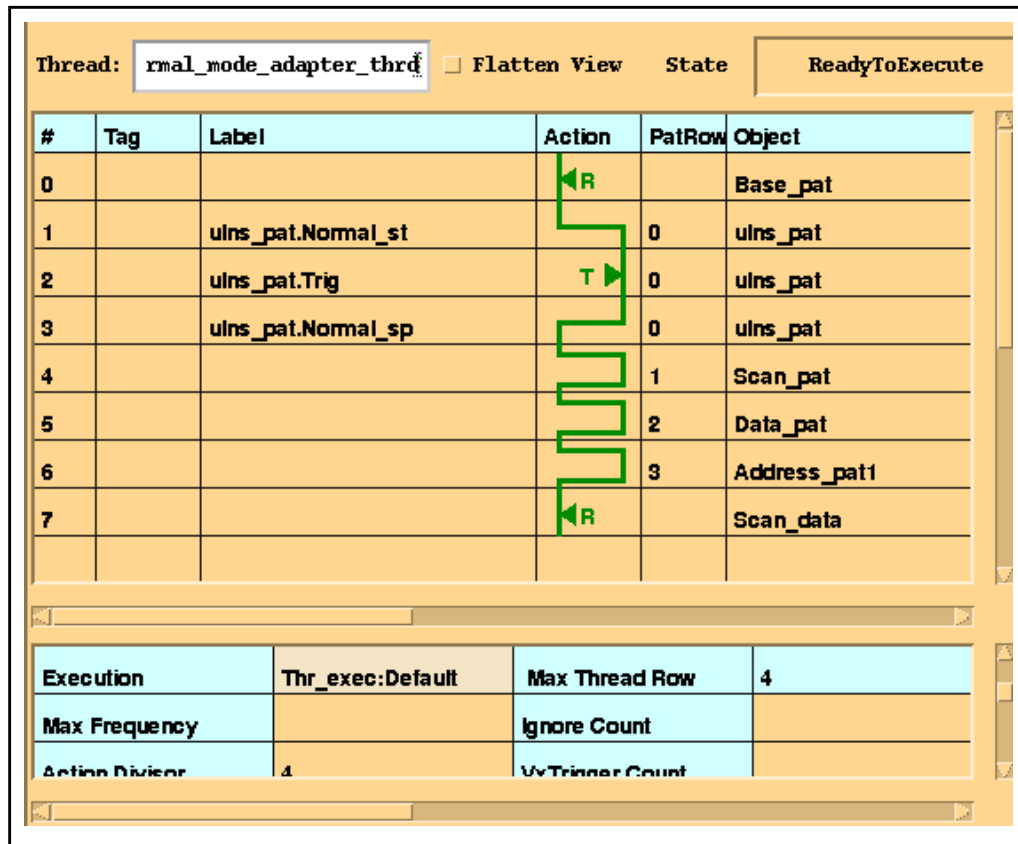


Figure 29.10: Thread Editor Pane

Table 29.12: PatternSequenceTool—Thread Editor

Item	Selection
Thread	<p>Text field displays the name of the viewed Thread. Editing the name here also changes the name of the object. You can also rename the Thread in the Threads pane by selecting the desired Thread and retyping the name; refer to Threads Pane.</p> <p>To view another object, you can also select the desired Thread by using the Threads pane.</p>
Flatten View	<p>Indicates viewing mode for currently-loaded Thread. If this button is set, all nested Threads are displayed in the flatten view, which expands any contained Threads. Expanded Threads are read-only and appear in the read-only (border) color.</p> <p>A Thread remains in the Flatten View state when it is set, but only for that Thread: it does not expand references to this Thread in other Threads.</p> <p>If Flatten View is not selected, all Thread references in the Object column are displayed as a single row.</p>
State	Text field indicates state of active Thread; refer to pattern type .

Table 29.12: PatternSequenceTool—Thread Editor (Continued)

Item	Selection
# (row number)	<p>Read-only column displays number of rows in currently-loaded Thread. If active Thread contains other Threads, number of rows may be different between the normal view and the Flatten View.</p> <p>Clicking M3 in this column opens the context-sensitive Rows menu; refer to Table 29.14.</p>
Tag	<p>Optional name identifies the row externally to the Thread object. Valid tag name:</p> <p><code>Thread.Column.Tag</code></p> <p>where:</p> <p>Thread is the name of the Thread;</p> <p>Column is either Tag, Label, Action, or Object;</p> <p>Tag is the tag name you enter.</p> <p>You can also address the Thread row by using the following index operator:</p> <p><code>Thread.Column[<i>index</i>]</code></p> <p>enVision supports Expression lookup of any external references to the Thread row by the user-defined tag. Thus, any Thread row can be named and directly addressed as an expression, independent of the row position in the Thread sequence.</p> <p>Clicking M3 in this column opens the context-sensitive Object menu; refer to Table 29.13.</p>
Label	<p>Use Labels to enter the additional information for the Action to complete a definition of a row, such as identifying a specific point within a pattern.</p> <p>Labels are normally described by the following syntax:</p> <p><code>PatternObject.Label</code></p> <p>Vector offsets are represented by an integer with an optional plus or minus sign preceding it. Sample Labels:</p> <p><code>MyPat.Label_1</code></p> <p><code>MyPat.Label_2 + 30</code></p> <p>Clicking M3 in this column opens the context-sensitive Object menu; refer to Table 29.13.</p>
Action	<p>Shows the selected Action for the row. If Display mode is Expression, this column shows the Action name or the expression that programmed the Action. Or, if Value is the Display mode, Actions appear as a waveform graphic of the execution stream. Reference Patterns Actions are indicated by a letter R, while Trigger Patterns are indicated by a letter T; refer to Figure 29.10.</p> <p>Clicking M3 in this column opens the context-sensitive Actions palette menu, like the one shown in Figure 29.9.</p>

Table 29.12: PatternSequenceTool—Thread Editor (Continued)

Item	Selection
Pat Row	<p>Displays the row number of the Thread, a group of vectors always executed together. In the normal mode, all rows are executed in one burst; however, in the Serial mode, each Thread row is executed separately.</p> <p>Clicking M3 in this column opens the context-sensitive Object menu; refer to Table 29.13.</p>
Object	<p>Displays the name of the object related to the specified Action. Patterns and Threads (themselves a list of <code>Pattern</code> objects) are the only objects displayed here.</p> <p>You can use the Label Expression Builder to create labels here.</p> <p>Clicking M3 in this column opens the context-sensitive Object menu; refer to Table 29.13.</p>
Object site <i>n</i>	<p>This column appears only for multisite test programs, while the Object column represents the pattern loaded for the first site.</p> <p>Use this column for sites other than the first one. These columns accept other objects, so different patterns can be executed in parallel on different sites. These entries are optional. If these fields for other sites are blank, the object specified in the Object column is replicated to the remaining sites.</p> <p>It is designed for multisite testing of sequential matches. Each site is matched separately, while the other sites keep the device running in a Keep-Alive mode. Figure 29.11 shows one method of dividing a pattern into 4 separate <code>Pattern</code> objects on a 4-site application.</p> <p>Clicking M3 in this column opens the context-sensitive Object menu; refer to Table 29.13.</p>
Execution	<p>Specifies the execution mode for the active Thread.</p> <p>When you create a Thread, it contains rows of executable segments (not the same as Thread Editor rows) that are assembled into a seamless pattern execution, which is named Default in this field.</p> <p>Edit this column by typing in the mode or click M3 while the cursor is in the column and select the mode. You can change the execution to Serial, so each row is executed separately; the result is multiple datalog events. Or, you can select SerialOnFail for execution in Default mode, followed by Serial, only if the Default execution fails. Or, select Expr, which opens ExpressionTool.</p>
Single Row Execution	<p>Executes only a single row of a Thread if its value is between 0 and the Max Thread Row.</p> <p>Clicking M3 in this column opens the context-sensitive Expression menu; refer to Table 29.13.</p>

Table 29.12: PatternSequenceTool—Thread Editor (Continued)

Item	Selection
Max Frequency	<p>To run a pattern in the Dual Vector Mode or other enhanced operating modes, you must specify a frequency greater than the maximum frequency of the normal operating mode.</p> <p>Clicking M3 in this column opens the context-sensitive Expression menu; refer to Table 29.13.</p>
Action Divisor	<p>ActionBus option only: enter the divisor for the ActionBus hardware. Default is 4, which supports the CPM (Action Bus is connected to CPM and has a small set of CPM micro-instructions) to be run at 125 MHz (125 MHz / 4 = 31.25 MHz).</p> <p>To set the ActionBus at a rate slower than 31.25 MHz, which improves the micro-code resolution, change the divider value.</p> <p>Clicking M3 in this column opens the context-sensitive Expression menu; refer to Table 29.13.</p>
Max Thread Row	<p>Read-only field displays maximum row number used by Thread.</p> <p>Clicking M3 in this column opens the context-sensitive Expression menu; refer to Table 29.13.</p>
Stop Count	<p>Even though the Thread contains all required information to determine the stop conditions, you may want to reuse the Threads with different Stop Counts by using the Stop Count argument. For example, you can vary the value for this integer expression value under control of an Axis object, which provides characterization data, such as Dynamic IDD versus Stop Count.</p> <p>Clicking M3 in this column opens the context-sensitive Expression menu; refer to Table 29.13.</p>
Ignore Count	<p>Enter the integer expression to ignore fails from the first executed vector to the specified count. This is one method of masking strobos. You can also mask strobos at any place in the pattern by using the Enable Fails and Disable Fails Actions.</p> <p>Clicking M3 in this column opens the context-sensitive Expression menu; refer to Table 29.13.</p>
VxTrigger Count	<p>Even though the Thread contains all required information to determine the VxTrigger conditions, you may want to reuse the Threads with different VxTrigger conditions by using the VxTrigger Count argument. For example, For example, you can vary the value for this integer expression value under control of an Axis object, which provides characterization data, such as Dynamic IDD versus VxTrigger Count.</p> <p>Clicking M3 in this column opens the context-sensitive Expression menu; refer to Table 29.13.</p>

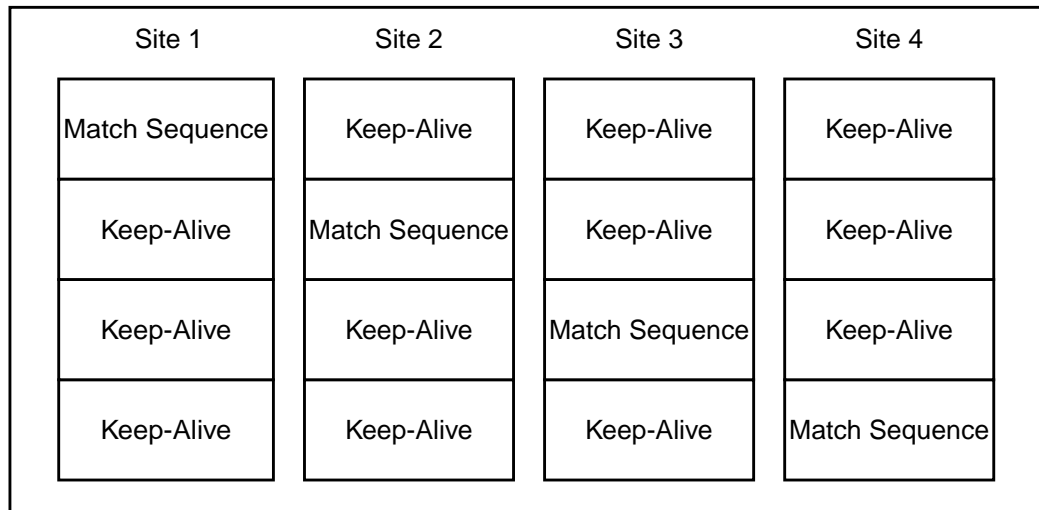


Figure 29.11: Example: Dividing Pattern Objects for MultiState Match Mode

Context-Sensitive Menus in Thread Editor Pane

One of the following context-sensitive menus opens when you click mouse button M3 over a Thread Editor column:

- Tab, Label, Pat Row, and Object columns: refer to Object menu, [Table 29.13](#).
- # (Thread number) column; refer to Row menu, [Table 29.14](#).
- Single Row Execution, Max Frequency, Max Thread Row, Stop, and Ignore Count columns; refer to Expression menu, [Table 29.15](#).

Table 29.13: Thread Editor—Context-Sensitive Object Menu

Item	Selection
Find Thread	Opens Finder Object Dialog to find Thread object.
Find Pattern	Opens Label Expression Builder to find Pattern object.
Examine	Spawns appropriate tool for viewing or editing selected object. Same as double-clicking icon representing the object.
Cut Row	Copies selected row to paste buffer and removes row.
Copy Row	Copies selected row contents to paste buffer.
Paste Row	Copies data from paste buffer into selected row.
Insert Row	Inserts a row at the cursor location of the selected Thread.

Table 29.13: Thread Editor—Context-Sensitive Object Menu (Continued)

Item	Selection
Usage	Opens Threads Ancestor dialog that reports the direct Ancestors to the selected object. If a Thread has multiple ancestors, a triangle appears in the top left corner of the row number cell.
Pattern Reload	Same as Pattern Reload option in Edit menu, Pattern section; refer to Table 29.3 .

Table 29.14: Thread Editor—Context-Sensitive Rows Menu

Item	Selection
Set Mark	Sets the beginning of a range of rows.
End Mark	Sets the end of a range of rows.
View Marks	Opens the Mark Ranges Dialog .
Cut Range	Copies all rows between the set and end mark points to the paste buffer and deletes these rows.
Copy Range	Copies the range data from paste buffer to the selected row.
Paste Range	Copies all rows between the set and end mark points to the paste buffer and deletes the rows.
Delete Range	Deletes all rows between the set mark and end mark points.
Reload Range	
Clear Reload Range	

Table 29.15: Thread Editor—Context-Sensitive Expression Menu

Item	Selection
Cut	Copies selected expression to paste buffer and clears column.
Copy	Copies selected expression to paste buffer.
Paste	Copies data in paste buffer to selected column
Expr	Opens ExpressionTool; refer to the <i>ExpressionTool</i> chapter in this manual.

WaveformTables Pane

Use this section, also known as the *Zipper Table*, to connect the WaveformTable References in the pattern file to the waveform and

timing information entered in WaveformTool for the WaveformTable definitions; see [Figure 29.12](#) and refer to [Table 29.16](#).

Every vector in a pattern file must contain a WaveformTable Reference that points to a row in Zipper Table. This row is referenced by a vector that identifies one or more WaveformTables that define the format and timing for the alias characters in that vector.

At the pattern language level, the Zipper Table resolves the list of waveform references from each vector of all patterns in the pattern sequence to a WaveformTable object. The WaveformTable object may be directly associated with the Pattern object or connected via the PatternSequence object. The connection via the PatternSequence data object is the most flexible because you can reuse the pattern set in another sequence with a different set of format and timing relationships.

You can open the context-sensitive Zipper Table menu by clicking mouse button M3 in a WaveformTable column; refer to [Table 29.17](#).

Waveform Table References are described in the *Patterns, Waveforms, and Timing* chapter in the *enVision Digital Programming* manual.

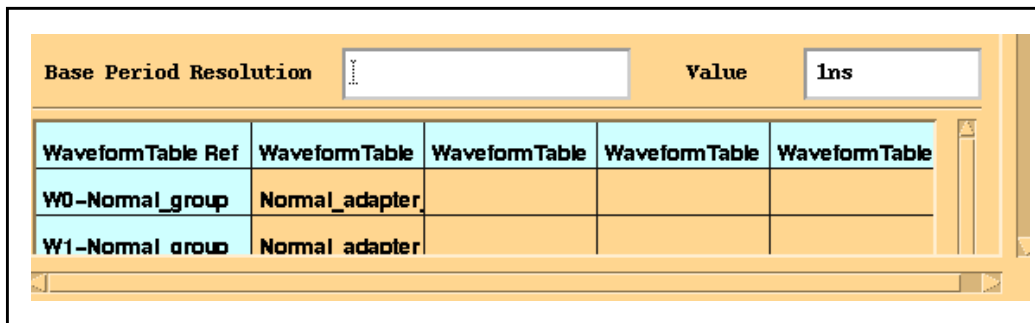


Figure 29.12: Waveform Table Pane

Table 29.16: PatternSequenceTool—Waveform Table Pane

Item	Selection
Base Period Resolution	Refer to Base Period Resolution in Edit menu of main window.
Value	Resolved value from Base Period Resolution expression.

Table 29.16: PatternSequenceTool—Waveform Table Pane (Continued)

Item	Selection
Waveform Table Ref	<p>Read-only list of all pattern vector references by all patterns in the <code>PatternSequence</code> object. When you enter or delete a pattern from the sequence list, the Waveform Table Reference list is updated to display the new patterns. Once these row titles appear, you can define Waveform Tables to resolve the formatting and timing for all aliases used by all pins in the <code>PatternGroup</code> object.</p> <p>All references use the following syntax: <code>Waveform_table_reference-pattern_group</code>.</p>
Waveform Table	<p>Displays the names of <code>WaveformTable</code> objects created with <code>WaveformTool</code>. Waveform Tables specified in these rows resolve the timing information in the specified Waveform Table Ref column.</p> <p>You can enter the Waveform Table name in this column or use the context-sensitive menu by clicking mouse button M3 while the cursor is in this column.</p> <p>Multiple Waveform Tables can resolve a Waveform Table Reference in a row, where each Waveform Table defines a subset of the pins or aliases. Thus, you can re-use a Waveform Table where the definitions are the same, and create a different Waveform Table for a unique waveform.</p>

Table 29.17: WaveformTable Pane—Context-Sensitive Zipper Table Menu

Item	Selection
Find	Opens the <code>enVision_object</code> popup for selecting a <code>WaveformTable</code> . After selecting the object, click OK.
Examine	Opens <code>WaveformTool</code> for the selected <code>WaveformTable</code> object.
Resync	Opens the Resync WaveformTables Dialog .
Cut	Copies selected cell contents to paste buffer and clears cell.
Copy	Copies selected cell contents to paste buffer.
Paste	Copies data from paste buffer into selected cell.
Add Column	Adds a column to the right of the existing columns in the WaveformTable pane.

Sample Applications

The first example shows how to group three DPM patterns and perform error masking in some blocks of the pattern. The second example uses the WaveformTables section to partition definitions across multiple WaveformTables.

Assume the first example must execute two `Pattern` objects (Z8600 and Z86DC) and mask fails in Z86DC in a certain portion of the pattern, which is executed twice; see [Figure 29.13](#). Consequently, you use the Thread Editor to create:

- pattern Thread to start with the Z8600 object
- F- Action on a Z86DC label
- F+ Action on another Z86DC label, Z8600 object
- F- Action on a Z86DC label
- F+ Action on another Z86DC label
- End with Z86DC object.

#	Thread	Pattern Type		
0	Group_pat	ReadyToExecute		
1	Data_patterns	DataPatterns		
Thread: <input type="text" value="Group_pat"/> <input type="checkbox"/> Flatten View State <input type="text" value="ReadyToExecute"/>				
#	Tag	Label	Action	Pattern Object
0				Z8600
1		Z86DC.BUSLEAK	F-	Z86DC
2		Z86DC.BUSVOL	F+	Z86DC
3				Z8600
4		Z86DC.S0VOH	F-	Z86DC
5		Z86DC.BUSVOH	F+	Z86DC
6		Z86DC.BUSVOL		Z86DC

Figure 29.13: Pattern Sequence Thread Example

In the second example, the alias definitions for one of the WaveformTable references is divided into three tables: input pins, output pins, and bidirectional pins. This method does not examine the contents of the WaveformTables themselves, but you use PatternSequenceTool to combines these separate tables. One of the references is resolved when you add the three tables in one row. Finally, you must open WaveformTool to flush them; see [Figure 29.14](#) and [Table 29.15](#)

WaveformTable Ref	WaveformTable	WaveformTable	WaveformTable
_0-Z86DC_pats	Input_timing	Output_timing	Bidirectional_timing

Figure 29.14: Sample PatternSequence Waveform

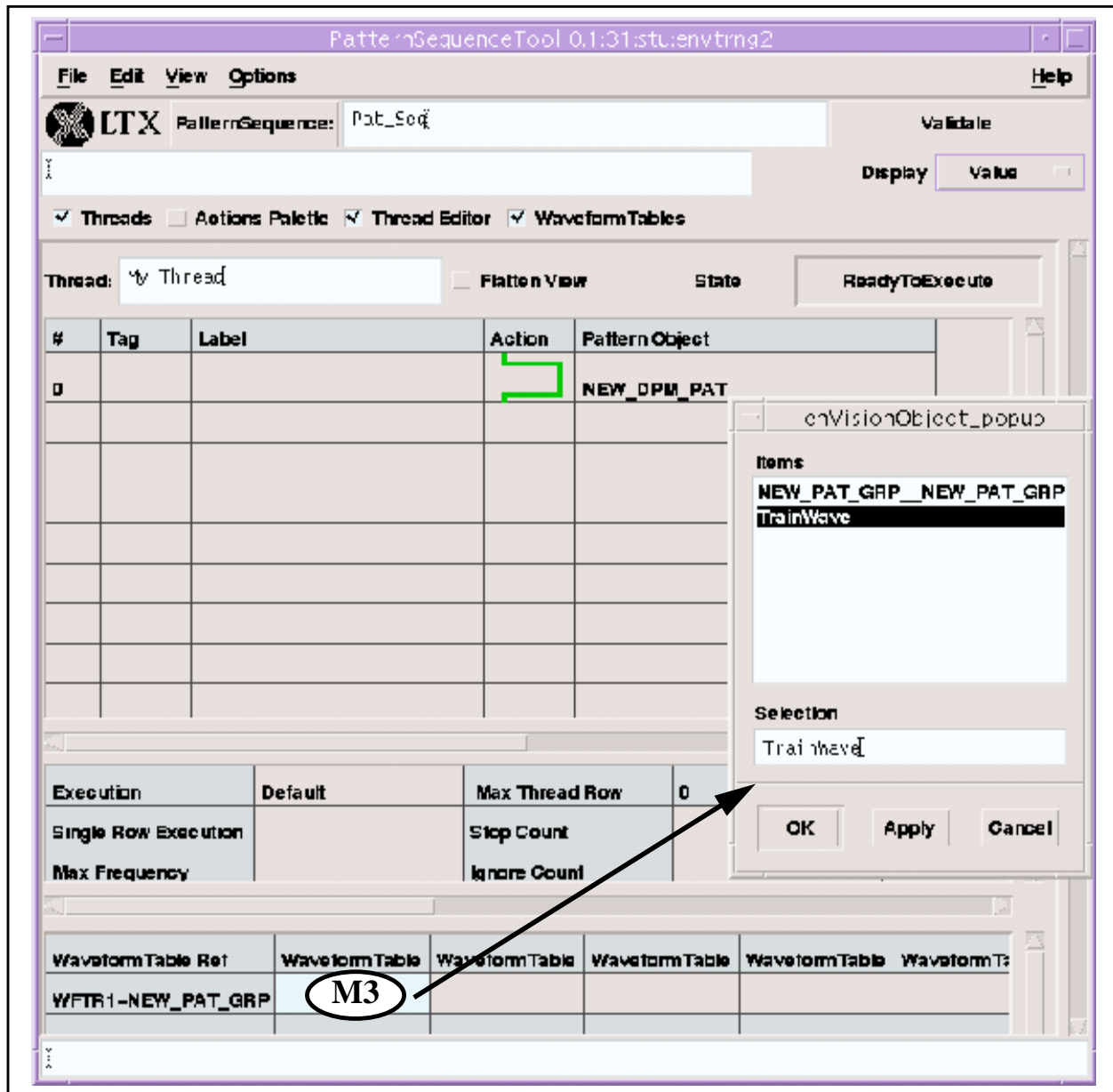


Figure 29.15: Adding WaveformTable to WaveformTable Reference in Pattern Sequence

PatternSequenceTool Tasks

You can modify the pattern sequence or operate and manage PatternSequenceTool by:

- [Setting the PatternSequenceTool .Xdefaults](#)

- [Externally Referencing the Active Thread](#)
- [Using the Mouse for PatternTool Operations](#)
- [Using Keyboard Shortcuts for PatternSequenceTool Operations](#)

Setting the PatternSequenceTool .Xdefaults

The following list are the .Xdefaults for this tool, with the default values:

```

PatternSequenceTool*textColor:           #000000
PatternSequenceTool*selectColor:        #F3E4C5
PatternSequenceTool*borderColor:        #111111
PatternSequenceTool*actionErrorColor:   #FFFFB9
PatternSequenceTool*enableFailColor:    #008B00
PatternSequenceTool*disableFailColor:   #B60042
PatternSequenceTool*readOnlyColor:      #cffffff
PatternSequenceTool*xdefFont:           variable
PatternSequenceTool*textFont:          variable
PatternSequenceTool*tableWindowHeight:  250
PatternSequenceTool*editorWindowHeight: 400
PatternSequenceTool*paletteWindowHeight: 44
PatternSequenceTool*threadsWindowHeight: 160
  
```

Externally Referencing the Active Thread

Use the following syntax to externally reference the active Thread in a test program:

- ➔ `pattern_sequence_name.Thread.thread_name`
- ➔ `pattern_sequence_name.Thread[#]`

MultiState Waveforms

The user model for the definition of waveforms and timing will be changed slightly to support partial mux mode patterns (only some of the pins in the pattern are muxed). If your program uses partial mux mode you will be required to make a separate WaveformTable for the mux pins. This done so you can define the period for these pins independent of the non mux pins. The zipper section of the PatternSequenceTool will then have at least two waveform tables specified for each waveform reference (one waveform table to resolve the mux pins, and one to resolve the non mux pins).

Note: These changes only apply to programs that have partial mux mode patterns. If all the pins are in mux mode, the user simply programs the period of the WaveformTable to the desired value.

Using the Mouse for PatternTool Operations

You can use the mouse buttons combined with other keys to perform certain functions; refer to [Table 29.18](#).

- Left mouse button (M1)—positions cursor and selects.
- Middle mouse button (M2)—moves and copies the selections and drags the field boundaries.
- Right mouse button (M3)—displays context-sensitive menus in most panes; refer to [Table 29.10](#), .

Mouse actions can be modified by modifier keys, which by themselves, do nothing. However, by pressing and holding them in combination with other keys or mouse buttons, the actions of the keys are modified; refer to [Modifier Keys](#).

Table 29.18: Mouse Functions in PatternTool

Button and Context	Function
M1 in Vectors and Comments	Click—moves cursor. Double-click—selects whole word. Triple-click—selects whole line or vector. Shift and click—extends or shrinks selection, or if no existing selection, begins new selection between cursor and mouse. Control and Shift and click—extends or shrinks selection rectangularly. Drag—selects text between where mouse was pressed and where it was released. Control and drag—selects rectangle between where mouse was pressed and where it was released.
M1 in Headers	Click—selects a signal. Double click—selects separated set of signals. Triple click—selects whole header. Shift and click—extends or shrinks selection. Drag—selects text between where mouse was pressed and where it was released.

Table 29.18: Mouse Functions in PatternTool (Continued)

Button and Context	Function
M2	Click—copies primary selection to clicked position. Shift and click—moves selection to clicked position, deleting it from its original position. Drag on field boundary or top-header boundary—extends or shrinks position of boundary.
M3	Displays context-sensitive menu of actions appropriate for vector or header. If multiple vectors are selected, all selected vectors are effected.

Table 29.19: Modifier Keys for Mouse Functions

Key	Modification
Shift key	Extends selection to mouse pointer.
Control key	Makes a selection rectangular.

Using Keyboard Shortcuts for PatternSequenceTool Operations

Keyboard shortcuts are listed on the right hand sides of the pull-down menus. Besides these keyboard shortcuts, you can use modifier keys to enable more shortcuts; refer to [Modifier Keys](#) and [Action Keys](#).

Modifier Keys

Pressing modifier keys by themselves has no effect, but pressing and holding them in combination with other keys, modifies the actions of those keys; refer to [Table 29.20](#).

Table 29.20: Modifier Keys

Keys	Modification
Control	Extends scope of action key. For example, Home normally moves cursor to start of a line, while pressing Control and Home moves the cursor to beginning of pattern. Back Space deletes one character, while Control and Back Space deletes one word.
Shift	Extends selection to cursor position. If nothing is already selected, begins selection point between the old and new cursor positions.
Alt	When modifying a selection, makes the selection rectangular.

Action Keys

Action keys and their functions are listed in [Table 29.21](#).

Table 29.21: Action Keys

Keys	Function
Escape	Cancels current operation: menu selection, drag, or selection. Equivalent to Cancel button in dialogs.
Back Space	Deletes character before cursor.
Control and Back Space	Deletes word before cursor.
Left Arrow	Moves cursor to left one character.
Control and Left Arrow	Moves cursor backward one word.
Right Arrow	Moves cursor to the right one character.
Control and Right Arrow	Moves cursor forward one word.
Up Arrow	Moves cursor up one line.
Down Arrow	Moves cursor down one line.
Return	Inserts a vector.
Shift and Return	Inserts a vector above vector containing cursor.
Control and Return	Inserts a comment after cursor vector.
Control and Shift and Return	Inserts a comment before cursor vector.
Tab	Moves to next field.
Shift and Tab	Moves to previous field.
Control and /	Selects everything. Same as Select All menu item or ^A.
Control and \	Unselects.
Control and U	Deletes to start of line.
Control and Insert	Copies primary selection to clipboard., Same as Copy menu item or ^C.
Shift and Control and Insert	Copies primary selection to cursor location.
Del	Deletes character after cursor (or under cursor in overstrike mode).
Control and Del	Deletes to end of line.

Table 29.21: Action Keys (Continued)

Keys	Function
Shift and Del	Removes currently selected text and places it in the clipboard. Same as Cut menu item or ^X.
Home	Moves cursor to beginning of line.
Control and Home	Moves cursor to the first vector of pattern.
End	Moves cursor to the end of line.
Control and End	Moves cursor to last vector of pattern.
Page Up	Scrolls up one page.
Page Down	Scrolls down one page
F10	Activates menu bar for keyboard input (Arrow Keys, Return, Escape, and Space Bar).

