This document is for information and instruction purposes. Mentor Graphics reserves the right to make changes in specifications and other information contained in this publication without prior notice, and the reader should, in all cases, consult Mentor Graphics to determine whether any changes have been made.

The terms and conditions governing the sale and licensing of Mentor Graphics products are set forth in written agreements between Mentor Graphics and its customers. No representation or other affirmation of fact contained in this publication shall be deemed to be a warranty or give rise to any liability of Mentor Graphics whatsoever.

MENTOR GRAPHICS MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

MENTOR GRAPHICS SHALL NOT BE LIABLE FOR ANY INCIDENTAL, INDIRECT, SPECIAL, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING BUT NOT LIMITED TO LOST PROFITS) ARISING OUT OF OR RELATED TO THIS PUBLICATION OR THE INFORMATION CONTAINED IN IT, EVEN IF MENTOR GRAPHICS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

U.S. GOVERNMENT LICENSE RIGHTS: The software and documentation were developed entirely at private expense and are commercial computer software and commercial computer software documentation within the meaning of the applicable acquisition regulations. Accordingly, pursuant to FAR 48 CFR 12.212 and DFARS 48 CFR 227.7202, use, duplication and disclosure by or for the U.S. Government or a U.S. Government subcontractor is subject solely to the terms and conditions set forth in the license agreement provided with the software, except for provisions which are contrary to applicable mandatory federal laws.

TRADEMARKS: The trademarks, logos and service marks ("Marks") used herein are the property of Mentor Graphics Corporation or other parties. No one is permitted to use these Marks without the prior written consent of Mentor Graphics or the owner of the Mark, as applicable. The use herein of a third-party Mark is not an attempt to indicate Mentor Graphics as a source of a product, but is intended to indicate a product from, or associated with, a particular third party. A current list of Mentor Graphics’ trademarks may be viewed at: www.mentor.com/trademarks.

The registered trademark Linux® is used pursuant to a sublicense from LMI, the exclusive licensee of Linus Torvalds, owner of the mark on a world-wide basis.

Mentor Graphics Corporation
8005 S.W. Boeckman Road, Wilsonville, Oregon 97070-7777
Telephone: 503.685.7000
Toll-Free Telephone: 800.592.2210
Website: www.mentor.com
SupportNet: supportnet.mentor.com/

Send Feedback on Documentation: supportnet.mentor.com/doc_feedback_form
# W-Edit 16 User Guide—Contents

## 1 Overview

**Product Capabilities** ................................................................. 9
Sweep Types ................................................................................. 10
Analysis Types .............................................................................. 10
Operating Point Analysis ............................................................ 10
Non-Operating Point Analysis ...................................................... 11
Parametric Variations .................................................................. 11
Parametric Analysis ..................................................................... 11

**Documentation Conventions** .................................................. 12
Keyboard and Mouse Usage .......................................................... 12
Menu Commands and Dialog Titles ............................................. 12
Special Fonts .............................................................................. 12

## 2 Interface and File Locations

**Launching W-Edit** ................................................................. 14
**Parts of the W-Edit Interface** ................................................ 15
Title Bar ...................................................................................... 16
Menu Bar ..................................................................................... 16
Toolbars ...................................................................................... 16
  Standard Toolbar ....................................................................... 16
  Draw Toolbar ............................................................................ 17
  Zoom and Pan Toolbar ............................................................ 17
  Locator Toolbar ........................................................................ 17
  Mouse Buttons Toolbar ........................................................... 17
  Customizing Toolbars .............................................................. 18
    Adding Toolbars ..................................................................... 18
    Adding Commands .................................................................. 19
    Adding and Displaying Keyboard Shortcuts ........................... 20
    Menu and Toolbar Display Options ..................................... 21
    Customizing Toolbar Buttons ............................................. 22
Status Bar ................................................................................... 23
Chart Window ............................................................................. 24
Command Window ..................................................................... 24
  Copying and Pasting in the Command Window ...................... 25
  Command Window Log Files .................................................. 25
    How to Locate Log Files .................................................... 26
  Setting Command Window Text Styles ................................ 26
  Filtering Information in the Log File ..................................... 28
Simulations Navigator ............................................................... 29
Variation Navigator .................................................................... 29
Chart Parameters ...................................................................... 30
Trace Navigator ......................................................................... 30
Properties Navigator .................................................................. 30
Waveform Calculator .................................................................. 31

**Arranging Interface Elements** ............................................... 31
Docking and Floating Windows in W-Edit ................................. 32
  Instantly Hiding Everything But the Chart Window .............. 32
Language Selection and Simulation Retention .......................... 33
Mouse Wheel Options ............................................................... 34
Text File Update Options .......................................................... 34
Text File Display Options .......................................................... 35
Saving and Reloading Setup Options ....................................... 38
Order of Preference for Setup File Locations ........................... 39
How to Use the Trace Styles Setup Table ................................................. 74
Setting Display Style for an Individual Trace ........................................... 74
Example: Varying Curve Display for Multiple Variables ................................ 74
Setting the Appearance of Selected and Unselected Objects .......................... 75

Using the Trace Navigator ................................................................. 76
Right-Click Menu in the Trace Navigator .................................................. 78
Adding and Removing Traces from a Chart .............................................. 79
Moving a Trace from One Plot to Another ................................................. 80
Expanding and Collapsing Traces ........................................................... 80
Putting Compatible Traces in Separate Plots—Expand .................................. 80
Putting Compatible Traces in the Same Plot—Collapse ................................. 81
Showing and Hiding Traces .................................................................. 82
Selecting Traces and Curves From the Chart Legend .................................. 82

Panning and Zooming in a Chart ............................................................ 82
Measuring Trace Data ......................................................................... 83
Missing Data Points ............................................................................. 83
The Draw Toolbar ................................................................................. 84
Data Point Tool Tips ............................................................................ 85

Cursor Bars and the Cursor Table .......................................................... 85
The Cursor Table ................................................................................. 86
Cursor Table Functions ....................................................................... 86
Vertical Cursor Bars ............................................................................. 87
Horizontal Cursor Bars ....................................................................... 88

Markers ............................................................................................... 88
Plain Markers ....................................................................................... 89
Labeled Markers .................................................................................. 89
Editing a Labeled Marker ...................................................................... 91
Text Labels .......................................................................................... 92
Editing Text Labels .............................................................................. 93

Exporting Trace Data .......................................................................... 94

4 Scripting with TCL ............................................................................. 95
TCL Commands Available in W-Edit ....................................................... 95
Running TCL Scripts ........................................................................... 96
Source Scripts ...................................................................................... 97
Executing Scripts Automatically ........................................................... 98
Running a Script when W-Edit Launches ............................................... 98
Running a Script when W-Edit Closes ................................................... 98
Running a Script when a Chartbook Opens .......................................... 98
Load Order for TCL Files ..................................................................... 98
Load Order for Setup Scripts ............................................................... 99

5 Waveform Calculator ....................................................................... 100
Using TCalc in W-Edit ....................................................................... 100
Basic Data Types ................................................................................ 100
Real Numbers ...................................................................................... 100
Scale Factors ...................................................................................... 101
Units .................................................................................................. 101
Complex Numbers .............................................................................. 102
Vectors ............................................................................................... 102
Traces ................................................................................................. 102

Constants .............................................................................................. 102
Mathematical Constants ..................................................................... 103
Physical Constants ............................................................................. 103
Logic Constants .................................................................................. 103

Operators ............................................................................................. 104
Mathematical Operators ..................................................................... 104
W-Edit 16 User Guide—Contents

(Continued)

Relational and Logical Operators ........................................ 104
Other Operators ................................................................ 104
Operator Precedence .......................................................... 104

Mathematical Functions ..................................................... 105
Exponential Functions ....................................................... 105
  HSPICE Compatibility Mode ............................................. 105
  Exponential Functions that Support Complex Arguments .... 106
Trigonometric Functions .................................................... 106
  Trigonometric Functions that Support Complex Arguments 106
Miscellaneous Functions ................................................... 106
Special Purpose Functions .................................................. 107
Differentiation and Integration Functions ............................... 107
User Defined Functions ..................................................... 107
Complex Functions ........................................................... 107
Vector Functions ............................................................... 108

Trace Functions .............................................................. 108
Trace Intervals ................................................................. 108
Waveforms and Edges ....................................................... 109
  absolutejitter ............................................................... 110
  amin ................................................................. 112
  amplitude ............................................................... 113
  average ................................................................. 114
  bandwidth .............................................................. 115
  clip ................................................................. 117
  compare ............................................................... 118
  cpk ................................................................. 119
  cross ................................................................. 120
  cursor ............................................................... 121
  delay ................................................................. 122
  derivative ............................................................. 123
  derivative2 ............................................................ 124
  derivative2at .......................................................... 125
  diff ................................................................. 126
  dpu ................................................................. 127
  dutycycle ............................................................. 130
  error ................................................................. 131
  falltime .............................................................. 132
  fft ................................................................. 133
  fit ................................................................. 134
  frequency ............................................................ 135
  gainmargin ............................................................ 136
  groupdelay ........................................................... 137
  histogram ............................................................. 138
  integral .............................................................. 139
  intersect .............................................................. 139
  maximum ............................................................. 140
  minimum ............................................................. 142
  nextedge .............................................................. 143
  nextextreme ........................................................... 144
  nextpoint ........................................................….. 144
  overshoot .............................................................. 145
  Parametrictrace define ................................................ 147
  period .............................................................. 148
1 Overview

W-Edit is a waveform viewer that provides ease of use, power and speed with a flexible user interface so you can visualize the complex numerical data resulting from circuit simulation. In addition, W-Edit provides saved work spaces, scriptable graph construction and trace calculation, and easy-to-export presentation-quality graphics.

As part of the Tanner EDA tool flow, W-Edit is optimized to display the full performance and capacity of the T-Spice circuit simulator and all advanced functionality in the S-Edit schematic editor. You can chart data generated by T-Spice and S-Edit directly without modification of the output data files, with dynamic display updates produced during simulation.

Product Capabilities

- **Analysis Platform** W-Edit is not only a waveform viewer, but is an analysis platform featuring built-in measurements that can be applied to selected traces. These include amax, amin, amplitude, average, baseline, compare, cross, delay, derivative, edgethreshold, error, falltime, frequency, integral, intersect, maximum, minimum, nextedge, nextextreme, nextpoint, overshoot, period, previousextreme, previouspoint, pulsedwidth, risetime, rms, slewrate, smooth, topline, undershoot, window, xval, ymax, ymin, and yval.

- **Arithmetic Traces** W-Edit supports the ability to create new traces from arithmetic expressions of other traces, using a calculator-style wizard for arguments and syntax. Newly created arithmetic traces can be saved in the W-Edit chartbook.

- **Multiple Simulations** W-Edit supports concurrent simulation and curve viewing so the results of multiple simulations can be plotted together for comparison and measurements. You can load and view multiple simulations from different output files, in one or more chart windows, and traces can be copied between charts.

- **Programmable** W-Edit is fully programmable with TCL so you can write scripts to automate, for example, results analysis, trace depiction or complex number arithmetic.

- **Performance** W-Edit performance capability is increased to handle the largest data files quickly and efficiently. The Trace navigator can display traces in a flat or hierarchical view, with filters that include wildcards and regular expressions.

- **Chartbooks** All configuration aspects of a chart, as well arithmetic traces, can be saved to a chartbook for later display and analysis.

- **Intuitive User Interface** All Tanner applications present a clear and intuitive Windows®1 interface that is easy to learn and use. In W-Edit this includes the a drag-and-drop feature for placing traces in a chart, full control over curve display aspect such as color, line weight, visibility or scaling, and multiple cursor types with built-in measurements.

---

1 Windows is a registered trademark of Microsoft Corporation in the United States and other countries.
Sweep Types

Almost any of the circuit's physical, electrical, or parametric values may be swept. Typically, the sweeps are for transient time, voltage and current source values, temperature, and parameter values.

- **lin**: A linear sweep from start value to stop value by increment.
- **oct**: A logarithmic sweep by octaves.
- **dec**: A logarithmic sweep by decades.
- **list or poi**: A simple list of values.
- **Transient**: Transient analysis generates time-dependent data, with dynamic sized time steps.
- **data**: A collection of multiple data values to be swept.
- **monte**: Values are generated from a random distribution function for Monte Carlo analysis.
- **step**: A special type of sweep, from the `.step` command. Step sweeps are processed as the outer loop of all other sweeps and simulation commands.

Analysis Types

A T-Spice or S-Edit simulation consists of one or more analysis commands written to determine a circuit’s behavior under a given set of operating conditions.

- **.OP**: Operating point analysis
- **.TRAN**: Transient analysis
- **.DC**: DC transfer curve analysis
- **.AC**: Frequency response analysis
- **.TF**: Small-signal transfer function

Operating Point Analysis

Operating point analysis computes the steady-state collection of node voltage and branch currents, together with each device’s internal small-signal values. Operating point analysis is always performed prior to the other types of analysis, but values are only printed out when the `.op` command is present, or if it is AC analysis.

Operating point solutions consist of a collection of scalar values:

- Node voltage values for all nodes in the circuit (sometimes including internal, state variable, nodes)
- Voltages source terminal currents
- Possibly the terminal current and charge values for all devices
- Possibly a collection of AC small-signal parameters for most devices (name-value pairs)
Normally, there is one OP data set per circuit configuration. T-Spice supports OP data sets at user-requested times of a transient analysis.

**Non-Operating Point Analysis**

Each of the other (non-OP) simulation analyses results in multiple traces of output data, according to the use of the `.print` (text printout) and `.probe` (binary printout) commands. Also, each analysis may contain multiple levels of data sweeps, resulting in an N-1 dimensional table of traces for N sweeps.

For instance, if we have the following two commands in our input netlist:

```
.print dc v(*)
.dc Vin 0 5 0.1 sweep Vref 1 2 0.1
```

We will obtain eleven traces of data per node, since the `Vref` voltage is swept from 1 to 2 volts in increments of 0.1. Each trace will consist of 51 XY data pairs, corresponding to node voltage versus `Vin` voltage, with `Vin` swept from 0 to 5 volts in increments of 0.1.

Note that `.print` results from T-Spice are capped at 50 traces and will be logged instead after that point.

**Parametric Variations**

The topology and physical or electrical characteristics of the circuit may be varied in a controlled manner, and each of the analysis commands will be executed for the new configuration.

The so-called parametric variation commands are:

- `.Temp` – temperature variations
- `.Step` – performs a multi-dimensional sweep outside of the analysis command sweep, i.e. as an outer loop for all analyses
- `.Alter` – modifies the circuit using all of the succeeding commands and options. This is functionally equivalent to re-running the entire input file up to the `.alter` statement, with all changes which follow the `.alter`

For each parametric variation, all of the analysis commands are performed, resulting in another block of datasets.

**Parametric Analysis**

T-Spice supports a number of parametric analysis and post-processing commands. Each of the parametric analysis commands results in a block of output data that summarizes the operation or measurement. The parametric analysis types are:

- `.four` Performs Fourier analysis to compute the harmonic components of transient data.
- `.measure` Computes measurements of output data such as min, max, slew, etc.
- `.power` Computes the dissipated or stored power of a source or a device for the transient analysis.
- `.noise` Computes circuit noise contributions during an AC analysis.
Monte Carlo analysis performs random variations of parameters and summarizes the results. You can plot the “.measure” parameter to a specific Monte Carlo swept parameter using a parametric plot.

Optimize performs systematic variations of parameter values to minimize an objective function and summarize results.

Documentation Conventions

This section contains information about the typographical and stylistic conventions used in this user guide.

Keyboard and Mouse Usage

References to mouse buttons are given in all capitals—for example, MOVE/EDIT.

The terms “left-click,” “right-click,” and “middle-click” all assume default mappings for mouse buttons.

When a key is to be pressed and held while a mouse button is used, the key and button are adjoined by a plus sign (+). For example, Shift+SELECT.

Similarly, if certain keys are to be pressed simultaneously, their abbreviations are adjoined by a plus sign (+). For example, Ctrl+R means that the Ctrl and R keys are pressed at the same time.

When certain keys are to be pressed in sequence, their abbreviations are separated by a space. For example, Alt+E R means that the Alt and E keys are pressed at the same time and then released, immediately after which the R key is pressed.

Abbreviations for alternative key-presses are separated by a slash (/). For example, Shift+↑ / ↓ means that the Shift key can be pressed together with either the up (↑) arrow or the down (↓) arrow.

Menu Commands and Dialog Titles

Elements in hierarchical menu paths are separated by a > sign. For example, File > Open means the Open command in the File menu.

Tabs in dialog boxes are set off from the command name or dialog box title by a dash. For example, Setup > Layers > General refers to the General tab of the Setup Layers dialog.

Special Fonts

The following references are represented by a bold font:

- Menu and simulation commands (For example: .print tran v(out).)
- Literal user input (For example: Enter 14.5.)
- Program output (For example: S-Edit generates names for the ports on the symbol based on the PAD string.)
- All dialog elements—fields, checkboxes, drop-down menus, titles, etc. (For example: Click Add.)
Freestanding quotations of input examples, file listings, and output messages are represented by a constant-width font—for example:

.ac DEC 5 1MEG 100MEG

Variables for which context-specific substitutions should be made are represented by bold italics—for example, `myfile.tdb`.

Sequential steps in a tutorial are set off with a checkbox (☑) in the margin.
Launching W-Edit

To launch W-Edit, click Start on the Windows taskbar, then All Programs, and navigate to Tanner EDA > Tanner Tools \texttt{vxx.y > W-Edit}, where \texttt{xx.y} is the version number of Tanner Tools.

You can also launch W-Edit from the T-Spice toolbar by pressing the W-Edit program icon.

There are also several other ways you can set the Tanner applications T-Spice or S-Edit to run W-Edit, as well.

From T-Spice, you can use Simulation > Simulation Settings—W-Edit to set W-Edit to launch and display waveforms during a simulation, after a simulation is complete, or not at all.

Similarly, in the S-Edit Setup Simulation dialog there is a Show Waveforms field where you can select During, After or Don’t Show from a drop-down list.

S-Edit will also launch W-Edit as the result of a probe command.
Parts of the W-Edit Interface

The W-Edit user interface consists of the components shown below.
Chapter 2: Interface and File Locations

Title Bar

The title bar shows the name of the current chart and simulation type, and lets you resize, move, or close the application window.

Menu Bar

The menu bar contains the W-Edit command menu titles.

Toolbars

You can display or hide individual toolbars using the View > Toolbars command, or by right-clicking in the toolbar region. Toolbars can be relocated and docked as you like.

You can also customize your toolbars; see “Customizing Toolbars” on page 18.

Standard Toolbar

The Standard toolbar provides buttons for commonly used file and editing commands, as well as operations specific to W-Edit such as “New chartbook.”

New File
Opens a new, blank text file in the active window.

New Chartbook
Opens a new chartbook. See “Creating a Chartbook” on page 49.

Open Chartbook
Opens an existing chartbook. See “Opening a Chartbook” on page 49.

Save Chartbook
Saves the active chartbook. See “Saving a Chartbook” on page 50.
Save all chartbooks

Saves the active chartbook plus any library files that have been modified and that you have exclusive access to.

Erase

Removes the highlighted trace from the active plot.

Cut

Deletes highlighted contents and writes it to the clipboard.

Copy

Copies highlighted content to the clipboard.

Paste

Copies the contents of the clipboard to the active window. When a trace is selected this will

Undo

Retracts the most recent edit operation and restores the file to the state it was in prior to that operation.

Redo

Restores the change previously reversed with the last undo command.

Print

Opens the Print dialog. See “Copying or Printing a Chart Image” on page 59.

About

Opens the About W-Edit dialog. See “W-Edit Product Support and Diagnostics” on page 44.

Waveform Viewer Manual

Launches Adobe Reader if it not already running, and opens the W-Edit User Guide.

Cancel script execution

Cancels execution of the active script, if one is running.

Draw Toolbar

The Draw toolbar provides tools to locate specific data points. See “The Draw Toolbar” on page 84.

Zoom and Pan Toolbar

The Zoom and Pan toolbar provides tools with which to quickly navigate within a chart. See “Panning and Zooming in a Chart” on page 82.

Locator Toolbar

The Locator toolbar displays the coordinates of the mouse cursor when it is on a plot.

Mouse Buttons Toolbar

The Mouse Buttons toolbar shows the current functions of the mouse buttons. Mouse buttons vary in function according to the location of the mouse and the state of the Shift, Ctrl and Alt keys.
All Tanner tools use **Ctrl+click** to modify selection behavior. For example, in W-Edit, **Ctrl+click** on a curve selects the trace rather than the specific curve. **Shift** causes the selection to be extended; that is, previous selections are maintained while the element clicked is also selected.

For a two-button mouse, the middle-button function is accessed by clicking the left and right buttons at the same time, or by pressing **Alt** while clicking the left mouse button.

**Customizing Toolbars**

This context-sensitive menu lets you completely customize the W-Edit toolbars. Right-click anywhere in the toolbar area and click on **Customize** to open the customization dialogs.

**Adding Toolbars**

Use **Customize > Toolbars** to add, remove and rename your own custom toolbars. The checkmarks control only whether or not a toolbar is displayed. **Reset** returns a toolbar to the default Tanner settings.

The **Reset** function applies only to the toolbar that is highlighted, and it will be applied even when that toolbar is not currently displayed.
Adding Commands

Use **Customize > Commands** to add an existing command or menu to any toolbar, or to create new commands and toolbars. To add an existing command, highlight a menu in the **Categories** pane then click and drag the desired command from the **Commands** pane to the toolbar of your choice. W-Edit will insert a button with the command text, or the icon if one is already defined.

To add a new command, choose **Custom** from the **Categories** pane, then drag “**Execute button text as Tcl**” to the toolbar of your choice.

When you replace the text “Execute button text as Tcl” with the name of a TCL function that is loaded in the current session of W-Edit, the button to execute that TCL operation.

If you save the TCL files for button functions to the **C:\Documents and Settings\<username>\Application Data\Tanner EDA\scripts\startup** directory, they will be available every
time W-Edit is launched. Buttons can be further customized as described in “Customizing Toolbar Buttons” on page 22.

Adding and Displaying Keyboard Shortcuts

Customize > Keyboard lets you add or change shortcut key assignments for menu commands.

![Customize Keyboard Window]

**Category**
Select the menu to which the command belongs.

**Commands**
Select the command for which you want to add or change a keyboard shortcut.

**Key assignments**
Displays existing key assignments. If blank, no shortcut is assigned. It is possible to have more than one shortcut for a command.

Highlight a command and click on **Remove** to remove a shortcut.

To restore the default settings click on **Reset All**.

**Press new shortcut key**
Highlight a command in the **Commands** pane, then use this field to enter the key(s) that will be the shortcut.

You can use any combination of the **Alt**, **Shift** and **Ctrl** keys with any of the character keys. W-Edit will warn you if your entry is already in use. (Since this field interprets any key you press literally, you cannot delete a value in this field—simply enter a different value.)

Click on **Assign** to save your shortcut.

**Description**
This display-only field shows each command’s tool tip.
### Menu and Toolbar Display Options

**Customize > Options** is an assortment of display controls for menus and toolbars.

![Customize Options](image)

<table>
<thead>
<tr>
<th><strong>Always show full menus</strong></th>
<th>Menus and toolbars will automatically adjust based on how often you use commands so that only the commands you use most often are displayed. If you prefer, you can choose the <strong>Always show full menus</strong> option so that all commands are displayed on the menu.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Show full menus after a short delay</strong></td>
<td>(Not operational.)</td>
</tr>
<tr>
<td><strong>Reset menu and toolbar usage data</strong></td>
<td>Deletes the record of all the commands used in W-Edit (for short menu display) and restores the default set of visible commands to the menus and toolbars. However, explicit changes you have made in the current or earlier sessions will remain.</td>
</tr>
<tr>
<td><strong>Large icons</strong></td>
<td>Check this box to display large toolbar buttons.</td>
</tr>
<tr>
<td><strong>Show ScreenTips on toolbars</strong></td>
<td>Check this box to display tool tips for the toolbar buttons.</td>
</tr>
<tr>
<td><strong>Show shortcut keys in ScreenTips</strong></td>
<td>(A tool tip is a brief reminder of the related command that is displayed when your mouse hovers over a toolbar button.)</td>
</tr>
<tr>
<td><strong>Menu animations</strong></td>
<td>You can choose from the list to add animation to the open menu operation.</td>
</tr>
</tbody>
</table>
Customizing Toolbar Buttons

If you right-click on a toolbar button while the toolbar Customize dialog is open, you will access the button controls shown below.

- **Reset** (Not operational.)
- **Delete** Deletes the selected button.
- **Name** Use this field to edit the button name when it is displayed as text. The tool tip will not be affected.
- **Copy Button Image** Copies the selected button image.
- **Paste Button Image** Pastes the selected button image.
- **Reset Button Image** Resets all changes to the button image and text.
- **Edit button Image** Opens the Button Editor where you can perform basic graphic functions.
## Change Button Image
Opens a small palette of clip art from which you can choose an icon.

<table>
<thead>
<tr>
<th>Image 1</th>
<th>Image 2</th>
<th>Image 3</th>
<th>Image 4</th>
<th>Image 5</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image 1" /></td>
<td><img src="image2.png" alt="Image 2" /></td>
<td><img src="image3.png" alt="Image 3" /></td>
<td><img src="image4.png" alt="Image 4" /></td>
<td><img src="image5.png" alt="Image 5" /></td>
</tr>
</tbody>
</table>

## Default Style
Displays the default (image only) for the selected toolbar button.

## Text Only
Displays just the contents of the Name field for the selected toolbar button.

## Image and Text
Displays both the icon and the text from the Name field for the selected toolbar button.

## Begin a Group
Inserts a fine line denoting a toolbar group to the left of the selected button.

## Status Bar
The Status Bar has several panes. Display varies with the type and number of objects selected and the active mode. You can use View > Status Bar to toggle display on and off.

- **Selection**: When the cursor is positioned over a toolbar button or highlighted menu command, the first pane displays a short description of the function of the button or command. When a legend item is selected in a chart, the first pane displays the trace name.
- **Select**: The second pane displays the mode that is currently active.
- **CAP NUM OVR**: The third pane displays the status of three special function buttons, where a visible abbreviation means the function is enabled. CAP indicates that Caps Lock is on, enabling all letters to be typed in capitol. NUM indicates that Num Lock is on, enabling the numeric keypad. OVR indicates that Scroll Lock is on, modifying the arrow keys so they scroll the contents of a text window instead of moving the cursor.
Chart Window

The *chart window* consists of the display area in W-Edit interface not occupied by the toolbars, navigators, command window or calculator. You can open, view, edit, resize, or rearrange multiple charts in the chart window. See “Working with Charts” on page 54.

You can redraw the chart window display at anytime by pressing the SPACE bar.

Command Window

All events that occur in the W-Edit chart area are recorded in tool command language (TCL) format to a log displayed in the *Command window*. TCL files are macro-like scripts that allow you to perform or repeat operations.

The command window serves as both a recording and a playback device for TCL files. As such, any action or operation performed by W-Edit can be copied or replayed. Text can be typed in, copied from executed operations and then pasted back into the command window, or written in from a saved TCL file to instantly perform the desired operations. This is especially useful for automating and simplifying difficult or repetitive tasks.

You can use **View > Activate Command Window** (shortcut backquote ‘) to open the command window if it is not open. If it is open, backquote will shift focus to it.
Copying and Pasting in the Command Window

A right-click in the command window opens the menu below.

<table>
<thead>
<tr>
<th>Command</th>
<th>Shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search</td>
<td>Ctrl+V</td>
</tr>
<tr>
<td>Paste</td>
<td>Ctrl+C</td>
</tr>
<tr>
<td>Copy (Visible Text Only)</td>
<td>Ctrl+C</td>
</tr>
<tr>
<td>Copy Commands Only</td>
<td>Ctrl+C</td>
</tr>
<tr>
<td>Copy (Include Hidden Text)</td>
<td>Ctrl+C</td>
</tr>
<tr>
<td>Copy to File</td>
<td>Ctrl+V</td>
</tr>
<tr>
<td>Delete Last Line</td>
<td>Ctrl+L</td>
</tr>
<tr>
<td>Clear All</td>
<td>F9</td>
</tr>
<tr>
<td>Customize</td>
<td>F8</td>
</tr>
</tbody>
</table>

Search
- Searches the contents of the command window.

Paste
-Copies the contents of the clipboard to the command window.

Copy (Visible Text Only)
-Copies just the text visible in the command window.

Copy Commands Only
-Copies just the TCL commands from the command window.

Copy (Include Hidden Text)
-Copies all formatting tags and hidden text from the command window.

Copy to File
-Opens a Save As window so you can paste the contents copied from the command window into a .tcl file.

Delete Last Line
-Deletes just the last line from the command window.

Clear All
-Clears the entire contents of the command window.

Customize
-Opens the Customize Command Window dialog, which is discussed in the following sections:
  - “Command Window Log Files,” below
  - “Setting Command Window Text Styles” on page 26
  - “Filtering Information in the Log File” on page 28

Command Window Log Files

Each time W-Edit is launched it creates a TCL log file that records each operation performed in the chart windows for the entire session. The last 10,000 lines (a default value which can be changed, see “Filtering Information in the Log File” on page 28) of this log are displayed in the Command window. Logs files are identified by the date and time the session was launched.
How to Locate Log Files

To set the directory where log files are stored, right-click in the Command window to open its context-sensitive menu. Select Customize (shortcut F8) to open the Customize Command Window dialog, then use the Folders tab to select or create a storage directory.

If no path is set in this dialog, then logs are stored in the location set by the TANNERLOGPATH environment variable.

If there is no TANNERLOGPATH environment variable, logs are stored in a folder “Tanner Logs” under the temp folder, which can be %TMP% or %TEMP% or even the WINDOWS folder on your hard drive, depending on the configuration of your Windows environment.

Setting Command Window Text Styles

Right-click in the Command window and select Customize (or use the shortcut F8 while the Command window is active) to open the Customize Command Window dialog, then use the General tab to set
Command window display characteristics for each of the categories of text—error messages, warnings, modules, etc.

Font

Use this field to **pick** a font or set it to the **default**.

Background

Use this field to **pick** a background color or set it to the **default**.

Editing

Use this field to set the behavior of the backspace and tab keys.

- **Delete acts as backspace**. When this checkbox is enabled, the **Delete** key functions like the **Backspace** key by removing text to the left of the cursor. When this checkbox is not checked, the **Delete** key removes text to the right of the cursor as usual.

- **Tab expands keywords**. When this box is checked, you can use the **Tab** key to expand all keywords.

Context highlighting

When you select a text type in the upper pane, you can check the boxes to make it **Bold** and **Italic**, and also **pick** a text color or set it to the **Default**.
Filtering Information in the Log File

Use **Customize Command Window—Filters** to limit the type or amount of information written to the log file. You can also set certain display controls for the Command window which do not effect the log file.

### Events to log

Check the boxes to include events categorized as **Warnings** and **Information** in the log file and Command window display. **(Actions and Errors** are always written to the log file.)

The higher the **Verbosity** value, the higher the detail included in event messages written to the log file.

### Modules

Each program module in W-Edit generates a set of messages, identified by a preceding `# module_name`, where `module_name` is a three character abbreviation such as “LIC” for license related messages. Enter these abbreviations, separated by commas, to exclude or include a type of message from the log file and Command window.

When **Exclude messages from module** is checked, messages from any module abbreviations listed in the corresponding entry field will not be written to the log file.

When **Include messages from module** is checked, messages from any module abbreviations listed in the corresponding entry field will be written to the log file.
Log viewer

Note: These options effect the Command window display only, not what is written to the log file.

Check the Display Timestamps box to display the timestamp when one is included in a message.

Enter an integer between 100 and 100,000 to set the Number of lines at the end of the log file to display.

Simulations Navigator

See “Using the Simulations Navigator” on page 50.

Variation Navigator

See “Using the Variation Navigator and Chart Parameters” on page 52.
Chapter 2: Interface and File Locations

Parts of the W-Edit Interface

**Chart Parameters**

See “Using the Variation Navigator and Chart Parameters” on page 52.

![Eye Diagram Parameters](image)

**Trace Navigator**

See “Adding and Removing Traces from a Chart” on page 79.

![W-Edit](image)

**Properties Navigator**

Each chart and trace has a set of properties associated with it, see “Properties Navigator for Charts and Traces” on page 57.
Waveform Calculator

See the chapter “Waveform Calculator” on page 100.

Arranging Interface Elements

The W-Edit interface is highly flexible, allowing you to display and arrange elements as you choose.

You can dock individual W-Edit toolbars, navigators and the chart window, or they can float anywhere in the application window. With the exception of the locator toolbar, toolbar buttons will be arranged vertically when docked to the sides. Chart windows can be shown, hidden or made dockable directly from the interface.

Display colors can be customized for simulation runs, charts, traces, etc., and font colors can be set differently in the text editor for each of the file types it reads. Settings for these and other configuration parameters are available in the Setup menu. The section “Unselected horizontal cursors and unselected labels remain on their original plot and remain unselected.” on page 82 describes how to change these settings.
Docking and Floating Windows in W-Edit

W-Edit uses standard Windows docking behavior. When you drag any of the interface windows and release the cursor over a blue docking arrow, it will dock as shown below.

### Edge Arrows
When you release a window over a docking arrow at an edge of the screen, it will dock at the corresponding edge, filling the entire application area.

### Center Arrows
When you release a window over a docking arrow at the center of the screen, it will dock on the corresponding side, between the other docked windows.

Instantly Hiding Everything But the Chart Window

The command **View > Hide Docked Views** acts as a toggle to hide or show all docked views so that the Chart Window zooms to fill the application screen and no other W-Edit panes are visible.
Language Selection and Simulation Retention

Use Setup > General to set save options and the user interface language. There are also three ways to control for the circumstances under which charts and simulations are saved.

### Language
Select a language for menu and dialog text from the drop-down list. Options are English (American), Japanese, simplified Chinese and traditional Chinese.

### Warn when chart closes that chart configuration is deleted
A check in this box sets W-Edit to provide a prompt when you close an unsaved simulation chart.

### Keep the newest _xx_ simulations
Use the box control to set the number of most recent simulations W-Edit lists under Files > Recent Files.

### Create a new chartbook for an incoming simulation with a different base name
When checked, automatically creates a new chartbook when W-Edit runs a simulation having a different name than the active simulation.

See “Preserving or Overwriting Simulation Files” on page 51 for how this setting interacts with related settings in S-Edit and T-Spice.

### Add .print traces for all incoming simulations
When checked, traces added to a tsim file by .print and .probe commands in the netlist are automatically loaded to existing or new charts.

This option is useful when you want to suppress new chart creation when adding a simulation to a chartbook, to decrease file load times. Unchecking this option suppresses any addition of printed traces to new or existing charts during a dynamic update.
Save chartbook using relative paths to simulation data

When checked, the paths to simulation data that are saved in the .tsim chartbook file will be relative to the location of the chartbook file. As this is a global option, all chartbooks will contain either relative or absolute paths to the tsim simulation databases.

If a chartbook is opened and the simulation data cannot be found — whether because the chartbook has been moved thus breaking the relative path, or because the target has been deleted thus breaking the absolute path — W-Edit will prompt you to browse for the data. Once the proper directory is located, W-Edit will use that path to resolve any subsequent broken until links until the directory does not resolve the problem, at which point it will prompt you again for another path location.

**Mouse Wheel Options**

Use **Setup > Mouse** to set how the chart window view changes when you spin the mouse wheel away from your hand.

| Scroll wheel |
|------------------|------------------|
| **Zoom in**      | zooms in to the active view to magnify the view by a factor of three. |
| **Zoom out**     | zooms out of the active view by a factor of three. |
| **Pan up (shift for right)** | pans the chart view up with an upward spin of the mouse wheel or down with a downward spin. Hold the Shift key to pan right with an upwards spin or left with a downwards spin. |

**Text File Update Options**

The **Setup > Text Editor and Styles > Text Editor** dialog governs if and how open text files are updated when they are saved outside the application.

The W-Edit text editor checks the stored version of a file for modifications whenever files are saved, first changed, and when the text window or application becomes active or is closed. If a file has not
been modified outside W-Edit, nothing will happen. If a file has been modified outside W-Edit, the selected action is triggered.

**Auto-Load**

- **Load all**—W-Edit automatically updates text files that have been modified outside of the text editor.
- **Prompt to Load**—(default) If a text file is modified outside W-Edit, you will be prompted to accept or ignore those changes.
- **Ignore all**—externally modified files are not updated to W-Edit.

**Text File Display Options**

Each of the predefined file formats in W-Edit has two setup pages for its display characteristics, **Setup > Text Editor and Styles > {filetype} > Style** and **Setup > Text Editor and Styles > {filetype} > Keywords**.

There are two setup pages, **Style** and **Keywords**, for each of the predefined file types.
For each of the given file types, **Setup > Text Editor and Styles > Styles > {filetype} > Style** sets default font attributes.

### Font
Select or enter a font from the **Face Name** drop-down menu and a point size in the **Size** field.

### Tabulation
Enter a positive integer value to set the increment, in spaces, of the tab spacing the text editor uses.

### File extensions
Enter the extensions of the files that W-Edit should include in the active category, separated by commas with no spaces.
For each of the given file types, Setup > Text Editor and Styles > Styles > {filetype} > Keywords sets how different categories of information, called keyword groups, are displayed.

**Keyword groups** are categories of text for which you can set display characteristics in the W-Edit text editor. In the example shown above, W-Edit is set to display the keyword group “SPICE Options Keywords” in teal text against a white background.

**Groups**

Displays the keyword groups defined for a given file type.

**Keyword group**

Use Add to enter the name of a new keyword group. Use Edit to enter the terms belonging to a keyword group. Use Remove to delete a keyword group.

**Note:** Each file type has a set of predefined keyword groups that cannot be edited or deleted.

**Colors**

Use Foreground and Background to set the respective colors for a keyword group.

In the log file shown here, warnings are formatted to display in blue text and errors are formatted to display in red text.
Saving and Reloading Setup Options

If you want setup values to apply only to the current editing session, use the Close button. If you want setup values to be saved so they can be loaded and reused, use the Save button. To retrieve saved setup values, use the Load button.

**Note:** You must place a check in the checkbox for a setup page to be saved.

When choosing a folder, it is important to understand the order in which W-Edit will search for and load files. Refer to “Order of Preference for Setup File Locations” on page 39 for details.

**Save**
Place a check in the checkbox for each setup page you want to save and click the Save button. W-Edit will save the TCL files to the folder you specify.

**Load**
Place a check in the checkbox for each setup page you want to load and click the Load button. W-Edit will load the TCL files from the folder you specify.

**To/from folder**
Use this drop-down list to select the folder to which a setup file will be saved, or from which a setup file will be retrieved. You can also browse to a directory of your choice to save or load a setup file. Setup scripts are not required to exist in either location, in which case the W-Edit default values are used.

*{project setup folder}* is the default setup folder in the project folder of the specific simulation shown in the title of the Setup dialog.

*{user preferences folder}* is a setup folder in the Tanner directory on a local computer from which W-Edit will automatically read user-defined setup values. Settings from the user preferences folder overwrite the default settings for a simulation.
If you have made setup changes without saving them, W-Edit will prompt you to confirm that choice.

**Order of Preference for Setup File Locations**

You can save as many setup configurations as you like. This is useful, for example, so that setup definitions can be copied to a new chartbook, or when you want to save setup values that differ from those W-Edit loads automatically. Scripts corresponding to setup dialog settings have special names as well as special folders. Putting other types of scripts in those folders will have no effect.

If a script of the same name is present in both locations, settings from the user preferences folder take precedence over settings from the project folder.

Scripts saved to the user preferences folder are executed on each new chartbook.

When you run a simulation the results are written to a .tsim file and a data folder. For example, simulation of RingVCO.sp will produce RingVCO.tsim and a folder named RingVCO.

When you save setup changes to the project setup folder, the setup folder is written in the same folder as the chartbook if there is one, otherwise to the same folder as the first .tsim file in the active chartbook.

Also, only a preferences folder should get created and he tcl files should go in the preferences folder. All options are now preferences, so a technology folder should not get created.

**Project Setup Folder Location**

The predefined location of the chartbook setup folder is `\<project location>\ <chartbookname>\`, where `project location` is the path and directory where the chartbook is saved.
When saving to the project setup folder, if a chartbook file exists, then in the location of the chartbook file, a folder is created called `scripts\open.chartbook` and the tcl files are placed in the `open.chartbook` folder.

If no chartbook exists, then in the location of the tsim file, W-Edit will create a folder called `scripts\open.chartbook` and place the tcl files in the `open.chartbook` folder. When opening a chartbook, the tcl files that are in the `scripts\open.chartbook` folder should be run. The association is to simply look for the scripts folder that is in the location of the chartbook file.

This means that setup scripts in the `scripts\open.chartbook` folder are run when any chartbook is opened from the same folder as the scripts subfolder. To create scripts that apply just to one chartbook, that chartbook has to be the only one in its parent folder.

When you save a chartbook, if a .cbk file does not already exist, the **Save As** dialog will default to the same folder as the first .tsim file in the active chartbook. If you perform a Save As, and a .cbk file already exists, then the location will default to the folder of that .cbk file.

**User Preferences Folder Location**

The predefined location of the user preferences folder is `C:\Documents and Settings <username>\Application Data\Tanner EDA\scripts\open.chartbook`, where `username` is the login name of the current user. You may, however, place it in any location you choose.

Note that scripts saved to the user preferences folder are executed on each new chartbook.
Running Scripts when Opening and Closing W-Edit or a Chartbook

You can set W-Edit to automatically run scripts when you open or close the application, when you open any chartbook or when you open a specific chartbook, as follows.

- To load a script when a specific chartbook is opened, place it in the `scripts\open.chartbook` folder for that chartbook.
- To load a script when any chartbook is opened, place it in `C:\Documents and Settings\<username>\Application Data\Tanner EDA\scripts\open.chartbook`.
- To load a script when W-Edit is started, place it in `C:\Documents and Settings\<username>\Application Data\Tanner EDA\scripts\startup.wedit`.
- To load script when W-Edit is shutdown, place it in `C:\Documents and Settings\<username>\Application Data\Tanner EDA\scripts\shutdown.wedit`.

Exiting W-Edit

Use `File > Exit` to close all open chartbooks and files and exit W-Edit. You will have the option to exit with or without saving files that have been modified but not saved.
File Input and Output Types

W-Edit opens and reads .tsim files generated by T-Spice. The simulation data is stored in a separate .tsdat file, and all other application settings are saved in various files in the same folder for a given simulation run.

These settings in S-Edit running a transient analysis on the cell RingVCO_TestBench in the design RingVCO:

<table>
<thead>
<tr>
<th>General</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference Temperature (deg, C)</td>
<td>Default</td>
</tr>
<tr>
<td>Accuracy and Performance</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Simulation Outputs</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Show Waveforms</td>
<td>During</td>
</tr>
<tr>
<td>Enable Waveform Voltage Probing</td>
<td>true</td>
</tr>
<tr>
<td>Enable Waveform Current Probing</td>
<td>true</td>
</tr>
<tr>
<td>Enable Waveform Charge Probing</td>
<td>False</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>File and Directory Names</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SPICE File Name</td>
<td>../SimulationResults/$(Design)/$(Cell)/$(Name).sp</td>
</tr>
<tr>
<td>File Search Path</td>
<td>../Process/Generic_250nm/Generic_250nm_Tech</td>
</tr>
<tr>
<td>Include Files</td>
<td></td>
</tr>
<tr>
<td>Library Files</td>
<td>&quot;Generic_250nm.lib&quot; TT</td>
</tr>
<tr>
<td>Verilog-A Search Path</td>
<td></td>
</tr>
<tr>
<td>Simulation Results Folder</td>
<td>../SimulationResults/$(Design)/$(Cell)/$(Name)/</td>
</tr>
<tr>
<td>Keep all simulation results</td>
<td>False</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>External Simulator Setup</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulator</td>
<td>T-Spice</td>
</tr>
<tr>
<td>Simulator Command</td>
<td></td>
</tr>
</tbody>
</table>

Yield these files in T-Spice:
– and these files that W-Edit uses.

### Working with Text Files

The commands **File > New > New File**, **File > Open > Open File**, **File > Save > Save File** and **File > Close > Close File** behave conventionally.

### Help and Diagnostics

#### W-Edit Documentation

In addition to this manual in PDF format, W-Edit is shipped with application notes, release notes and a T-Spice tutorial that highlights basic schematic entry and editing operations.

W-Edit Product Support and Diagnostics

Use Help > About W-Edit—W-Edit to view the product version you are using.

Use Help > About W-Edit—Memory to display the operating system, processor and memory information for the computer you are using.
3 Viewing Simulations

Terminology for the Basic W-Edit Components

Simulation

A W-Edit simulation is the set of results from a given T-Spice simulation run. The default name for a simulation generated by Tanner EDA’s T-Spice or S-Edit is the SPICE file name followed by a timestamp for the dataset, in the following form. (Time is expressed using a 24-hour clock):

SPICEfilename_yyyymmdd_hhmmss

Note that in order to save each set of simulation results to a unique folder you must set the Keep all simulation results field in the S-Edit Setup SPICE Simulation—General dialog to True.

It is this “True” setting that appends a timestamp and creates the related folder, otherwise S-Edit will overwrite the results of a given simulation each time you run it. (See “Preserving or Overwriting Simulation Files” on page 51 for how this setting interacts with related settings in T-Spice and W-Edit.)

Chartbook

A chartbook file (extension .cbk) contains information related to how W-Edit constructs a visible representation of simulation data, including the charts and plots that have been created, color, line type, units, labels, analysis types and parameter values, along with a pointer to the name and path of the data file that contains the numerical simulation data. A chartbook can contain multiple charts. Note that W-Edit does not save the simulation data to a chartbook file.

Chart

A chart is the viewing window in W-Edit for a given simulation. Each chart contains a single independent variable as the y-axis unit and a legend for each trace it displays. It can contain one or more plots, each with a different y-axis, and display the results of one or more simulations.

You can pan back and forth and zoom in and out of chart views, including specifying the exact x-y coordinate range that W-Edit displays. You can also use the mouse to measure positions and distances between points easily and precisely.

A chart can only be saved as part of a chartbook.

Plot

If simulation results require more than one set of axes, each axis set is displayed in a different plot.

A plot is one set of x- and y-axes within a chart. A chart can contain multiple plots, but all plots in a chart must have the same x-axis. Each plot can only contain traces with the same measured quantity and associated units.

Trace

A trace represents a single simulation run for a given dependent variable.
You can view traces from a given simulation in a single chart or in multiple charts. When you first load a simulation, W-Edit displays the traces that use the same x- and y-axis system in a single chart. If the simulation has traces with different axis systems, W-Edit creates multiple charts or plots.

Traces corresponding to multiple simulation runs for the same dependent variable are shown on the same chart. Each chart contains either a curve or an independent trace.

**Curve**

Traces from a `.step` or `.alter` command contain all the parameter sweeps or `.alter` simulations for a dependent variable. Each individual sweep is represented by a single curve.

Each of these curves in such a trace, often called a *trace family*, can be edited individually, but all trace family members share the same label and a single entry in the Trace navigator.

**Working with Chartbooks**

A set of charts can only be saved as part of a chartbook. W-Edit creates a new chartbook for the first run if a given simulation, with the SPICE filename as the default chartbook title, and adds the new chartbook to the drop-down list at the top of the simulations navigator. Subsequent simulation runs are added to the chartbook. You can also create a new chartbook explicitly at any time.

**Chart Display Characteristics**

The various parts of the chart are identified as shown below:
Setting Default Chart Display

Use **Setup > Chart Styles** to set global default display characteristics for the various chart areas.

![Setup - Chart Styles](image)

**Titles**

Text entered in these fields will display in the upper left, center and right margins respectively. The default values are:

- **Left** – Product name and version number (tcl command `[workspace getactive -simversion]`).
- **Center** – Full path and file name of result file on display (tcl command `[workspace getactive -simfile]`).
- **Right** – time and datestamp (tcl command `[workspace getactive -simdate]`).
Chapter 3: Viewing Simulations

Setting Individual Chart Display

Chart colors can be changed for an individual chart by right-clicking in the chart and selecting Chart Properties or by using Chart > Chart Properties.

Legend detail
Select which elements are displayed in the chart legend.

- **None** – No legend items are displayed.
- **Traces** – Only one legend item per trace is displayed. The legend item label contains only the base name of the trace with all qualifiers omitted. Selecting a legend item selects all curves whose base names match the legend item label. When selecting curves, the legend items will not be selected unless all of the curves corresponding to the legend item label are selected.

When consecutive subset labels would have the same trace name, the trace name is only displayed in the first label and is suppressed in trailing labels. Controlled by the following tcl command (tcl command [test chartsetting -name AbridgeConsecutiveTraceNames -value 0]. The default value is 1, true.

- **Curves** – The legend item labels are abbreviated as follows: 1) All name/value qualifiers common to all curves are omitted. 2) All qualifier names common to all curves have the name=value present in the first item but only the value present in the remaining items. 3) All remaining qualifiers are added to the end of the legend item label in the name=value format.
- **Full** – Show the full item legend.

Legend position
Select where the chart legend is positioned within the chart window from the options Top, Bottom, Left or Right.

Show cursor tables in charts
When checked, cursor table values are displayed in the chart window.

When legend items on a chart have a one-to-one correspondence with the chart curves, if a chart has many curves most of the chart area will be consumed by the legend. The maximum number of trace legends that can be placed in a chart is 50. Use the Legend settings to control display characteristics that will reduce the legend size.

Colors
Use these fields to pick the display color for the Background, Margins, Grid, Axes and Title areas.

Grid

- **Use Direction** to pick whether None, Horizontal, Vertical, or Both of the axes are displayed.
- **Use Line style** to display axis lines as Thin, Thick, Dotted, or Dashed.

Digits
Enter the number of decimal places that will be displayed in the axis values for the x-axis, y-axis and cursors. (Significant) digits refer to the total number of digits allowed in an engineering formatted number. (Decimal) digits refer to the number of digits following the decimal point in an engineering formatted number.

Reset to default
Sets all values to the Tanner shipping default values.

W-Edit 16 User Guide
Changes made in the **Chart Properties** dialog apply only to the selected chart. In **Chart Properties**, blue text indicates a property inherited from the default settings and black text indicates a local override.

**Creating a Chartbook**

To create a new empty chartbook, use **File > New > New Chartbook** or the **New Chartbook** ( ) icon. You can then use the **Add** button to add simulations to the chartbook.

**Opening a Chartbook**

To open an existing chartbook (extension .cbk), use **File > Open > Open Chartbook** or the **Open Chartbook** ( ) icon. Multiple chartbooks can be open at the same time.

When you open a .cbk file, W-Edit automatically loads the associated data along with the saved chart information and simulations.

**Warning:**

If a data file is deleted, renamed, or moved to a different directory, or if the column headings have changed, W-Edit will be unable to open any .cbk file that points to that data file.

**Closing a Chartbook**

To close a chartbook, use **File > Close > Close Chartbook**. If the chartbook has been modified W-Edit will prompt you to save it. You can also use **File > Close > Close All Chartbooks**.
**Saving a Chartbook**

*File > Save > Save Chartbook “nnn”* (or the *Save Chartbook* button) saves information on chart, trace, axis, and environment settings in files with the `.cbk` (chartbook) extension. Calculated traces are also stored in the work space, in the form of the expressions used to calculate them. However, the `.cbk` file does not save the experimental data.

**Using the Simulations Navigator**

The Simulations navigator lets you open and close chartbooks, and add and remove simulations.

![Simulations navigator](image)

(Chartbook drop-down) Lists the chartbooks currently open.

(Simulations navigator) Lists the title and timestamp of all simulations currently open in the selected chartbook.

**Add**

Opens a standard Windows *Open* dialog so you can search for and add `.tsim` simulation files. Multiple simulation files can be open at the same time.

The specified `.tsim` file is added to the current chartbook unless one is not open. If no chartbook is open, a new one will open.

**Remove**

Removes the highlighted simulation(s) from the current chartbook.

*Use Ctrl + right-click* to highlight multiple simulations.

Note that there is no confirmation or second step when you press this button.

**Right-Click Menu in the Simulations Navigator**

This context-sensitive menu controls trace display in the active chart by simulation.

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hide Simulation</td>
<td>Hides traces from the highlighted simulation(s) so that they are not displayed in the active chart.</td>
</tr>
<tr>
<td>Hide All But Selected Simulations</td>
<td>Hides traces from all but the highlighted simulation(s) so that they are not displayed in the active chart.</td>
</tr>
</tbody>
</table>
Preserving or Overwriting Simulation Files

There are several ways to control whether W-Edit merges or overwrites simulations.

W-Edit default behavior is to create a new chartbook when displaying a new simulation with a different name from other simulations that are currently open.

Therefore, W-Edit does not create a new chartbook for a new simulation submitted from S-Edit, where circuits are given cell names. Traces from the new simulation are added to the existing chartbook.

Saving revisions from S-Edit

To keep multiple revisions of a given simulation, you can set the S-Edit option “Keep all simulation results” to True in Setup > SPICE Simulations. This appends a timestamp to the simulation name (as set in the SPICE File Name field) and saves it to a folder that also has the timestamp.
Saving revisions from T-Spice

Alternately, you can check the same option in **T-Spice** under **Setup > Simulation Settings**. (The “keep all simulation results” in S-Edit saves to a single name in T-Spice, not as separate simulations)

Saving revisions from W-Edit

In W-Edit, use **Create a new chartbook for an incoming simulation with a different base name** in **Setup > General**. When this is **not** checked, W-Edit opens simulations in the active chartbook, so if you run a simulation on a different circuit, you will see traces from both circuits.

When this is **checked**, W-Edit opens a simulation in the active chartbook if the filename matches. Otherwise, it creates a new chartbook.

Using the Variation Navigator and Chart Parameters

The Variation navigator works together with the Chart Parameters window to filter and sort the trace display, and to let you set dependent variables so they can be assigned display characteristics.

You must either open or **activate** (bring into focus or into the fore) the Chart Parameters window in order to enter the required variables. W-Edit will display the appropriate parameter fields depending on the type of chart you have open.

For parametric plots, W-Edit will automatically select the independent variable based on the variable that has the most data members.
Chapter 3: Viewing Simulations Working with Chartbooks

Right-Click Menu in the Variation Navigator

This context-sensitive menu controls trace display in the active chart by variable value.

- **Hide Selected Variations**
- **Hide All But Selected Variations**
- **Show Selected Variations**
- **Show All Variations**
- **Traces Style...**

**Hide selected variations**
Hides traces from the highlighted variation(s) in the active chart.

**Hide all but selected variations**
Hides traces from all but the highlighted variation(s) in the active chart.

**Show selected variations**
Shows traces from the highlighted variation in the active chart.

**Show all variations**
Shows traces from all variations in the active chart.

Variable 1, Variable 2, Variable 3, Bin Size, Period, Offset, Chart Width, etc.

Use these fields to control chart display by setting an index parameter for up to three dependent variables, so they can be assigned a display characteristic. (See “Display Settings for Traces” on page 72.)

Variation Navigator

Select **Filter** to enable filtering and enter alpha-numeric characters in the entry field to limit the display of variables.

Chart Parameters

W-Edit displays the appropriate parameter fields depending on the chart type that is active.

Use **View > Chart Parameters View** to open this window.
Working with Charts

Chart Axes

W-Edit supports a rectangular axis system with three types of x-axis coordinate systems: Data, Time, Parametric and Frequency, where the data is a measured quantity and associated units, such as Voltage (V), Current (A), Noise (V), etc.

W-Edit supports non-monotonically increasing step data, and allows you to plot one signal against another so the x-axis can be defined as one of the traces instead of time. For example, many MOS capacitors are non-linear; it is good to look at their capacitance for a given voltage across the capacitor. A transient simulation can be written to put a fixed amount of current onto the capacitor over a given time. The capacitance can then be calculated by \( C = \text{InputCurrent} \times \frac{\text{derivative(Time)}}{\text{derivative(VoltageAcrossCap)}} \). The graph of capacitance versus time does not show anything useful, but a graph of capacitance against the voltage on one plate of the capacitor would give you the useful C vs. V chart.

Chart Legends and Unit Display

Every trace has associated axes that are automatically determined from the data file, and only one axes type can be represented on a given plot.

Each chart contains a single, unique y-axis system. An axis system consists of grid lines, tick marks (on the left and bottom sides), numbers associated with the tick marks, labels and units for the x and y-axes, and a legend for each trace or curve that is loaded. Each chart is restricted to a single axis scale, either logarithmic or linear.

W-Edit reads information in the simulation data file to automatically assign the x-axis system, and uses built-in ratios to determine the scale of the axis system and the positions of the title, labels, and legends in a chart. When you change the chart size, W-Edit scales the size and position of chart elements accordingly.

Once an axis system is assigned, incompatible traces (traces with units that do not match the x-axis system) cannot be added to the chart. W-Edit will instead open a new plot within the same chart window.

For expressions in which they are defined in the node name, T-Spice can track and output units so they are displayed in W-Edit. Note that this does not apply to units in print statements.
Setting Text Size in a Chart

Use **Setup > Chart Text** to set the overall text size in the chart window, and to scale the size of various text elements individually.

Text size

Choose Small, Medium, or Large for the general font size.

Font

Use the pull-down menu to choose a font face from any true type font installed on the local machine.

Individual elements

Use the slide controls to adjust the scale of individual elements relative to the general font size. Applies to text in the “Chart title,” “Axis title,” “Axis Numbers,” “Legend,” “Labels” and “Cursor table” areas.

Tool Tip Font

Launchs a standard Font dialog that applies to text in the data point tool tips (see “Data Point Tool Tips” on page 85).

Reset to default

Restore font size and scales to the default shipping settings.

You can also right-click in the chart window area and click on **Chart Text** to enter custom text settings for the active chart. This dialog provides the same parameters for controlling the font face,
general font size, and individual element font size as the setup dialog. Blue text indicates settings that are the same as the setup values, black text indicates that the setting is different from the setup value.

Adding a Chart to a Chartbook

**Chart > New Chart** adds a new, untitled chart of the analysis type you select. The new chart opens in the selected state and all other charts in the active window are deselected. You can also copy a chart and all its traces with **Chart > Duplicate Chart**.
Chapter 3: Viewing Simulations

Working with Charts

Selecting and Deselecting Chart Objects

Most operations and commands work only on selected objects within the active chart window. W-Edit defines four types of objects: traces, cursors, markers and text labels.

To select an object with the mouse, simply click on it in the chart window. When you select an object with the mouse, all other objects in the chart window are automatically deselected. To add an object to the selected group, press and hold the Shift key as you click on the object.

Use Ctrl+A to select all objects in the chart window and Alt+A to deselect all objects in the chart window.

You can also use the Edit > Select and Edit > Deselect commands to select all objects of a given type — Cursors, Markers, Text Labels, Traces or All objects.

Note that Edit > Select clears the previous selection. When one object type is selected, if you select another the first object type will be deselected.

Properties Navigator for Charts and Traces

Each chart and trace has a set of properties associated with it. The Properties navigator displays the system-level property values and lets you edit the user-specified property values of whatever is

<table>
<thead>
<tr>
<th>Chart Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transient</td>
<td>Chart for transient analysis data with axes Voltage, Current or Charge (y-axis) vs. Time (x-axis).</td>
</tr>
<tr>
<td>AC</td>
<td>Chart for alternating current analysis containing plots of magnitude (dB) vs. frequency and phase (deg) vs. frequency. For AC analysis the default x-axis is logarithmic.</td>
</tr>
<tr>
<td>XY</td>
<td>Chart for direct current data with axes Data (y-axis) vs. Frequency (x-axis).</td>
</tr>
<tr>
<td>Parametric</td>
<td>Chart for direct current data with axes Data (y-axis) vs. Frequency (x-axis). A parametric plot allows you to plot one trace against another using one trace as the x-axis. From a parametric plot you can create a scatterplot, which plots the datapoints without connecting them with a line.</td>
</tr>
<tr>
<td>Histogram</td>
<td>A histogram groups the frequencies of a data set into intervals with an area equal to the density of data point in the interval. (See “Histograms” on page 61.)</td>
</tr>
<tr>
<td>Eye</td>
<td>Chart for displaying a transient curve when repetitively sampled and applied to the vertical input, where the data rate is used to trigger the horizontal sweep. (See “Eye Diagrams” on page 62.)</td>
</tr>
</tbody>
</table>
currently selected. If a trace is not selected, W-Edit will display the chart properties, you can also click in the background area to display chart properties.

**Duplicating a Chart**

You can duplicate a chart with all its traces and display characteristics with Chart > Duplicate Chart.

**Renaming a Chart**

Use Chart > Rename to change the name of an existing chart.

**Exporting Chart Data**

Chart data can be exported for inclusion in external applications. File > Export Chart Data exports all trace data present in the active chart as a tab delimited (.txt) text file, a comma delimited (.csv) text file, or a piece-wise linear (.pwl) file.

The default precision number of significant digits exported for each trace is 12. You can use the following tcl command in the command window to change the format string to six; use **%19.11e** to return it to twelve. Note that setting the format string to an invalid value may cause W-Edit to crash during export.

```plaintext
test chartsetting -name ExportFormat -value {%g}
```
There is also a TCL variable “tanner_digits” that controls the number of significant digits displayed in the conversion of doubles to strings in the command-line for best appearance when printed, with a default value of 8 (including digits on both sides of the decimal point.)

TCL_BIND_INT(tanner_digits, 8);

See also “Exporting Trace Data” on page 94 for how to export data for an individual trace.

Copying or Printing a Chart Image

You can use **File > Image > Copy to clipboard** to copy the selected traces and objects to the clipboard in regular metafile format (.wmf) or portable network graphic (.png) format. Sometimes traces that do not fully fit into a plot will expand off the chart or into another plot when pasted. In this case the .png format works best. (The tcl commands generated are chart image - format <wmf, png>.)

Alternately, use **File > Image > Save to file** to create an image of a chart which can then be copied or imported to another program.

In the Command window, the **chart image** command uses the following options:

- **-filepath <filename>**
  - If present, the image is written to a file, otherwise, it is written to the clipboard.

- **-format <bitmap type>**
  - If present, use this value for the format. Valid values are “jpg”, “jpeg”, “bmp”, “wmf”, “emf”, “png” and “svg”. If not present, use the extension of the filename. If no filename is present or the extension is empty, use metafile format.

- **-sizex, -sizey**
  - The size is in pixels (or in 0.01 mm, if metafile). If both of these are missing, we use the chart size on the screen. If one of these is missing, W-Edit will construct it by scaling the other according to the aspect ratio of the chart on the screen. Note that when inserting a W-Edit svg file into other file types such as a spreadsheet document, the output chart is usually much smaller than that displayed in W-Edit (on average 1/10 to 1/20 the width and height or 1/100 to 1/400 the area).

W-Edit also has a modified version of the Windows standard print dialog under **File > Print**. The active chart will print in color on a white background unless you select **Print in black on white**.
**File > Print Preview** from any dialog opens a page preview in the W-Edit chart window.

**Closing a Chart**

**Chart > Close Chart** closes the active chart, **Chart > Close All Charts** closes all open charts.
**Histograms**

W-Edit provides histogram charts that group the frequencies of a data set into intervals with an area equal to the density of data point in the interval.

The “trace define” tcl command automatically adds an index number for the trace domain when the “-data” parameters references an array of values. The following tcl command:

```
trace define -name RiseTimeOutV1 -data [calc risetime(trace(Out:V))]
```

will generate an X - Y chart of the rise time values of Out:V when generating a new chart. By adding a “-xunits samples” as follow:a new histogram chart will be generated instead:

```
trace define -name RiseTimeOutV2 -data [calc risetime(trace(Out:V))] -xunits samples
```
**Eye Diagrams**

W-Edit provides the following fields in the Chart Parameters window for eye diagrams. Note that it is important to include the x-axis variable in the **Period** field.

<table>
<thead>
<tr>
<th>Eye Diagram Parameters</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Period</strong></td>
<td>$0.00000000m</td>
</tr>
<tr>
<td><strong>Offset</strong></td>
<td>$0.00000000m</td>
</tr>
<tr>
<td><strong>Chart Width</strong></td>
<td>$70.00000000m</td>
</tr>
</tbody>
</table>

**Period** - the width of the portion that will overlap in the eye chart

**Offset** - the distance from zero where the eye chart begins

**Chart Width** - the width of the viewing window. S-Edit supplies a default minimum of 140% of the Period value and the W-Edit limit is .25*Period

---

**Working with Plots**

W-Edit can open up to sixteen plots for a given chart. When adding a trace to a chart will cause the number of plots to exceed the maximum, W-Edit will provide a warning and will not add the trace.

**Opening a Plot**

Each plot is restricted to a single, unique y-axis data type and axis scale (log or linear). Thus when you add a trace to a chart that has a different y-axis data type, W-Edit will automatically display it in a new, compatible plot.
If you try to add a trace to a plot in which it cannot be displayed, for example AC results onto a transient plot, W-Edit will alert you to the incompatibility with the message:

# WED Error: Trace does not match any data compatible with chart: {name}

Traces are grouped for display by trace family, then y-axis data type, then all traces with the same x-axis units. See “Expanding and Collapsing Traces” on page 80 for the complete algorithm by which W-Edit opens new plots.
As simulations are run or added, W-Edit dynamically updates the chart display. A chart will be updated if it can be appended according to the following criteria:

[1] No two plots have the same units class.

[2] All trace expressions are simple (i.e. consist only of the trace name with no qualifiers).

[3] There are no cursors, markers or labels on the chart.

The first chart found to be appendable has the printed traces added to it. A new chart is created only if all existing charts cannot be appended.
Adding a Plot to a Chart

Use Chart > Plot > New Plot to add a new, empty plot to a chart. The new plot will have the same x-axis but the y-axis is determined by the first trace added. You can also right-click in the chart window and click on Insert Plot in the context menu.

Moving a Plot

The Chart > Plot > Move menu command lets you change the position of a plot either relative to its current (source) position, using Move up by or Move down by, or to an absolute position using Move to plot number, where the lowest number, 1, is at the top.

When a plot is inserted in the target location, all other plots maintain their relative positions.

If you initiate a move without a plot selected, you must specify which plot to move in the Source Plot field, by entering the position number of the plot (where 1 is at the top). When you initiate a move with a plot selected, the Source Plot value is given and so disabled.

You can also use Shift+Up arrow and Shift+Down arrow to move a plot up or down, or Ctrl+Up arrow or Ctrl+Down arrow to move curves in a stacked plot up or down.

You can also right-click in the chart window and use Move Plot in the context menu to select either Up, Down, or Move to open the Move Plot dialog.
Removing a Plot from a Chart

Use **Chart > Plot > Delete** to remove a plot from the active chart by entering its plot position. If there is more than one plot in the chart window a dialog will prompt you to specify which of the plots to delete, where 1 is the top position.

You can also right-click in the chart window and use **Delete Plot** from the context menu.

![Delete Plot dialog](image)

Removing Empty Plots

**Chart > Plot > Delete Empty Plots** simply removes all plots that contain no traces.

Changing How a Plot is Scaled or Sized

To resize a plot, place the mouse over the x-axis and drag up to make the plot smaller or down to make the plot larger.

A context-menu item “Reset Plot Heights” will restore all plots to the same size.

You can also right-click in the plot area and choose **Plot Properties** to rescale the x- and y-axes independently, set both the x- and y-axis to show linear or logarithmic scales, add a plot **Title** and label its **Units**, and set the display range of each axis independently to scale the visibility and spacing of curves.

![Plot Properties dialog](image)
Chapter 3: Viewing Simulations

Stacked Plots

For transient plots only, you can use the Plot Properties dialog to set the display Type to Stacked, as shown below, which displays curves separately one above the other. Stacked displays are useful for digital signals.

---

Title

Use this field to enter any text you prefer to display in the title that precedes the unit display for each axis. When blank, displays the default axis unit (Coulombs, amps, volts, etc.)

Units

Use this field to enter a unit value that will be displayed in the title area. The default display is the unit, without scale.

Scale

Use this pull-down menu to select a non-default unit display for the axis tick marks.

Show units scale in title

When checked, the unit scale will be displayed in the title area. When unchecked, the unit scale is displayed with the values along the axes.

Type

Normal – chart for displaying a transient curve when repetitively sampled and applied to the vertical input, where the data rate is used to trigger the horizontal sweep. (See also “Eye Diagrams” on page 62.)

Stacked – for transient charts only, scales the minimum to maximum height of each curve to equal height and stacks them on top of each other in the same plot. (See “Stacked Plots,” below.)

View Port

You can select either a Linear or Logarithmic axis. You can also use the Min and Max fields to set the extent of the axis display numerically.

---

Stacked Plots

For transient plots only, you can use the Plot Properties dialog to set the display Type to Stacked, as shown below, which displays curves separately one above the other. Stacked displays are useful for digital signals.
Chapter 3: Viewing Simulations

Chapter 3: Viewing Simulations

Note that because each curve may be scaled differently, if you add a text label to a plot while it is in stacked display mode, it will not be displayed. When you toggle to normal display type you will be able to see and edit the text label as usual. Marker and cursor behavior is normal.

You can use **Ctrl+Up** arrow or **Ctrl+Down** arrow to move the plots that contain selected curves up or down, or you can use the menu commands **Chart > Move Curves > Move Curves Up** and **Chart > Move Curves > Move Curves Down**. Similarly, the keystroke combinations **Shift+Up** arrow or **Shift+Down** arrow move curves in a stacked plot up or down. If these keyboard shortcut are not working, we recommend that you right-click on any toolbar, choose **Customize**, then press the **Reset All** button in the **Keyboard** tab. (See “Adding and Displaying Keyboard Shortcuts” on page 20 for more information.)

**Right-Click Menu for Plots, Charts and Traces**

You can access the plot commands by right-clicking anywhere in the chart window to open the context-sensitive menu shown below. The domain of action for these context-sensitive commands is the plot over which the cursor is located.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Insert Plot</strong></td>
<td>Adds a new, empty plot to the active chart.</td>
</tr>
<tr>
<td><strong>Delete Plot</strong></td>
<td>Removes the selected plot from the active chart.</td>
</tr>
<tr>
<td><strong>Move Plot</strong></td>
<td>Lets you move the active plot up or down in the chart window, and lets you open the <strong>Move Plot</strong> window. (See “Moving a Plot” on page 65.)</td>
</tr>
<tr>
<td><strong>Move Curves</strong></td>
<td>For stacked plots, a selected curve can be moved up or down in a plot.</td>
</tr>
<tr>
<td><strong>Chart Properties</strong></td>
<td>Opens the <strong>Chart Properties</strong> window (see “Properties Navigator for Charts and Traces” on page 57.)</td>
</tr>
<tr>
<td><strong>Chart Text</strong></td>
<td>Opens a window where you can adjust plot text sizes (see “Setting Text Size in a Chart,” below.)</td>
</tr>
<tr>
<td><strong>Plot Properties</strong></td>
<td>Opens the <strong>Plot Properties</strong> window (see “Changing How a Plot is Scaled or Sized” on page 66.)</td>
</tr>
<tr>
<td><strong>Reset Plot Heights</strong></td>
<td>If you have resized any plot windows, restores them to their original size.</td>
</tr>
</tbody>
</table>

**Insert Plot**

**Delete Plot**

**Move Plot**

**Move Curves**

**Chart Properties**

**Chart Text**

**Plot Properties**

**Reset Plot Heights**
Working with Traces and Curves

A trace contains all data from all sweeps and all simulations for a given node in the circuit. Each trace has a name, labels, and four possible display properties: color, contrast, line type and point type (see “Display Settings for Traces” on page 72).

W-Edit can take any number of traces in a chart. A maximum of 50 traces will be automatically placed on a chart following a simulation, however more than 50 can subsequently be loaded. When the number of traces in a chart exceeds the maximum (as determined internally for each), the trace legends will be suppressed. When this occurs, W-Edit sends a message to the Cursors window stating: “Too many curves.”
If a trace contains an infinite value, an undefined double value or a value that is not a number, W-Edit will simply omit that point from the trace, causing a break in the chart where the missing point is not plotted.

**Trace Names**

The name of a trace is determined from the data file; you cannot edit trace names in W-Edit. The fully qualified name is displayed in the status bar when a trace is selected, in the following format:

```
Curve trace (trace name, variable 1=value, variable 2=value, ...
```

Simulation runs in a curve are identified by the numeric value of the swept parameters. (For example, *alter* is treated as a parameter.) W-Edit names individual traces by appending the trace label with parameter values, separated by commas. Parameter values are listed in the order in which they were applied in the simulation.

Trace names that include special arithmetic characters are enclosed in double quotes when needed to prevent an error in W-Edit.

**Curve Display Units**

W-Edit can display curves having the following units of measure.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amps</td>
<td>A</td>
</tr>
<tr>
<td>Amps2</td>
<td>A^2</td>
</tr>
<tr>
<td>AmpsPerM</td>
<td>A/m</td>
</tr>
<tr>
<td>AmpsPerM2</td>
<td>A/m^2</td>
</tr>
<tr>
<td>AmpsPerRtHz</td>
<td>A/Hz^(1/2)</td>
</tr>
<tr>
<td>AmpsPerV2</td>
<td>A/v^(1/2)</td>
</tr>
<tr>
<td>Bool</td>
<td>boolean</td>
</tr>
<tr>
<td>CM2perVolt</td>
<td>cm^2/v</td>
</tr>
<tr>
<td>Coulombs</td>
<td>C</td>
</tr>
<tr>
<td>Decibels</td>
<td>dB</td>
</tr>
<tr>
<td>Degrees</td>
<td>deg</td>
</tr>
<tr>
<td>DegreesC</td>
<td>degC</td>
</tr>
<tr>
<td>DegreesF</td>
<td>degF</td>
</tr>
<tr>
<td>DegreesK</td>
<td>degK</td>
</tr>
<tr>
<td>Ev</td>
<td>eV</td>
</tr>
<tr>
<td>EvPerDegree</td>
<td>eV/deg</td>
</tr>
<tr>
<td>Farads</td>
<td>F</td>
</tr>
<tr>
<td>FperM</td>
<td>F/m</td>
</tr>
<tr>
<td>FperM2</td>
<td>F/m^2</td>
</tr>
<tr>
<td>Henrys</td>
<td>H</td>
</tr>
<tr>
<td><strong>Parameter</strong></td>
<td><strong>Units</strong></td>
</tr>
<tr>
<td>---------------</td>
<td>------------</td>
</tr>
<tr>
<td>Hertz</td>
<td>Hz</td>
</tr>
<tr>
<td>Joules</td>
<td>J</td>
</tr>
<tr>
<td>Meters</td>
<td>m</td>
</tr>
<tr>
<td>Meters2</td>
<td>m*2</td>
</tr>
<tr>
<td>Mhos</td>
<td>mhos</td>
</tr>
<tr>
<td>MperSecond</td>
<td>m/s</td>
</tr>
<tr>
<td>Ohms</td>
<td>ohms</td>
</tr>
<tr>
<td>OhmsPerSq</td>
<td>ohms/sq</td>
</tr>
<tr>
<td>PerAmp</td>
<td>1/A</td>
</tr>
<tr>
<td>Percent</td>
<td>percent</td>
</tr>
<tr>
<td>PerCentimeter</td>
<td>1/cm</td>
</tr>
<tr>
<td>PerCentimeter2</td>
<td>1/cm^2</td>
</tr>
<tr>
<td>PerCentimeter3</td>
<td>1/cm^3</td>
</tr>
<tr>
<td>PerDegree</td>
<td>1/deg</td>
</tr>
<tr>
<td>PerDegree2</td>
<td>1/deg^2</td>
</tr>
<tr>
<td>PerMeter</td>
<td>1/m</td>
</tr>
<tr>
<td>PerMeter2</td>
<td>1/m^2</td>
</tr>
<tr>
<td>PerMeter3</td>
<td>1/m^3</td>
</tr>
<tr>
<td>PerVolt</td>
<td>1/V</td>
</tr>
<tr>
<td>Radians</td>
<td>rad</td>
</tr>
<tr>
<td>Seconds</td>
<td>s</td>
</tr>
<tr>
<td>Siemens</td>
<td>S</td>
</tr>
<tr>
<td>Squares</td>
<td>sq</td>
</tr>
<tr>
<td>Tesla</td>
<td>T</td>
</tr>
<tr>
<td>V2perHz</td>
<td>V^2/Hz</td>
</tr>
<tr>
<td>VperA</td>
<td>V/A</td>
</tr>
<tr>
<td>VperCm</td>
<td>V/cm</td>
</tr>
<tr>
<td>VperDegree</td>
<td>V/deg</td>
</tr>
<tr>
<td>VperRtHz</td>
<td>V/Hz^(1/2)</td>
</tr>
<tr>
<td>VperV</td>
<td>V/V</td>
</tr>
<tr>
<td>Volts</td>
<td>V</td>
</tr>
<tr>
<td>Volts2</td>
<td>V^2</td>
</tr>
<tr>
<td>VRt</td>
<td>V^(1/2)</td>
</tr>
<tr>
<td>Watts</td>
<td>W</td>
</tr>
</tbody>
</table>
### Display Settings for Traces

Use **Setup > Trace Styles** and **Selection Style** to set global display characteristics for traces.

You can also use the right-click menu in the chart window to set **Chart Properties**, **Plot Properties** and **Trace Style**.

When you use the right-click menu in either the Trace navigator, the Variations navigator or the chart window to open the Trace Style dialog, you will override the setup values by applying display characteristics to just the selected traces or curves (see “Setting Display Style for an Individual Trace” on page 74.)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Webers</td>
<td>Wb</td>
</tr>
<tr>
<td>Samples</td>
<td>Samples</td>
</tr>
<tr>
<td>dBm</td>
<td>dBm</td>
</tr>
<tr>
<td>dBu</td>
<td>dBu</td>
</tr>
<tr>
<td>Undefined</td>
<td>“undefined”</td>
</tr>
</tbody>
</table>
Setting Trace Color, Line Type and Point Type

Use this dialog to pick which display characteristic – for example, color or line type – will change for each chart element – such as simulation or dependent variable. You can also set the granularity (that is, the number of values available) for a given display characteristic from one to ten.

Note:
Remember that for a setup page to be saved, you must place a check in the corresponding checkbox and press the Save button. See “Saving and Reloading Setup Options” on page 38 for more detail.

Display Characteristics

There are four display characteristics for each chart—color, contrast, line type (different line types such as solid or dashed, and line weights), and point type (different marker shapes for the individual data points on a curve). Traces are assigned display characteristics in the order in which they are read into W-Edit. The higher the contrast value, the brighter the display.

Granularity—Number of Values to Use

You can control the number of values used for each of the display characteristics by setting the value in the scroll box from 1 to 10. If the number of traces in a chart window exceeds the number of values for a display characteristic, the display cycles to the beginning of the list.

Note that W-Edit will not retain settings when you increase the number of available values after decreasing them.
Chart Elements

Each of the four display characteristics can be assigned to one of these chart elements—Simulation, Trace, Variable 1, 2 and 3 or Default. If you pick Default, all traces are assigned the same value (the first value) for the associated display characteristic.

How to Use the Trace Styles Setup Table

You start by selecting which chart element is differentiated by a given display characteristic. For example, if you want each trace to be a different color, you would select Trace in the Color drop-down list.

Next you pick the number of values you want available for the chart element before it cycles to the beginning of the queue. In this example, you are setting the number of colors. For example, if you want the traces display to cycle through the colors of the rainbow, you would set the granularity to 7, and pick red in the first color drop-down field, orange in the second one, yellow in the third, green in the fourth, etc.

You then repeat the process to differentiate another chart element using one of the remaining display characteristics. Note that you do not have to use all of the display characteristics or all of the chart elements. However, any chart element can only be assigned to one display characteristic – you cannot differentiate traces, for example, by both line type and color.

Setting Display Style for an Individual Trace

You can select one or more traces or curves and use the right-click menu in the Variations navigator, the Trace navigator, or the chart window to open Trace Styles, which allows you to change the color, line type and point type settings for just those objects that are selected.

Note that there is a hierarchy to how settings are applied from the different right-click menus. Settings made in the Trace Style dialog when it is accessed from 1) the chart window override settings in the same dialog when it is accessed from 2) the Trace navigator, which override settings made when it is accessed from 3) the Variations navigator, and settings made from any of these three right-click menus override the settings in 4) Setup > Trace Styles.

Example: Varying Curve Display for Multiple Variables

In this example Color is set to change as the trace changes, Contrast is set to change as Variable 1 changes, and Line type is set to change as Variable 2 changes.

In W-Edit Setup > Trace Styles, Color display is set to vary by Trace at a granularity of four, changing from magenta to green to blue to yellow. Contrast is set to vary by Variable 1 at a granularity
of three, changing from dark (100, or highest luminosity) to medium (85) to light (70). Variable 2 is set to vary by Line Type and Point Type is set to a Default of a tiny dot.

Setting the Appearance of Selected and Unselected Objects

You can control the color, line type and point type of curves and traces, and set the color of cursors, markers and text labels when they are in the selected state using Setup > Selection Style.

By default, an unselected text label is displayed in yellow, an unselected marker is displayed as a green X, a selected marker is displayed as a blue X enclosed in a circle, an unselected cursor is displayed as a thin yellow line and a selected cursor is displayed as a thicker dotted line.

The default color for selected traces, as shown below, is yellow. Selected cursors, markers and labels are all assigned the same color to indicated that they are selected. Either setting may be changed at any time.
The **Fade out** field applies to non-selected traces, where the lower the percentage, the dimmer all other curves will appear when one curve is selected. For example, if you set it to 0%, unselected traces will fade completely and have zero visibility. 20% is the suggested **Fade out** setting.

Using **Fade out** can slow redraw performance on traces with a large number of points. It also has the unrelated effect of disabling the evaluation of certain tcl commands in the text label name field (see “Panning and Zooming in a Chart” on page 82.)

**Using the Trace Navigator**

The Trace navigator will only show traces for the analysis of the plot that is active.

The Filter drop-down options, characters in the entry field and the trace type buttons are function as filters and work according to a Boolean OR. Thus, if you set multiple filter criteria, a trace meeting *any one of* the individual criteria will be included in the list.

The display color of a trace name indicates the trace type or status. A pale gold background indicates that not all traces are listed, the list is incomplete due to filtering.
The **Filter** option lets you enter letters or numbers in the adjacent entry field so the trace list will only show trace names that contain those characters, in the order entered. The remaining options in the pull-down menu provide the following filters:

- **Top-level**—filters the list to show only those traces that are from the top level device in the original schematic.
- **Arithmetic**—filters the list to show just arithmetic traces.
- **In use in chart**—filters the list to show only those traces that are displayed in the active chart.
- **Regular expression**—filters the list to show just those traces that meet the criteria expressed in the regular expression entered.
- **Wildcard**—choosing this option allows you to use wildcards in the filter field. Use an asterisk (*) to select any number of unspecified characters. Use a question mark (?) to select any single unspecified character.

**Note:** If you apply more than one criteria using filters, a trace will be listed if it meets any of those criteria.

**View traces hierarchy**

- Toggles the display list to show traces according to their hierarchy in the simulation netlist (depressed) or simply in alpha-numeric order.

**Include V traces**

- When depressed, filters the navigator to show Voltage (V) traces.

**Include I traces**

- When depressed, filters the navigator to show Current (I) traces.

**Include Q traces**

- When depressed, filters the navigator to show Charge (Q) traces.

**Include other traces**

- When depressed, filters the navigator traces other than V, I or Q (ex. .measure parameters such as rise time or average delay.)

**xx/yy**

- Shows the number of traces shown in the traces list (xx), compared to the total number of traces from all open simulations (yy).
Traces

Lists the traces loaded from all open simulations. A pale gold background indicates that the list is incomplete due to filtering.

The display color of a trace name indicates its status, as follows:

- **Black** – not currently loaded to a chart.
- **Brown** – used in the active chart and originally from a simulation, including scalars.
- **Red** – present in database but not from a simulation and not an arithmetic trace. Such traces can come from the “**trace add**” command executed on a trace not yet found in the simulation, or from those created when you expand a trace to individual curves, or when you “**Remove**” a chartbook from the Simulation Navigator.
- **Blue** – arithmetic traces.
- **Gray** – hidden traces.
- **Green** – traces representing scalars.

Right-Click Menu in the Trace Navigator

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add to Active Chart</td>
<td>Adds the highlighted trace to the active chart.</td>
</tr>
<tr>
<td>Remove from Active Chart</td>
<td>Removes the highlighted trace from the active chart.</td>
</tr>
<tr>
<td>Remove All Traces from Active Chart</td>
<td>Removes all traces from the active chart.</td>
</tr>
<tr>
<td>Insert in Waveform Calculator</td>
<td>Inserts the highlighted trace name in the Waveform Calculator entry window.</td>
</tr>
<tr>
<td>Activate Waveform Calculator</td>
<td>Launches the Waveform Calculator if it is not currently open, or changes focus to the Waveform Calculator if it is already open.</td>
</tr>
<tr>
<td>Edit Expression in Waveform Calculator</td>
<td>Places the expression for an arithmetic trace into the Waveform Calculator edit window.</td>
</tr>
</tbody>
</table>
Adding and Removing Traces from a Chart

You use the *Trace navigator* to manage which traces are displayed in a chart. The traces from all open simulations are loaded in the navigator list. However, not all traces need to be listed at once. The trace navigator supports flat or hierarchical browsing of signals, and you can filter by signal name, by type (V, I, Q) and by analysis in multi-analysis simulations.

You can also drag and drop a trace from the navigator list to load and display it in a chart. When the scalar is the result of a sweep, the drag and drop operation into a DC chart will plot the scalar versus the sweep variable.

You can also right-click anywhere within the Trace navigator to access the command. as well as a variety of other trace commands (see “Right-Click Menu in the Trace Navigator” on page 78).

You can also use *Shift+Drag + Drop* to create a text label with the measurement value in the form “*name=[measure calc name]*” for a singly-selected trace filter. (Note that W-Edit will sometimes add curly braces, double quotes or both to trace expression to evaluate them properly.)

When you drag a trace from the Trace navigator into the chart area, W-Edit automatically detects the type and creates a new chart if only one result type exists for the trace. If multiple result types exist for the trace, W-Edit will open the new chart dialog.
Moving a Trace from One Plot to Another

You can select a curve and drag-and-drop it from one plot to another. Curves that are selected can also be cut, copied or pasted using the right-click menu from the chart window – see “Right-Click Menu for Plots, Charts and Traces” on page 68.

You can also use the keyboard shortcuts Ctrl+Up arrow or Ctrl+Down arrow to move selected curves in a stacked plot up or down; or Shift+Up arrow and Shift+Down arrow to move a plot up or down in the chart window.

Or, you can use the Plot field in the Properties navigator to enter the number of the plot in which you would like a trace to be displayed.

Expanding and Collapsing Traces

You can view traces from a data file in a single chart or in multiple plots. When you first load a data file, W-Edit displays the traces that use the same x- and y-axis system in a single chart. If traces have different axis systems, W-Edit creates multiple plots.

Putting Compatible Traces in Separate Plots—Expand

The Chart > Expand Traces command separates traces in the selected plot into multiple new plots, containing different subsets of traces. You can also use the toolbar button ( ) or right-click anywhere in the chart window to open a context-sensitive menu with the commands Expand Traces and Collapse Traces.

Trace expansion is a two step process. If you are expanding multiple traces in a single plot, the first expansion separates the traces into separate plots. The next expand command separates individual curves of that trace into separate plots. If you are expanding a single trace in a plot, the expansion separates the trace into separate curves on separate plots.

If traces are selected when the expand command is issued, only those selected traces will be expanded. If no traces are selected, all traces are expanded.

If you invoke the expand command using the right-click menu, only the plot in which you clicked is expanded. If you invoke the expand command using the toolbar button, all plots in the active chart are expanded (unless traces are selected, in which case all those traces are expanded, regardless of the plot in which they are displayed.)

Note: The maximum number of plots you can display on one screen is 16. If the expand traces command would result in there needing to be more than 16 plots, W-Edit will ask if you want to proceed with a partial expansion. A “yes” means some traces that should be expanded will instead be placed together. A “no” cancels the command.
When you use the **Chart > Expand Traces** command, W-Edit applies the following algorithm to the selected chart, for each plot from top to bottom.

If the plot contains a non-empty selection:

1. Find the highest-level non-degenerate (i.e., multi-valued) variation among all the curves in the plot. This may be a signal name, an alter, a variable, etc.
2. Use that variation to move the selected curves into new plot(s), inserted above the existing plot. These are sorted by trace, name then value using the same priority metric used to split curves into plots (e.g., signal name, alter value, variable, etc.)
3. Remove those curves from the original plot.
4. If the original plot is now empty, delete it.
5. Leave the newly expanded curves selected.

Horizontal cursors and markers associated with a curve and on the same plot as their associated curve move with their associated curve whether selected or not. Labels that are linked with a marker are considered part of the marker and are moved or copied along with the marker.

Selected horizontal cursors, markers, and labels are copied to all new plots sourced from the plot containing them and will remain selected. Objects copied from selected objects retain the state of selection.

Unselected horizontal cursors, markers, and labels are not copied to new plots and remain unselected.

**Putting Compatible Traces in the Same Plot—Collapse**

You can combine multiple traces that use the same axis system into a single chart. To combine traces, select the desired traces and use **Chart > Collapse Traces**. You can also use the toolbar button or you can right-click anywhere in the chart window to open the context-sensitive menu with the **Collapse Trace** command.

W-Edit tries to put all the selected curves on the same plot, provided they are compatible. If no curves are selected, W-Edit collapses all curves on all plots that have the same y-axis onto a single plot.

If traces are selected, W-Edit determines which plots are collapsible. A plot is collapsible if it contains a selected curve. If no curves are selected, than all plots are collapsible.

All curves on collapsible plots for a given class of units will be collapsed onto a single plot for that unit.

Any plots no longer containing curves are removed - along with any other content contained on the plot such as cursors, markers, and labels.

If a linked label is selected and its linked marker is not selected, then the marker is selected. If a linked label is unselected and its linked marker is selected, then the label is selected.

Horizontal cursors and markers associated with a curve move with the curve whether selected or not.

Selected horizontal cursors, selected markers, and selected labels on a plot containing selected curves are moved to the same destination plot as the curves and remain selected.

Selected horizontal cursors, selected markers, and selected labels on a plot not containing selected curves remain on their original plot and remain selected.
Unselected horizontal cursors and unselected labels remain on their original plot and remain unselected.

**Showing and Hiding Traces**

Not all traces that are loaded into a chart need to be displayed at once. You can use the “Right-Click Menu in the Trace Navigator” on page 78 to show or hide traces that have been loaded into a chart.

**Selecting Traces and Curves From the Chart Legend**

You can select or deselect a trace or curve by clicking on its name in the chart legend. When a trace is selected, the color bar to the left of its legend will change according to the settings in Setup > Selection Styles (See “Setting the Appearance of Selected and Unselected Objects” on page 75.) In the example below, the legend was used to select all Out:V curves.

---

**Panning and Zooming in a Chart**

You can use the mouse wheel to zoom in (up) and out (down,) or to pan vertically using Ctrl+wheel.

The Zoom and Pan toolbar provides additional tools with which to quickly navigate within a chart.

- **Fit view in window**
  - Returns the chart view to the view when the simulation first opened in W-Edit. You can also use the menu command View > Fit or the Home key.

- **Zoom In (horizontal)**
  - Zooms in by a factor of 150%.
  - You can also use the menu command View > Zoom > Zoom In Horizontal or the + (plus) key.

- **Zoom Out (horizontal)**
  - Zooms out by a factor of 66%.
  - You can also use the menu command View > Zoom > Zoom In Vertical or the - (minus) key.
Measuring Trace Data

W-Edit provides basic graphic measurement tools you can use in the chart window.

For example, you can display the distance between data points and the derivative of a curve segment using a pair of vertical cursor bars. Or, you can position a horizontal cursor bar, or place a single point markers at an exact data point on a selected trace. The “The Draw Toolbar,” below, also provides a set of common cursor measurements.

Missing Data Points

Missing or invalid data points in a trace are marked by a “gap mark” consisting of an exclamation point inside a triangle positioned at the center of the data gap. The gap mark is red if unselected, or the designated selection color if the curve is selected.

This feature is supported by the following chart setting control defaults:
test chartsetting -name EnableGapMarks -value 1
test chartsetting -name MaxGapMarks -value 100

The Draw Toolbar

The buttons on this toolbar place provide built-in functions for selecting or locating key data points, in the form of cursor bars that locate certain key y-axis data points. Select a curve first and then use the desired measurement button.

Select
Click this icon to place W-Edit in select mode.

Draw Text Label
Click on this icon to add text to a chart.

Vertical Cursor Bars
Adds two vertical cursor bars to the active chart. See “Vertical Cursor Bars” on page 87.

Draw Vertical Cursor
Adds a vertical cursor bar to the active chart. See “Vertical Cursor Bars” on page 87.

Draw Horizontal Cursor
Adds a horizontal cursor bars to the active chart. See “Horizontal Cursor Bars” on page 88.

Draw Marker
Adds a marker to the active chart. See “Markers” on page 88.

Draw Labeled Marker
Adds a marker with associated text to the active chart. See “Markers” on page 88.

Find Maximum
Finds the maximum data point in the vertical direction for the selected curve.

Find Minimum
Finds the minimum data point in the vertical direction for the selected curve.

Find previous data point
With respect to the current cursor location, finds the previous data point on the selected curve.

Find next data point
With respect to the current cursor location, finds the next data point on the selected curve.

Find previous extreme
With respect to the current cursor location, finds the previous data extreme on the selected curve. (There is a 1% tolerance so that semi-flat portions of less than 1% of total Y range are skipped.)

Find next extreme
With respect to the current cursor location, finds the next data extreme on the selected curve.

Find previous edge
With respect to the current cursor location, finds the previous edge on the selected curve.

Find next edge
With respect to the current cursor location, finds the next edge on the selected curve.
Data Point Tool Tips

If you hover the cursor slowly over a trace, W-Edit will display the trace name and x, y axis values for each data point in the form of a tool tip. For a selected trace, the data points are easy to find; for an unselected trace you will need to move slowly enough to see the cursor change to a hand pointer, which indicates a data point.

You can use the Tool Tip Font button in Setup > Chart Text to set tool tip font characteristics, and the following chart settings functions are also available (defaults shown).

```
test chartsetting -name EnableToolTips -value 1
test chartsetting -name MillisecondsStationaryToolTipIsDisplayed -value 5000
test chartsetting -name MillisecondsDelayForInitialDisplayOfToolTip -value 10
test chartsetting -name MillisecondsDelayForSubsequentDisplayOfToolTip -value 100
test chartsetting -name ToolTipSearchRadiusInPixels -value 20
```

Cursor Bars and the Cursor Table

Cursor bars in W-Edit provide a way to view data for all traces and curves in a given plot, with dynamic updating as the cursor moves.

You can display the distance between data points and the derivative of a curve segment in the chart window using a pair of vertical cursor bars. You can also position a horizontal cursor bar, vertical cursor bar or single point markers at an exact data points on a trace.

Cursor bars are also used with functions on the Draw toolbar to find values such as a minimum, the next data point or a previous extreme - see “The Draw Toolbar” on page 84.

You can view cursor data separately in the Cursor Table, or by checking “Show cursor tables in chart” in Setup > Chart Styles to see it in the chart window, or you can use both. The cursor table displays more information and provides more controls than the chart window. Cursor data displayed in the chart window shows just the x- or y-axis value, the delta and the slope, if applicable (see below).
Chapter 3: Viewing Simulations

Measuring Trace Data

The Cursor Table

The cursor table displays the coordinate data for any two vertical or horizontal cursors or markers, plus the delta, the slope and the first two derivatives. It allows you to change which cursor or marker is considered primary or secondary.

The cursor table in the chart window can be turned on or off, and the data for only a selected cursor can be displayed in the cursor table view. You can also select column headers to choose what is displayed in the cursor table view.

You can also use the header size to adjust the acuity of the value display – as you lengthen the header, W-Edit displays additional digits to the maximum specified in your setup. As you shorten the header, instead of merely truncating it, W-Edit will round the value as needed to fit the display within the column width.

In the examples above, note that for the Cap=50f trace, the cursor table shows the axis value to thirteen decimal points while the chart window shows only six decimal points.

Clicking on the Primary Cursor or Secondary Cursor button will select the respective cursor in the chart.

Cursor Table Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>TCalc Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>See “average” (page 114).</td>
<td>average(trace,x1,x2)</td>
</tr>
<tr>
<td>Delta</td>
<td>The difference between the y values.</td>
<td>yval(trace,x2)-yval(trace,x1)</td>
</tr>
<tr>
<td>Derivative</td>
<td>See “derivative” (page 123).</td>
<td>derive(trace,x)</td>
</tr>
<tr>
<td>Duty cycle</td>
<td>See “dutycycle” (page 130).</td>
<td>dutycycle(trace,x1,x2)</td>
</tr>
<tr>
<td>Falltime</td>
<td>See “falltime” (page 132).</td>
<td>falltime(trace,10,90,,mean,x1,x2)</td>
</tr>
<tr>
<td>Jitter</td>
<td>The standard deviation of jitter.</td>
<td>jitter(x,0,x1,x2)</td>
</tr>
<tr>
<td>Max</td>
<td>See “maximum” (page 141).</td>
<td>ymax(trace,x1,x2)</td>
</tr>
</tbody>
</table>

W-Edit 16 User Guide 86
Chapter 3: Viewing Simulations

Measuring Trace Data

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Description</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Falltime</td>
<td>The slowest falltime.</td>
<td>( \text{vmax(falltime(trace,10,90,,,x1,x2))} )</td>
</tr>
<tr>
<td>Max Risetime</td>
<td>The slowest risetime.</td>
<td>( \text{vmax(risetime(trace,10,90,,,x1,x2))} )</td>
</tr>
<tr>
<td>Max Slew Rate</td>
<td>The largest slew rate (fastest edge).</td>
<td>( \text{vmax(slewrate(trace,,,0,x1,x2))} )</td>
</tr>
<tr>
<td>Min</td>
<td>See “minimum” (page 142).</td>
<td>( \text{ymin(trace,x1,x2)} )</td>
</tr>
<tr>
<td>Min Falltime</td>
<td>The fastest falltime.</td>
<td>( \text{vmin(falltime(trace,10,90,,,x1,x2))} )</td>
</tr>
<tr>
<td>Min Risetime</td>
<td>The fastest risetime.</td>
<td>( \text{vmin(risetime(trace,10,90,,,x1,x2))} )</td>
</tr>
<tr>
<td>Min Slew Rate</td>
<td>The smallest slew rate (slowest edge).</td>
<td>( \text{vmin(slewrate(trace,,,0,x1,x2))} )</td>
</tr>
<tr>
<td>PeakToPeak</td>
<td>The peak-to-peak amplitude, taking the mean/average value.</td>
<td>( \text{amplitude(trace,,,mean,x1,x2)} )</td>
</tr>
<tr>
<td>Risetime</td>
<td>The average risetime. See “risetime” (page 155).</td>
<td>( \text{risetime(trace,10,90,,mean,x1,x2)} )</td>
</tr>
<tr>
<td>RMS</td>
<td>The RMS amplitude. See “rms” (page 156).</td>
<td>( \text{amplitude(trace,,,rms,x1,x2)} )</td>
</tr>
<tr>
<td>Sample Average - x</td>
<td>The x-average for data points between cursors.</td>
<td>( \text{mean(xvalues(trace,x1,x2))} )</td>
</tr>
<tr>
<td>Sample Average - y</td>
<td>The y-average for data points between cursors.</td>
<td>( \text{mean(yvalues(trace,x1,x2))} )</td>
</tr>
<tr>
<td>Samples</td>
<td>A count of data points between cursors.</td>
<td>( \text{size(yvalues(trace,x1,x2))} )</td>
</tr>
<tr>
<td>SampleStdDev - x</td>
<td>The x-standard deviation of data.</td>
<td>( \text{stddev(xvalues(trace,x1,x2))} )</td>
</tr>
<tr>
<td>SampleStdDev - y</td>
<td>The y-standard deviation of data.</td>
<td>( \text{stddev(yvalues(trace,x1,x2))} )</td>
</tr>
<tr>
<td>Slew Rate</td>
<td>The average slewrate (absolute value f dY/dX).</td>
<td>( \text{slewrate(trace,,,mean,x1,x2)} )</td>
</tr>
<tr>
<td>Slope</td>
<td>The Delta divided by dX.</td>
<td>( \frac{(yval(trace,x2)-yval(trace,x1))}{(x2-x1)} )</td>
</tr>
<tr>
<td>Y</td>
<td>The waveform value at the cursor location.</td>
<td>( \text{yval(trace,x)} )</td>
</tr>
<tr>
<td>Y1, Y2, etc.</td>
<td>The waveform value at the indicated cursor.</td>
<td>( \text{yval(trace,x1)} )</td>
</tr>
</tbody>
</table>

**Vertical Cursor Bars**

The **Vertical Cursor Bars** icon ( ) places a pair of vertical bars and the **Draw Vertical Cursor Bar** icon ( ) adds a single vertical bar for measurement in the selected chart.

You may add as many vertical cursor bars as you like, but the legend will only display values for two cursor at a time.
Vertical cursor bars indicate their exact x-coordinate value, the corresponding y-coordinates for each of the curves with which they intersect, the y-difference (vertical separation) of the bars (Delta), and the derivative calculated for the portion of the curve between the two bars (Derivative).

When you add a cursor bar it is associated with the trace that is selected. Cursor bars are synchronized in plots so that when you move a vertical bar, the vertical bars in all plots are aligned and move together.

To move a bar, select it with the left mouse button and drag it to a new position using the middle mouse button. When selected, the cursor bar is displayed as a dashed line.

The cursor bars and their related legends are displayed in a different color. Note that their identity does not change with position. If you move the second bar (VCursor2) to the left of the first bar (VCursor1), the delta values will change sign.

You can edit the x- or y-coordinate of a cursor bar by selecting it and changing the value in the Properties navigator. To remove cursor bar display simply click on the icon, which functions as a toggle.

**Horizontal Cursor Bars**

The Horizontal Cursor Bar icon ( ) places a horizontal cursor bar in the active plot. To move a horizontal bar, select and drag it with the left mouse button. A selected bar is displayed as a dashed line. If you first select a trace and then add the cursor, the horizontal cursor will be associated with that trace and be moved or expanded with it.

As with a vertical cursor bar, if you move the horizontal cursor up and down, the data points of each trace that intersect with the cursor are displayed, and if there are two horizontal cursor bars the delta between them will be displayed.

**Markers**

W-Edit provides markers so you can measure any single point in the trace window. A marker can be used in the cursor table and in measurements as if it were a cursor.

Markers snap to trace data. Markers are also associated with traces, in the same manner as cursors. The association is created when the marker is initially placed if there is a just one trace selected, or the association is created later if a marker is dragged when just one trace is selected.
Chapter 3: Viewing Simulations

Measuring Trace Data

An unselected marker appears as an X. A selected marker appears as an X within a circle.

Plain Markers

Use the Draw Marker icon ( ) to place a marker that does not have associated text and may be placed anywhere in the chart window. W-Edit displays the x- and y-coordinates of a marker in the Properties navigator.

![Properties Example](image)

Labeled Markers

Use the Draw Labeled Marker icon ( ) to place a marker that has an associated text label. If you have a trace selected before you choose the labeled marker, it will snap to the data points on that trace. Otherwise the marker can be placed anywhere.

A labeled marker is evaluated, so it will display the results of an expression or TCL command. For example, the text label `ymax [return \n][measure ymax]` will return an annotation with “ymax =” and the maximum y value of the trace.

The marker and marker text are not symmetrically associated. If you select the marker both the text and the marker will move together. However, if you select just the label, it will not move.

The label is associated to the marker by a properties field. It cannot be moved independent of the marker unless you break the association. The LinkageName field in the label properties designates the marker the text is associated with (in this case “Marker2”). The Text field contains the expression or text that is displayed in the chart.
The default labeled marker display is the marker’s x and y coordinate data via Tcl function `[dtos [measure cursor -cursor $LinkageName] -digits $labeled_marker_digits]`, as shown below. (Note that the coordinates for the label are, of course, slightly different than those of the marker itself.)

By default, the $Xe$ and $Ye$ properties are set to the marker's x-position and y-position using the Tcl functions `[measure cursor -cursor $LinkageName -x]` and `[measure cursor -cursor $LinkageName -y]` respectively.

Note that the X Y coordinates for the label are, of course, slightly different than those of the marker itself.

The properties for the associated marker show that the Name is Marker2.
**Editing a Labeled Marker**

To edit the text for a labeled marker, select it and choose **Edit Text** from the right-click menu, or make changes to the Properties fields when the marker label is selected.

![Image of Edit Text Label](Image)

- **Name**: Enter a name or evaluated property for the label. For example, if you enter `[measure amax]` the label will display the maximum point of the selected trace, and will update as you select different traces.

- **Plot**: Indicates the number of the plot where the text label is located.

- **X**: Indicates the initial X-axis placement of the label.

- **Y**: Indicates the initial Y-axis placement of the label.

- **Xe**: Use this field to enter an X-axis value where you would like to position the label.

  You can also snap the text label to a `vertical` cursor bar, if one exists, by typing `[measure cursor <CursorName>]` into the Xe field of a label. The annotation will move and update with the cursor bar.

- **Ye**: Use this field to enter an Y-axis value where you would like to position the label.

  You can also use this field to snap the text label to a `horizontal` cursor bar, if one exists, by typing `[measure cursor <CursorName>]`. The annotation will move and update with the cursor bar.

- **Text**: The expression or text that is displayed in the chart.

- **LinkageName**: The object with which the label is associated.
**Text Labels**

W-Edit provides text labels so you can add text to charts manually (as opposed to chart titles, axis labels, tick mark numbers and legends, which are internally generated.) You can move a text label by selecting it and dragging it to a new position. When you cut or copy a chart, any associated text labels move with the chart. You can also copy a text label from one chart to another.

To place a text label, click on the **Draw Text Label** icon ( ), move the cursor to the desired position and click the left mouse button to open the **Text Label Settings** dialog shown below. Once you type text in the **Name** field and use the other settings as desired, click **OK** to place the text.

![Text Label Settings Dialog](image)

**Name**

Enter the text you want to place in the chart window.

**Direction**

Specify **Normal** (horizontal) or **Up** (vertical, with the first character at the bottom.)
**Editing Text Labels**

You can edit a text label by double-clicking on it to activate the Properties window, or you can select it and choose **Edit Text** from the right-click menu.

<table>
<thead>
<tr>
<th>Name</th>
<th>TextLabel</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td></td>
</tr>
<tr>
<td>i</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>ChartText</td>
</tr>
<tr>
<td>Plot</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>-212.000000</td>
</tr>
<tr>
<td>Y</td>
<td>756.74044</td>
</tr>
<tr>
<td>Xe</td>
<td>[measure cursor cursor $linkageName -x]</td>
</tr>
<tr>
<td>Ye</td>
<td>[measure cursor cursor $linkageName -y]</td>
</tr>
<tr>
<td>Text</td>
<td>$linkageName</td>
</tr>
<tr>
<td>LinkageName</td>
<td>Marker1</td>
</tr>
</tbody>
</table>

- **Name**: Enter a name or evaluated property for the label. For example, if you enter `[measure amax]` the label will display the maximum point of the selected trace, and will update as you select different traces.

- **Plot**: Indicates the number of the plot where the text label is located.

- **X**: Indicates the initial X-axis placement of the label.

- **Y**: Indicates the initial Y-axis placement of the label.

- **Xe**: Use this field to enter an X-axis value where you would like to position the label.

  You can also snap the text label to a **vertical** cursor bar, if one exists, by typing `[measure cursor <CursorName>]` into the Xe field of a label. The annotation will move and update with the cursor bar.

- **Ye**: Use this field to enter a Y-axis value where you would like to position the label.

  You can also use this field to snap the text label to a **horizontal** cursor bar, if one exists, by typing `[measure cursor <CursorName>]`. The annotation will move and update with the cursor bar.
You can also edit text labels from the Properties pane when the text label is selected.

### Exporting Trace Data

You can export the data points for a single curve for inclusion in external applications. Select a trace in the Chart window, open the right-click menu, and use the option **Copy Data to Clipboard**.

Curve data can then be pasted from the clipboard. **Copy Data to Clipboard** writes out the data points in ASCII format, with x- and y- data separated by tabs, and each subsequent point separated by a new line. Data is formatted in scientific notation, as shown below.

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6.25e-012</td>
<td>1.71664e-007</td>
</tr>
<tr>
<td>6.875e-011</td>
<td>3.73739e-006</td>
</tr>
<tr>
<td>2.61761e-010</td>
<td>1.25543e-005</td>
</tr>
<tr>
<td>4.27974e-010</td>
<td>1.51427e-005</td>
</tr>
<tr>
<td>5.43376e-010</td>
<td>1.38623e-005</td>
</tr>
<tr>
<td>6.42009e-010</td>
<td>1.2637e-005</td>
</tr>
<tr>
<td>7.50702e-010</td>
<td>1.17156e-005</td>
</tr>
<tr>
<td>8.86371e-010</td>
<td>9.28609e-006</td>
</tr>
<tr>
<td>1e-009</td>
<td>3.98138e-006</td>
</tr>
<tr>
<td>1.1541e-009</td>
<td>8.18236e-007</td>
</tr>
<tr>
<td>1.30237e-009</td>
<td>1.17115e-006</td>
</tr>
<tr>
<td>1.44411e-009</td>
<td>7.64579e-007</td>
</tr>
<tr>
<td>1.58382e-009</td>
<td>5.56262e-007</td>
</tr>
<tr>
<td>1.73253e-009</td>
<td>4.62898e-007</td>
</tr>
<tr>
<td>1.98255e-009</td>
<td>5.74952e-007</td>
</tr>
<tr>
<td>2.13255e-009</td>
<td>1.23598e-006</td>
</tr>
<tr>
<td>2.33255e-009</td>
<td>9.39736e-006</td>
</tr>
</tbody>
</table>
4 Scripting with TCL

W-Edit uses the TCL scripting language to execute all operations. When you initiate an operation in the W-Edit graphical interface, the corresponding TCL command is written to the Command window.

In fact, it is the TCL commands in the Command window that cause W-Edit to execute an operation. Therefore, you can also write scripts directly into the Command window to execute operations.

Text can be typed in directly, copied from executed operations and then pasted back into the Command window, or invoked from a saved TCL file.

You can also drag and drop a TCL file from a browser into the Command window to execute it immediately.

You can also run scripts automatically when a chartbook is loaded and when W-Edit opens or closes by placing TCL files in certain special directories (see “Executing Scripts Automatically” on page 98).

TCL Commands Available in W-Edit

A full list of W-Edit TCL commands is available by typing “help” in the Command window. The TCL commands specific to W-Edit are listed below.

The available W-Edit commands are listed below. Help on any specific command, as well as a list of subcommands and options, can be obtained by entering the command followed by “-help”.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>annotation</td>
<td>Annotation command</td>
</tr>
<tr>
<td>calc</td>
<td>Trace calculator</td>
</tr>
<tr>
<td>chart</td>
<td>Chart command</td>
</tr>
<tr>
<td>chartbook</td>
<td>Chartbook command</td>
</tr>
<tr>
<td>delete</td>
<td>Delete command</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------------------------------------------------</td>
</tr>
<tr>
<td>database</td>
<td>Database query commands</td>
</tr>
<tr>
<td>debug</td>
<td>Debug command</td>
</tr>
<tr>
<td>delete</td>
<td>Delete command</td>
</tr>
<tr>
<td>document</td>
<td>Document command</td>
</tr>
<tr>
<td>dtos</td>
<td>Double to string –</td>
</tr>
<tr>
<td>exit</td>
<td>Exit command</td>
</tr>
<tr>
<td>find</td>
<td>Finds objects, nets, etc.</td>
</tr>
<tr>
<td>help</td>
<td>Displays a list of the TCL commands related to W-Edit.</td>
</tr>
<tr>
<td>logop</td>
<td>Executes a TCL command and logs the execution time and description to the operation log</td>
</tr>
<tr>
<td>measure</td>
<td>Measure command</td>
</tr>
<tr>
<td>mode</td>
<td>Mode command</td>
</tr>
<tr>
<td>path</td>
<td>Path command</td>
</tr>
<tr>
<td>plot</td>
<td>Plot command</td>
</tr>
<tr>
<td>point</td>
<td>Point command</td>
</tr>
<tr>
<td>print</td>
<td>Print command</td>
</tr>
<tr>
<td>probe</td>
<td>Probe device or terminal</td>
</tr>
<tr>
<td>property</td>
<td>Property command</td>
</tr>
<tr>
<td>putsmessage</td>
<td>Similar to puts but also sends output to the debug window</td>
</tr>
<tr>
<td>redo</td>
<td>Redo command</td>
</tr>
<tr>
<td>setup</td>
<td>Setup command</td>
</tr>
<tr>
<td>simulation</td>
<td>Simulation command</td>
</tr>
<tr>
<td>stod</td>
<td>String to double</td>
</tr>
<tr>
<td>test</td>
<td>Test command</td>
</tr>
<tr>
<td>texteditor</td>
<td>Texteditor command</td>
</tr>
<tr>
<td>textlabel</td>
<td>Textlabel command</td>
</tr>
<tr>
<td>trace</td>
<td>Trace command</td>
</tr>
<tr>
<td>tsource</td>
<td>Source override</td>
</tr>
<tr>
<td>undo</td>
<td>Undo command</td>
</tr>
<tr>
<td>variation</td>
<td>Variable command</td>
</tr>
<tr>
<td>window</td>
<td>Window command</td>
</tr>
<tr>
<td>workspace</td>
<td>Workspace command</td>
</tr>
</tbody>
</table>

**Running TCL Scripts**

To run a TCL script in W-Edit, you can:
Drag and drop the script into the Command window from a browser.

Invoke File > Open > Execute Script

Invoke File > Recent Scripts and select a previously run script.

Type “source” followed by the path and filename for the script, as described in “Source Scripts,” below.

(To open a TCL file in the W-Edit text editor without executing it, or to open any other kind of text file, use File > Open > Open File.)

**Source Scripts**

Source scripts take the form:

```
source filename | foldername [-subfolders] [-relativeto user|chartbook]
[-mru false|true]
```

The source command logs a warning when W-Edit cannot locate a specified file. If you are not seeing these warnings it could be that your verbosity setting is below that of the warning. To increase the verbosity setting, right-click on the log window and choose Customize; the verbosity setting is in the Filters tab.

- **filename**
  - Source runs the specified script file. The filename can contain an absolute or relative path.

- **foldername**
  - Source runs all scripts in the specified folder in alphabetical order. The foldername can contain an absolute or relative path. Does not recurse into subfolders by default.

- **-relativeto user | chartbook**
  - If a relative path is given, and -relativeto user is specified, the root for a windows installation is C:/Documents and Settings/<username>/Application Data/Tanner EDA/scripts.

  - If a relative path is given, and -relativeto chartbook is specified, the root is the chartbook folder. Note that “source -path test -relativeto chartbook” runs all the scripts in the folder “<chartbook>/scripts/test” folder. (The “scripts” folder name is appended to the <chartbook> folder name.)

  - -relativeto user is the default if no -relativeto option is specified.

- **-subfolders**
  - When a subfolder is specified, the source search path recurses breadth first into subfolders. (All scripts in the specified folder are run first, in alphabetical order, then subfolders are recurred into in alphabetical order and scripts within each subfolder are read alphabetically.)
Chapter 4: Scripting with TCL

Executing Scripts Automatically

Each chartbook in W-Edit has several folders that it reads automatically.

Though file naming is unrestricted, W-Edit reads files in alphabetical order. Thus if you have a required sequence, use file names to control the order in which scripts in any folder are executed.

Running a Script when W-Edit Launches

To run a script automatically when W-Edit opens, save the script to the following folder. Scripts saved in the “startup.wedit” directory will run on any chartbook that is opened.

`.../<username>/Application Data/Tanner EDA/scripts/startup.wedit`

For Vista operating systems use:

`C:/Users/<username>/AppData/Roaming/Tanner EDA/scripts/startup.wedit`

Running a Script when W-Edit Closes

To run a script automatically when W-Edit closes, place the script in the folder. Scripts saved in the “shutdown.wedit” directory will run on any chartbook that is opened.

`.../Documents and Settings/<username>/Application Data/Tanner EDA/scripts/shutdown.wedit`

For Vista operating systems use:

`C:/Users/<username>/AppData/Roaming/Tanner EDA/scripts/shutdown.wedit`

Running a Script when a Chartbook Opens

To run a script when any chartbook is opened, place it in the following folder.

`C:/Documents and Settings/<username>/Application Data/Tanner EDA/scripts/open.chartbook`

W-Edit also creates an open.chartbook folder for each chartbook. All scripts placed in this directory will execute automatically whenever that specific chartbook opens.

`.../<chartbookname>/scripts/open.chartbook`

Load Order for TCL Files

When you open a chartbook, W-Edit reads TCL files and folders in the order described below. At its simplest, this sequential load order allows for a universal setup intended for all users of a given
chartbook and also for individual users to modify the universal setup by saving their own setup preferences to a user preferences folder.

Scripts in any folder are executed in alphabetical order of file name. Therefore, if you have a required sequence, use file names to control the order in which your scripts are executed.

[4] Setup scripts are read from the chartbook folder [..\chartbookname\setup.wedit] first.

[5] Setup scripts are read from the user preferences folder [C:\Documents and Settings\<username>\Application Data\Tanner EDA\scripts\open.chartbook\setup]

[6] If one is defined, a source statement in [..\username\scripts\open.chartbook] runs scripts in that folder relative to user settings.

[7] If one is defined, a source statement in [..\chartbookname\scripts\open.chartbook] runs scripts in that folder relative to chartbook settings.

**Load Order for Setup Scripts**

W-Edit uses depth-first recursion to read TCL files as follows:

[1] Setup scripts in the chartbook folder [..\libraryname\setup]

[2] Setup scripts in the user preferences folder [..\username\scripts\open.chartbook\setup]

[3] All other scripts in the user preferences folder [..\username\scripts\open.chartbook]

[4] All other scripts in [..\chartbookname\scripts\open.chartbook]
5 Waveform Calculator

The Waveform Calculator is a wizard-like tool that speeds and simplifies the process of creating formulas and expressions for arithmetic traces.

TCalc, the Tanner Calculator, serves as the core calculator within T-Spice and W-Edit for evaluating numeric values and solving equations. It provides all of the functionality that you would expect from a modern scientific programmable calculator, such as a full suite of algebraic and trigonometric functions.

Moreover, support is provided for both real and complex data types, plus real and complex valued vectors and traces (circuit simulation results). There is a large collection of high level vector and trace functions that perform common circuit analysis post-processing operations, data analysis and waveform measurements, such as calculating amplitude, frequency, risetime, falltime, cross points, Fourier Transforms and others.

Like a conventional calculator, TCalc contains a keypad, a set of basic included mathematical functions and several programmable function keys. In addition, there are keys for loading traces into and out of the chart window, the Trace navigator, or from an S-Edit probe.

Expressions in the calculator can be nested, used to operate on one another, and applied as an argument to one another. You can resolve the expression in the calculator either by graphing a curve or listing data points.

Using TCalc in W-Edit

In W-Edit, TCalc is used in the background to compute all user-defined arithmetic traces, and to evaluate many of the user settings. For example, all of the cursor bar operations - find maximum/minimum, find previous/next data point, find next/previous extreme - are powered by TCalc.

The calculator is directly accessible through the TCL "calc" command. calc evaluates an expression and then either sends the text results to the log window or constructs a new trace that may be added to the active chart or used as input in another calculation.

The functionality of the TCL calc command can also be easily accessed through the graphical user interface of the Waveform Calculator window. This tool greatly simplifies the process of performing calculations by providing a keypad with buttons for arithmetic operations and function calls.

Basic Data Types

Real Numbers

Real numbers can be specified in signed scientific or engineering notation. For example,

3.1415
1.0e-10
-10nano
**Scale Factors**

Numbers may include the SI prefix scale factors, which are powers of 10. The supported scale factors and their abbreviations are:

<table>
<thead>
<tr>
<th>Scale Factor</th>
<th>Abbreviation</th>
<th>Exponent</th>
</tr>
</thead>
<tbody>
<tr>
<td>tera</td>
<td>t or T</td>
<td>10^{12}</td>
</tr>
<tr>
<td>giga</td>
<td>gig, g or G</td>
<td>10^{9}</td>
</tr>
<tr>
<td>mega</td>
<td>meg or X or M*</td>
<td>10^{6}</td>
</tr>
<tr>
<td>kilo</td>
<td>k or K</td>
<td>10^{3}</td>
</tr>
<tr>
<td>milli</td>
<td>m or M</td>
<td>10^{-3}</td>
</tr>
<tr>
<td>micro</td>
<td>mu</td>
<td>10^{-6}</td>
</tr>
<tr>
<td>nano</td>
<td>n or N</td>
<td>10^{-9}</td>
</tr>
<tr>
<td>pico</td>
<td>p or P</td>
<td>10^{-12}</td>
</tr>
<tr>
<td>femto</td>
<td>f or F</td>
<td>10^{-15}</td>
</tr>
<tr>
<td>atto</td>
<td>a or A</td>
<td>10^{-18}</td>
</tr>
</tbody>
</table>

Either the full prefix name or the abbreviation may be used.

e.g. 3g and 3giga both mean 3e9

Multiple prefixes may not be used together.

The following unit names are used as scale factors for converting to meters:

- milmi.0254 scale factor, corresponding to 10^{-3} inches
- inchesinchi.305/12 scale factor, inches to meters conversion
- feetft.305 scale factor, feet to meters conversion

* M may refer to either mega or milli depending upon the global units option setting.

**Units**

Units may also be included after numbers and scale prefixes. The list of supported units and abbreviations are:

- amperes, ampere, amps or a
- coulombs, coulomb or c
- farads, farad, fd, or f
- henrys, henry, henries or h
- hertz or hz
- meters, meter or m
- ohms, ohm or o
- seconds, second, sec or s
- volts, volt, v
A much more extensive set of unit settings is possible with the units, xunits, and yunits functions. To obtain a list of all units that are supported by these functions use the TCL command “measure units.”

**Complex Numbers**

Complex numbers may be entered using any of the notations:

\[
\begin{align*}
    a + bi \\
    a + jb \\
    (a, b)
\end{align*}
\]

where a and b are real numbers in either engineering or scientific notation.

**Vectors**

In TCalc, a vector is a sequential collection of real or complex values that may be operated upon in entirety.

Two functions, `vector(s_1,s_2,\ldots,s_n)` and `complexvector(z_1,z_2,\ldots,z_n)`, are provided for constructing vectors from individual real or complex data values.

**Traces**

A trace is an ordered collection of (x,y) pairs. In TCalc, the y value may be either real or complex. Generally, traces are the product of T-Spice .print commands.

The `tracedefine(x,y)` function provides a mechanism for constructing a trace from an existing x and y vector. `ParametricTrace(x,y)` allows you to form a trace from the x coordinates of one input trace and the y coordinates of a second trace, evaluated at the first trace x coordinates.

**Constants**

The calculator recognizes a number of mathematical and physical constants by name. When possible, these numbers are represented in the full numerical precision that is available or to 9 decimal places. Note that many of these constants have counterparts in the Verilog-A language.
You can use the calculator to display the value of a constant in the command window by either typing it in the waveform window or using the **Constants** button to select from the list shown below, and then clicking the **Measure** button.

<table>
<thead>
<tr>
<th>e</th>
<th>log2(e)</th>
</tr>
</thead>
<tbody>
<tr>
<td>log10(e)</td>
<td>pi</td>
</tr>
<tr>
<td>ln(2)</td>
<td>ln(10)</td>
</tr>
<tr>
<td>2*pi</td>
<td>pi/2</td>
</tr>
<tr>
<td>pi/4</td>
<td>1/pi</td>
</tr>
<tr>
<td>2/pi</td>
<td>2/sqrt(pi)</td>
</tr>
<tr>
<td>sqrt(2)</td>
<td>sqrt(1/2)</td>
</tr>
</tbody>
</table>

**Mathematical Constants**

<table>
<thead>
<tr>
<th>M_E</th>
<th>e, the base number for natural logarithms, 2.7182818284590452354</th>
</tr>
</thead>
<tbody>
<tr>
<td>M_LOG2E</td>
<td>log2(e), base 2 log of e</td>
</tr>
<tr>
<td>M_LOG10E</td>
<td>log10(e), base 10 log of e</td>
</tr>
<tr>
<td>M_LN2</td>
<td>ln(2), natural log of 2</td>
</tr>
<tr>
<td>M_LN10</td>
<td>ln(10), natural log of 10</td>
</tr>
<tr>
<td>M_PI</td>
<td>pi</td>
</tr>
<tr>
<td>M_TWO_PI</td>
<td>2*pi</td>
</tr>
<tr>
<td>M_PI_2</td>
<td>pi/2</td>
</tr>
<tr>
<td>M_PI_4</td>
<td>pi/4</td>
</tr>
<tr>
<td>M_1_PI</td>
<td>1/pi</td>
</tr>
<tr>
<td>M_2_PI</td>
<td>2/pi</td>
</tr>
<tr>
<td>M_2_SQRTPI</td>
<td>2/sqrt(pi)</td>
</tr>
<tr>
<td>M_SQRT2</td>
<td>sqrt(2)</td>
</tr>
<tr>
<td>M_SQRT1_2</td>
<td>1/sqrt(2)</td>
</tr>
<tr>
<td>M_DEGPERRAD</td>
<td>degrees per radian conversion factor</td>
</tr>
</tbody>
</table>

**Physical Constants**

<table>
<thead>
<tr>
<th>P_Q</th>
<th>charge of an electron in Coulombs</th>
</tr>
</thead>
<tbody>
<tr>
<td>P_C</td>
<td>speed of light in a vacuum in meters/sec</td>
</tr>
<tr>
<td>P_K</td>
<td>Boltzman's constant in joules/kelvin</td>
</tr>
<tr>
<td>P_H</td>
<td>Planck's constant in joules*sec</td>
</tr>
<tr>
<td>P_U0</td>
<td>permeability of a vacuum in henrys/meter</td>
</tr>
<tr>
<td>P_EPS0</td>
<td>permittivity of a vacuum in farads/meter</td>
</tr>
<tr>
<td>P_CELESIUS0</td>
<td>0 degrees celsius converted to kelvin, 273.15</td>
</tr>
</tbody>
</table>

**Logic Constants**

<table>
<thead>
<tr>
<th>true</th>
<th>logical true value, 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>logical not true value, 0</td>
</tr>
</tbody>
</table>
Operators

**Mathematical Operators**

- \( x + y \) addition
- \( x - y \) subtraction
- \( x * y \) multiplication
- \( x / y \) division
- \( x \% y \) modulo, the floating point remainder of \( x/y \)
- \( x^y \), \( x**y \) exponentiation, \( x^y \)
- \(-x\) unary negation

**Relational and Logical Operators**

Relational and logical operators return 1 for a true value and 0 for false.

- \( x == y \) equality
- \( x != y \), \( x<>y \) inequality
- \( x < y \) less than
- \( x<= y \) less than or equal
- \( x > y \) greater than
- \( x >= y \) greater than or equal
- \( x && y \) logical AND, test whether \( x \) and \( y \) are true (non-zero)
- \( x || y \) logical OR, test whether \( x \) or \( y \) is true (non-zero)

**Other Operators**

- \( (x) \) parentheses, grouping of operations to control order of evaluation
- \([...]\) TCL evaluation, contents of bracket will be evaluated as TCL code
- \(?:\) conditional expression: \( a?b:c \)
  - if \( a \) is true
  - then evaluate and return \( b \)
  - else evaluate and return \( c \)

**Operator Precedence**

The following table lists TCalc's operator precedence. Operators with the highest precedence are evaluated first. Operators with equal precedence are evaluated left to right.

<table>
<thead>
<tr>
<th>First (highest)</th>
<th>Second</th>
<th>Third</th>
<th>Fourth</th>
<th>Fifth</th>
<th>Sixth</th>
</tr>
</thead>
<tbody>
<tr>
<td>( (x) )</td>
<td>( f(x) )</td>
<td>(-x)</td>
<td>( x' )</td>
<td>( x*y )</td>
<td>( x + y )</td>
</tr>
<tr>
<td>parentheses</td>
<td>function call</td>
<td>unary negation</td>
<td>exponentiation</td>
<td>multiplication</td>
<td>addition</td>
</tr>
<tr>
<td></td>
<td>( [...] )</td>
<td></td>
<td></td>
<td>( x/y )</td>
<td>( x - y )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>( x/y )</td>
<td>subtraction</td>
</tr>
</tbody>
</table>
Chapter 5: Waveform Calculator Mathematical Functions

For example,

\[ 3 + 7 \times 5 = 38, \] multiplication has higher precedence than addition
\[ (3 + 7) \times 5 = 50, \] parentheses have the highest precedence and are evaluated first
\[ 4 / 2 \times 8 = 16, \] multiplication and division have equal precedence and are evaluated left to right
\[ 4 / (2 \times 8) = .25 \]
\[ 2^3 \times 2 = 16 \]
\[ 4 < 3 \times 4 = 1 \] (true)

### Mathematical Functions

#### Exponential Functions

- \( \text{db}(x) \) compute the log of a field, \( 20 \times \text{log}10(x) \); same as \( \text{db}20(x) \)
- \( \text{db10}(x) \) compute the log of a power, \( 10 \times \text{log}10(x) \)
- \( \text{db20}(x) \) compute the log of a field, \( 20 \times \text{log}10(x) \); same as \( \text{db}(x) \)
- \( \exp(x) \) exponential
- \( \text{ldexp}(m, e) \) compute a real number from the mantissa and integer exponent: \( m \times 2^e \)
- \( \ln(x) \) natural logarithm; same as \( \text{log}(x) \)
- \( \log(x) \) natural logarithm; same as \( \ln(x) \)
- \( \log10(x) \) base 10 logarithm (common log)
- \( \log2(x) \) base 2 logarithm
- \( \text{pow}(x, y) \) power calculation; compute \( x^y \)
- \( \text{power}(x, y) \) power calculation; compute \( x^y \)
- \( \text{pwr}(x, y) \) signed power calculation; if \( x < 0, -|x|^y \), otherwise \( x^y \)
- \( \text{pwrs}(x, y) \) same as \( \text{pwr} \)
- \( \text{sqrt}(x) \) compute the square root of \( x \)

#### HSPICE Compatibility Mode

- \( \text{pow}(x, n) \) integer power; for integer part of \( n \), compute \( x^n \)
- \( \text{power}(x, y) \) 'safe' power function;
  - if \( x < 0 \) return \( x^n \) for integer part of \( y \)
  - otherwise if \( x = 0 \) return 0
  - otherwise return \( x^y \)

The following are 'signed' functions in HSPICE compatible mode, i.e. \( f(x) = \text{sign}(x) \times f(|x|) \)
db(x) for negative x, returns \(-20\log(|x|)\), otherwise \(20\log(x)\).
db10(x) for negative x, returns \(-10\log(|x|)\), otherwise \(10\log(x)\).
db20(x)
ln(x)
log2(x) for negative x, returns \(-\log2(|x|)\), otherwise \(\log2(x)\)
log(x) for negative x, returns \(-\log(|x|)\), otherwise \(\log(x)\)
log10(x) for negative x, returns \(-\log10(|x|)\), otherwise \(\log10(x)\)
sqrt(x) for negative x, returns \(-\sqrt{|x|}\), otherwise \(\sqrt{x}\).

Exponential Functions that Support Complex Arguments
exp(z)
log(z)
pow(z1,z2)
pow(x,z)
pow(z,y)
sqrt(z)

Trignometric Functions
acos(x) inverse cosine; result in radians
asin(x) inverse sine; result in radians
atan(x) inverse tangent; result in radians
atan2(y,x) four quadrant inverse tangent; result in radians
cos(x) cosine of argument in radians
cosh(x) hyperbolic cosine of argument in radians
degrees(x) converts angle x from radians to degrees
radians(x) converts angle x from degrees to radians
sin(x) sine of argument in radians
sinh(x) hyperbolic sine of argument in radians
tan(x) tangent of argument in radians
tanh(x) hyperbolic tangent of argument in radians

Trignometric Functions that Support Complex Arguments
cos(z)
cosh(z)
sin(z)
sinh(z)
tan(z)
tanh(z)

Miscellaneous Functions
abs(x) absolute value of x
ceil(x) round x towards infinity
err(x1,x2) error ratio of x1 and x2; equivalent to \(\frac{\text{abs}(x_1,x_2)}{\text{max}(\text{abs}(x_1),\text{abs}(x_2))}\)
fabs(x) absolute value of x; same as abs(x)
floor(x) round x towards negative infinity
fmod(x,y) calculate the floating point remainder of x/y
hypot(x,y) calculate the hypotenuse; equivalent to \(\sqrt{x^2+y^2}\)
if(a,b,c) conditional execution, equivalent to conditional operator \(a?b:c\)
int(x) truncate x to an integer; same as trunc(x)
max(x,y) find the maximum of two numbers
min(x,y) find the minimum of two numbers
rint(x) round x to the nearest integer
random(x) generate a random number between 0 and x
sgn(x) return the sign of x, -1 or 1
sign(x,y)  sign transfer; returns x with the sign of y
trunc(x)  truncate x to an integer; same as int(x)
stp(x)  step function
    if x<0
    then return 0
    else return 1
stp(x1,x2)  continuously varying step function
    if \( x_1<-x_2 \)
    then return 0
    else if \( x_1>x_2 \) then return 1
    else return \( (x_1+x_2)/(2*x_2) \)
table(x,x1,y1,x2,y2,…,xn,yn)  
The table function evaluates the piecewise linear function defined by
the points \( x,x_1,y_1,x_2,y_2,…,x_n,y_n \), connected by straight lines.
The function value is the y-value of that function at x. The points
are automatically sorted in ascending order of x values to form
the piecewise linear function. If x is less than the smallest \( x_k \) then
the return value is the y-value corresponding to the smallest \( x_k \).
Similarly, if x is greater than the largest \( x_k \) then the return value
is the y-value corresponding to the largest \( x_k \).
mean(v)  find the average y value of a trace
median(v)  find the median y value of a trace
stddev(v)  find the standard deviation of a trace (the square root of the
variance)

Special Purpose Functions

units(u)  Set the units for the expression to the left
xunits(u)  Set the x axis units for the trace to the left
yunits(u)  Set the y axis units for the trace to the left
$\simparam(p)  Get a simulator parameter value

Differentiation and Integration Functions

ddt(f)  derivative of f with respect to time
d2dt(f)  second derivative of f with respect to time
idt(f)  integrate f in time
sdt(f)  same as idt(f)
ddx(f,x)  derivative of f with respect to x
d2dx(f,x)  second derivative of f with respect to x
idx(f,x)  integrate f in x
sdx(f,x)  same as idx(f,x)

User Defined Functions

W-Edit does not currently provide a way to specify user-defined functions.

In T-Spice, user-defined functions are expressions involving one or more arguments that are defined
once, and then reevaluated from other expressions using different arguments. They are functionally like
built-in TCalc functions that are defined by the user. See “Waveform Calculator” on page 100.

Complex Functions

The following are complex argument functions that return a real value:

abs(z)  absolute value or modulus of a complex number; same as mag(z)
arg(z)  argument (phase) of z; the angle in radians from the positive
        real axis to z
mag(z)  magnitude of a complex number; same as abs(z)
Phase angle of $z$ in degrees
Real part of a complex number
Imaginary part of a complex number

Function with a complex argument and complex return value:

Complex conjugate, for $a+bi$ returns $a-bi$

Function with a complex vector or trace argument and same return type:

Unwrap the phase of a complex vector to remove artificial phase jumps and produce smooth transitions

Vector Functions

The following are vector reduction functions which return a single real value for a given vector argument:

Find the maximum value in vector $v$
Compute the mean of vector $v$ elements
Compute the median value of vector $v$ elements
Find the minimum value in vector $v$
Return the vector size of $v$ (number of elements)
Compute the standard deviation for all elements of vector $v$
Sum all elements of vector $v$

The vector and vector complex functions form a vector from discrete data values:

Form a vector from individual data values
Form a complex vector from individual complex data values

Trace Functions

TCalc offers a large number of specialized functions for performing trace measurements and analyzing trace waveforms. For instance, you can easily compute the frequency, amplitude, period, and other wave measurements, or the risetime and falltime of an edge. Other measurements offer the ability to measure the relationship between two different waves, calculating delay, cross points, or other characteristics.

Trace Intervals

Most of the trace functions allow for the option of performing the operation over a sub-interval of the x domain, specified using the from and to parameters. These function parameters are optional, and when left unspecified the function will operate across the entire trace length, from the first point to the last. When from and to are set, then additional points are inserted at the specified x coordinates using a cubic interpolant, and the operation is performed within that interval.

As an example, suppose that trace V(out) has a transient time domain of 0 to 3 microseconds. The function $y_{max}(V(out))$ would compute the maximum y value of the entire trace, whereas $y_{max}(V(out),2u,3u)$ would compute the maximum y value for the last 1 microsecond of the simulation.
Waveforms and Edges

A number of the trace functions are targeted towards measuring waveform characteristics, such as frequency, amplitude, risetime, falltime, etc. The determination of beginning and ending x coordinates for each waveform is foundational to these functions. The approach that TCalc uses to determine these values is the concept of a threshold y value, through which each wave leading or trailing edge passes.

Optimally, this threshold value would be the exact midpoint between the wave top and bottom horizontals. For instance, a sin wave would have a threshold value of 0.0, since all waves oscillate uniformly between 1 and -1. Once the threshold value has been established, either by user input or algorithm, every occurrence of the trace crossing through the threshold value is considered to be a rising or a falling wave edge.

The algorithm that is used to determine the threshold y value is based upon a histogram of the trace points, weighted according to the segment lengths. The topline and baseline of the waveforms are then determined by finding the histogram bins that have the largest weights, and that are separated by more than 25% of the y value range.

This threshold calculation is available to the TCalc user through the threshold(trace, nBins, from, to) - make linkfunction.

Point Snapping Algorithm

Some trace functions can operate on previous function return values, passing them in as the starting position for the next evaluation. For example, nextpoint(trace,x,from,to) finds the next trace point after the input x value, where x may have been returned from a previous nextpoint function call. If there is any numeric rounding that occurs, this will cause problems. The TCL interface is heavily dependent upon conversions from numeric data to text and vice versa. In this process, rounding of data is unavoidable.

For example, the number 3.4136893849e-8 may get rounded to 3.413689e-8, and when this value is passed into the nextpoint function, it would continue to return the exact same point since it is continually being repositioned to slightly before this point with each function call.

To avoid these errors with numeric precision, many of the trace functions will utilize a point snapping algorithm. In point snapping, an input point coordinate is tested for close proximity to an existing trace point, and it is reset to be equal to such a close point coordinate.

The tolerance for point snapping is 10% of the delta x value of trace points in the interval that contains the point (consecutive point region containing x). The trace functions that use point snapping are “nextedge” on page 143, “nextextreme” on page 144, “nextpoint” on page 145, “previousextreme” on page 152 and “previouspoint” on page 153.

Missing or Invalid Data Points

When we have convergence problem in a sweep, T-Spice skips that point and continues with the next. The broken plots makes the chart very difficult to understand.

putting a “red dot” there to make the discontinuity stand out. -- The gap mark is a triangle consisting an exclamation point enclosed inside of triangle
absolutejitter

Compute the absolute RMS jitter of a periodic wave.

Syntax

Trace absolutejitter(periods,reference,begin)

- **periods**: The trace of period versus time, input must be computed by the period() function
- **reference**: The reference period value; default is the mean
- **begin**: The x coordinate the function begins with
amax

Find the x coordinate of the maximum y value of a trace.

Syntax

double amax(trace, from, to)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>trace</td>
<td>The real valued trace</td>
</tr>
<tr>
<td>from</td>
<td>The beginning x value of the interval (optional, default: first point)</td>
</tr>
<tr>
<td>to</td>
<td>The last x value of the interval (optional, default: final point)</td>
</tr>
</tbody>
</table>

Description

Returns the x coordinate of the first occurrence of the maximum y value within the interval [from,to].
amin

Finds the x coordinate of the minimum y value of a trace.

Syntax

double amin(trace, from, to)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>trace</td>
<td>The real valued trace</td>
</tr>
<tr>
<td>from</td>
<td>The beginning x value of the interval (optional, default:first point)</td>
</tr>
<tr>
<td>to</td>
<td>The last x value of the interval (optional, default:final point)</td>
</tr>
</tbody>
</table>

Description

Returns the x coordinate of the first occurrence of the minimum y value within the interval [from,to].
amplitude

Measures the peak-to-peak amplitude of waveforms.

Syntax

**Vector amplitude** (trace, threshold, edge, number, from, to)

- **trace**
  The real valued trace
- **threshold**
  Threshold y value for determining an edge (optional, default: computed)
- **edge**
  Begin measurement with a certain edge of the wave (optional, default: 0), where 1 = rising edge, -1 = falling edge and 0 = either edge.
- **number**
  Measure the nth wave amplitude (optional, default: MEAN)
- **from**
  The beginning x value of the interval (optional, default: first point)
- **to**
  The last x value of the interval (optional, default: final point)

Description

Returns a vector containing the amplitudes of the selected waves within the interval [from,to].

1=rising, -1=falling, 0=either

MEAN=mean of all, MEDIAN=median of all, 0=all, 1=first, 2=second, etc.
**average**

Computes the average y value of a trace.

**Syntax**

```plaintext
double average(trace, from, to)
```

- **trace**
  - The real valued trace
- **from**
  - The beginning x value of the interval (optional, default:first point)
- **to**
  - The last x value of the interval (optional, default:final point)

**Description**

Returns the average y value over the trace interval [from, to]. The trace values are interpolated and the average is computed from the continuous curve within the specified interval.

This is different from the statistical mean of the y data point values, which can be obtained with the mean( ) function.
**bandwidth**

Compute the bandwidth of an AC trace.

**Syntax**

```plaintext
double bandwidth(trace, bound, from, to)
```

- **trace**
  - The real valued trace
- **bound**
  - The drop of the signal from peak value
- **from**
  - The beginning x value of the interval (optional, default:first point)
- **to**
  - The last x value of the interval (optional, default:final point)
baseline

Computes the baseline measure of a trace.

Syntax

\[
\text{double baseline}(\text{trace, bins, threshold, from, to})
\]

- \text{trace} \quad \text{The real valued trace}
- \text{bins} \quad \text{The number of bins in the histogram (optional, default: 100)}
- \text{threshold} \quad \text{Threshold value estimate of edge centerlines (optional, default: computed)}
- \text{from} \quad \text{The beginning x value of the interval (optional, default: first point)}
- \text{to} \quad \text{The last x value of the interval (optional, default: final point)}

Description

Returns the baseline y value of the trace within the interval \([\text{from}, \text{to}]\).

The baseline is an estimate of the steady state value of a wave bottom, and is found by taking the center point of the value distribution curve for the bottom of the wave.

This is used to find the lower value of a periodic waveform such as a digital signal that has a lot of noise or overshoots and undershoots. The baseline and topline values are calculated by using the probability density histogram method where the waveform is sampled at equally spaced time intervals and a histogram is created from the signal's y values. The bin above the signal midpoint with the most samples will define the topline and the bin below the midpoint with the most samples will define the baseline.

In W-Edit, the histogram is created as a weighted point value calculation, where the segment length between points is the weight, so it is similar to an equal spaced points algorithm but uses all the data points of the signal.
clip

Clips a trace for x interval \([\text{from},\text{to}]\) with interpolated endpoints.

Syntax

\textbf{Trace clip}(\text{trace}, \text{from}, \text{to})

- \textit{trace}: The real valued trace
- \textit{from}: The beginning x value of the interval (default: first point)
- \textit{to}: The last x value of the interval (default: final point)

Description

Returns a new trace that is formed by clipping the input trace at x coordinates \([\text{from},\text{to}]\), and interpolates new endpoints at x coordinates from and to.
**compare**

Compares two traces and forms a new trace that shows the point-by-point comparison of the first trace values with respect to the second trace values.

**Syntax**

`Trace compare(trace1, trace2, from, to)`

- `trace1` - The first real valued trace in the comparison
- `trace2` - The second real valued trace in the comparison
- `from` - The beginning x value of the interval (optional, default: the largest of the two trace's first point coordinates)
- `to` - The last x value of the interval (optional, default: the smallest of the two trace's first point coordinates)

**Description**

Returns a new trace that for each x coordinate in either the first or second trace, shows the positioning of the first trace with respect to the second. There are three possible y values for each x coordinate:

\[
\begin{align*}
    t(x) &= -\text{scale} & \text{if } \text{trace1} < \text{trace2} \text{ at coordinate } x \\
    t(x) &= 0 & \text{if } \text{trace1} = \text{trace2} \text{ at coordinate } x \\
    t(x) &= \text{scale} & \text{if } \text{trace1} > \text{trace2} \text{ at coordinate } x
\end{align*}
\]

Wherever a coordinate point is missing in either trace, interpolation is performed.
Chapter 5: Waveform Calculator Trace Functions

**cpk**

Calculates an indicator of the process capability for a waveform relative to specified upper and lower limits. Measure pk is calculated using the following equation:

\[
\frac{\text{upper} - \text{mean}}{3 \times \text{stddev}} \text{ Or } \frac{\text{mean} - \text{lower}}{3 \times \text{stddev}}
\]

(4.1)

In this calculation, \(\text{mean}\) represents the mean value of the scatter plot, \(\text{upper}\) and \(\text{lower}\) represent the specification limits you specify, and \(\text{stddev}\) represents the standard deviations of the scatter plot. When both the upper-mean and lower-mean values are provided, the smaller result of these two calculations is displayed as the measurement.

**Syntax**

```
Trace cpk(trace, lower threshold, upper threshold, from, to, versus)
```

- **trace**
  - Trace name or expression

- **lower threshold**
  - The lower threshold level.

- **upper threshold**
  - The upper threshold level.

- **lower peak**
  - Optional override of data minimum

- **upper peak**
  - Optional override of data maximum

- **from**
  - Start of measurement range

- **to**
  - End of measurement range

- **versus**
  - New independent variable

- **thresholds_as_percentage**
  - Expresses upper and/or lower threshold levels as a percentage of the peak to peak value
**cross**

Finds all x coordinates where a trace crosses through value y.

**Syntax**

`Vector cross(trace, value, edge, number, from, to)`

- `trace` The real valued trace
- `value` y value of the cross point(s)
- `edge` Only count crossings in a certain edge direction (optional, default: 0)
  - 1 = rising edge
  - 1-1 = falling edge
  - 0 = either edge
- `number` Find the nth cross point (Optional, default: 0)
  - 0 = all, 1 = first, 2 = second, etc.
- `from` The beginning x value of the interval (optional, default: first point)
- `to` The last x value of the interval (optional, default: final point)

**Description**

Returns a list (vector) of x coordinates where a trace crosses through a y value in a given direction. Interpolation is used to compute the cross points.
cursor

Return the x position of a vertical cursor, the y position of a horizontal cursor or the (x,y) position of a marker.

Syntax

**Trace intersect** (cursor, x, y)

- **cursor**
  - Name of the cursor to query.
- **x**
  - Return the x value of a marker
- **y**
  - Return the y values of a marker.
delay

Measures the delay between edges of traces.

Syntax

double delay(Trace1, value1, edge1, number1, Trace2, value2, edge2, number2, from)

trace1 The real valued trace
value1 First trace y value that triggers the start of the delay calculation.
edge1 Only count y value crossings in a certain edge direction.
  • 1 = rising edge
  • -1 = falling edge
  • 0 = either edge
number1 Measure from the nth edge of the trace, where 1 = first, 2 = second, etc.
trace2 The second real valued trace, which may be the same as trace1.
value2 Second trace y value that is the target for the end of the delay calculation.
edge2 Only count y value crossings in a certain edge direction.
  • 1 = rising edge
  • -1 = falling edge
  • 0 = either edge
number2 Measure to the nth edge of the trace, where 1 = first, 2 = second, etc.
from The beginning x value of the interval (optional, default: the largest of the two trace's first point coordinates)

Description

Returns the x axis delay between the two selected trace edges.
**derivative**

Computes the first derivative of a trace at each point.

**Syntax**

```
Trace derivative(trace, from, to)
```

- **trace**
  - The real valued trace
- **from**
  - The beginning x value of the interval (optional, default:first point)
- **to**
  - The last x value of the interval (optional, default:final point)

**Description**

Compute a new trace by computing the first derivative of the input trace at each point of the input trace. A cubic spline is computed through each data point, and the derivative evaluated from the cubic.
derivative2

Computes the second derivative of a trace at each point.

Syntax

\texttt{Trace \textbf{derivative2}(trace, from, to)}

- \texttt{trace} \hfill The real valued trace
- \texttt{from} \hfill The beginning x value of the interval (optional, default:first point)
- \texttt{to} \hfill The last x value of the interval (optional, default:final point)

Description

Compute a new trace by computing the second derivative of the input trace at each point of the input trace. A cubic spline is computed through each data point, and the derivative evaluated from the cubic.
derivativeat

Computes the first derivative of a trace at a given x coordinate

Syntax

Trace derive(trace, x)

trace The real valued trace
x Where to compute the derivative

Description

Compute and return the first derivative of the trace with respect to the independent variable at x. The derivative is computed using a cubic spline.
derivative2at

Computes the second derivative of a trace at a given x coordinate.

Syntax

\texttt{Trace \texttt{der}i\texttt{v}e2(trace, x)}

\begin{itemize}
\item \textit{trace} \quad The real valued trace
\item \textit{x} \quad Where to compute the derivative
\end{itemize}

Description

Compute and return the second derivative of the trace with respect to the independent variable at x. The derivative is computed using a cubic spline.
### diff

Compares two traces at a time, reporting either a binary “match/differ” result, or a more detailed point-by-point comparison.

#### Syntax

**Trace compare**(*reference*, *target*, *abstol*, *reltol*, *xtol*, *from*, *to*, *interpolation*, *match reference*, *step*, *delay*, *glitch*, *clock*, *detail*, *chart*, *chartbook*, *table*, *percent*, *data*)

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>reference</strong></td>
<td>The reference trace name or expression</td>
</tr>
<tr>
<td><strong>target</strong></td>
<td>The target trace name or expression</td>
</tr>
<tr>
<td><strong>abstol</strong></td>
<td>The absolute vertical tolerance is min(abstol, value*reltol). If not provided, the defaults are v_abstol = 50uv, i_abstol = 1na, v_reltol = 0.001, i_reltol = 0.01.</td>
</tr>
<tr>
<td><strong>reltol</strong></td>
<td>The relative vertical tolerance is min(abstol, value*reltol). If not provided, the defaults are v_abstol = 50uv, i_abstol = 1na, v_reltol = 0.001, i_reltol = 0.01.</td>
</tr>
<tr>
<td><strong>xtol</strong></td>
<td>Default value is 0</td>
</tr>
<tr>
<td><strong>from</strong></td>
<td>Start of the measurement range on the x-axis.</td>
</tr>
<tr>
<td><strong>to</strong></td>
<td>End of the measurement range on the x-axis.</td>
</tr>
<tr>
<td><strong>interpolation</strong></td>
<td>The y values on the corresponding trace with the same x coordinate as computed by interpolation. Valid values are none</td>
</tr>
<tr>
<td><strong>match reference</strong></td>
<td>If provided, a match reference tests all the points in the reference trace, a target reference tests all the points in the target trace, and both tests both. The default is reference.</td>
</tr>
<tr>
<td><strong>step</strong></td>
<td>A time value such that comparisons are made at specified points in the source that are a step apart</td>
</tr>
<tr>
<td><strong>delay</strong></td>
<td>The target trace is delayed by the specified amount</td>
</tr>
<tr>
<td><strong>glitch</strong></td>
<td>Excludes for the report those differences that are shorter than the specified distance</td>
</tr>
<tr>
<td><strong>clock</strong></td>
<td>Computs the differences at the rising midpoints of this trace name or expression</td>
</tr>
<tr>
<td><strong>detail</strong></td>
<td>The level of detail reported in the table output, at the following levels: normal</td>
</tr>
<tr>
<td><strong>chart</strong></td>
<td>Name of chart</td>
</tr>
<tr>
<td><strong>chartbook</strong></td>
<td>Name of chartbook</td>
</tr>
<tr>
<td><strong>table</strong></td>
<td>Output point-by-point differences</td>
</tr>
<tr>
<td><strong>percent</strong></td>
<td>Outputs result as the fraction of the domain over which points differ</td>
</tr>
<tr>
<td><strong>data</strong></td>
<td>Outputs result as tcl {{x,0}{x,1}} data for point-by-point comparison</td>
</tr>
</tbody>
</table>
**Description**

Compares a reference trace and a target trace. The points where the two input traces are compared may be at the reference points, the target points, both reference and target points, points defined by an optional clock trace, or at points defined by a step argument.

The diff measure creates a differences trace containing the x coordinates of the comparison points and setting the y coordinates according to whether the reference and target traces differ at the corresponding x coordinate.

**Example**

```plaintext
trace define arithtrace0 diff(trace(Out:V,Cap=5f),trace(Out:V,Cap=50f))
trace define arithtrace1 {abs(trace(Out:V,Cap=5f) - 1.25) + 1.25}
trace define diff diff(trace(Out:V,Cap=5f),arithtrace1)

measure diff trace(Out:V,Cap=5f) trace(Out:V,Cap=50f)
measure diff trace(Out:V,Cap=5f) trace(Out:V,Cap=50f) -table
measure diff trace(Out:V,Cap=5f) trace(Out:V,Cap=50f) -data
measure diff trace(Out:V,Cap=5f) trace(Out:V,Cap=50f) -percent
measure diff trace(Out:V,Cap=5f) trace(Out:V,Cap=50f) -detail concise
measure diff trace(Out:V,Cap=5f) trace(Out:V,Cap=50f) -detail verbose

measure diff trace(Out:V,Cap=5f) trace(Out:V,Cap=5f) -detail verbose
measure diff trace(Out:V,Cap=5f) arithtrace1 -table

trace define arithtrace2 sweeprange(.1n,0,100n) -xunits seconds -yunits volts
trace define arithtrace3 sweep(.1n,100n)

trace define arithtrace4 1.25+1.25*sin(8*M_PI*sweeprange(.1n,0,100n)/100n)
-xunits seconds -yunits volts
```
**dpu**

*Compares two traces and forms a new trace that shows the point-by-point comparison of the first trace values with respect to the second trace values.*

**Syntax**

\[ \text{Trace compare}(\text{trace1, trace2, from, to}) \]

- **trace1**: The first real valued trace in the comparison
- **trace2**: The second real valued trace in the comparison
- **from**: The beginning x value of the interval ((optional, default: the largest of the two trace's first point coordinates))
- **to**: The last x value of the interval ((optional, default: the smallest of the two trace's first point coordinates))

**Description**

*Returns a new trace that for each x coordinate in either the first or second trace, shows the positioning of the first trace with respect to the second. There are three possible y values for each x coordinate:*

\[
\begin{align*}
t(x) &= -\text{scale} & \text{if } \text{trace1} < \text{trace2} \text{ at coordinate } x \\
t(x) &= 0 & \text{if } \text{trace1} = \text{trace2} \text{ at coordinate } x \\
t(x) &= \text{scale} & \text{if } \text{trace1} > \text{trace2} \text{ at coordinate } x 
\end{align*}
\]

*Wherever a coordinate point is missing in either trace, interpolation is performed.*
**dutycycle**

Measures the dutycycle of a trace, which is the time that is spent in an active state as a fraction of the total time under consideration.

**Syntax**

`Trace dutycycle(trace, threshold, from, to, versus, chart, chartbook)`

- `trace`: The real valued trace
- `threshold`: Threshold y value for determining an edge (optional, default: computed value)
- `from`: The beginning x value of the interval (optional, default: first point)
- `to`: The last x value of the interval (optional, default: final point)
- `versus`: When a simulation contains sweeps over one or more variables, the “versus” parameter is used to select one of these as the independent variable against which the measurement is taken. This can be used to create a new trace, for plotting a measured result against a sweep parameter.
- `chart`: The name of the chart.
- `chartbook`: The name of the chartbook.

**Description**

Compute a new trace by computing the second derivative of the input trace at each point of the input trace. A cubic spline is computed through each data point, and the derivative evaluated from the cubic.
**error**

Computes an error value for two traces.

**Syntax**

```plaintext
double error(Trace1, Trace2, errortype, minval, from, to)
```

- **trace1** The first trace
- **trace2** The second trace
- **errortype** Select the type of error calculation:
  - 1 or RelativeRMS
  - 2 or RelativeAverage
  - 3 or RelativeLogAverage
  - 4 or RMS
  - 5 or Average
  - 6 or Maximum
- **minval** The minimum value in the denominator of error calculations (optional, default: 1e-12)
- **from** The beginning x value of the interval (optional, default: the largest of the two trace's first point coordinates)
- **to** The last x value of the interval (optional, default: the smallest of the two trace's final point coordinates)

**Description**

Compute the error for a point-by-point comparison of two traces. Note that the input traces may be either real or complex valued.
falltime

Measures the falltime (Tf) of a trace's falling edge or edges.

Syntax

**Vector falltime**(trace, lower_threshold, upper_threshold, threshold, number, from, to)

- **trace**: The real valued trace
- **lower_threshold**: Lower threshold y value (optional, default: 10%)
- **upper_threshold**: Upper threshold y value (optional, default: 90%)
- **threshold**: Threshold y value for determining an edge (optional, default: computed value)
- **number**: Measure nth edge falltime, where 0=all, 1=first, 2=second, etc.
- **from**: The beginning x value of the interval (optional, default: first point)
- **to**: The last x value of the interval (optional, default: final point)

Description

Compute the falltime, in x axis units, of the selected trace edge(s). The default lower and upper threshold settings provide a 90% - 10% typical measurement.
fft

Computes the FFT of a trace.

Syntax

`TComplexTrace fft(trace, points, function, DCremove, from, to)`

- **trace**: The real valued trace
- **points**: FFT length in samples, must be a power of 2 (optional, default: computed)
- **function**: Window function selection (optional, default: rectangular)
- **DCremove**: flag (0 or 1) indicating DC component should be removed (optional, default: 0)
- **from**: The beginning x value of the interval (optional, default: first point)
- **to**: The last x value of the interval (optional, default: final point)

Description

Compute the frequency response of a transient trace.
fit

A linear regression measurement that fits polynomials, returning coefficients and uncertainties as standard errors.

Syntax

```plaintext
```

Description

The "measure fit" command performs a weighted-least-squares polynomial regression fit on the data, and reports the N+1 coefficients (along with N+1 statistical uncertainties in the coefficients) when asked to fit an order-N polynomial. By default, N=1, resulting in a linear least-squares fit to the data.

This measurement is commonly used to extract slopes and intercepts from noisy, complicated data (often obtained via statistical sampling (i.e. Monte Carlo analysis)).
frequency

Measures the frequency of a wave.

Syntax

**Vector frequency** (trace, threshold, edge, number, from, to)

- **trace**: The real valued trace
- **threshold**: Threshold y value for determining an edge (optional, default: computed value)
- **edge**: Begin measurement with a certain edge of the wave (optional, default: 0)
  - 1 = rising edge
  - -1 = falling edge
  - 0 = either edge
- **number**: Measure nth wave frequency, where 0=all, 1=first, 2=second, etc.
- **from**: The beginning x value of the interval (optional, default: first point)
- **to**: The last x value of the interval (optional, default: final point)

Description

Compute the frequency in Hertz of the selected waves of a trace.
gainmargin

Compute the gain margin of an AC trace.

Syntax

double gainmargin(trace, from, to)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>trace</td>
<td>The real valued trace</td>
</tr>
<tr>
<td>from</td>
<td>The beginning x value of the interval (optional, default: first point)</td>
</tr>
<tr>
<td>to</td>
<td>The last x value of the interval (optional, default: final point)</td>
</tr>
</tbody>
</table>
groupdelay

Compute the group delay of an AC trace.

Syntax

Vector groupdelay(trace, from, to)

- **trace**: The real valued trace
- **from**: The beginning x value of the interval (optional, default:first point)
- **to**: The last x value of the interval (optional, default:final point)
**histogram**

Forms a histogram of a trace.

**Syntax**

```
Trace histogram(trace, bins, style, from, to)
```

- **trace**: The real valued trace
- **bins**: The number of bins in the histogram (optional, default: 100)
- **style**: The style of output trace (optional, default: 1)
  - 0 = one point per histogram bin
  - 1 = histogram points form a rectangular plot
- **DCremove**: flag (0 or 1) indicating DC component should be removed (optional, default: 0)
- **from**: The beginning x value of the interval (optional, default: first point)
- **to**: The last x value of the interval (optional, default: final point)

**Description**

The histogram function forms a histogram from the input trace. A spline is computed through all of the trace points within the interval [from, to], and then the definite integral is computed. Each bin of the histogram corresponds to the length of curve that falls within the bin's y value range, normalized as a percentage of the entire curve. For example, a bin that has a value of 3.2 means that 3.2% of the trace length falls within the bin's value limits.

**Note:** Because the data generated is meant to turned into a square wave chart, some y values may appear multiple times to stop the formation of peaks. To prevent this, re-align with more than 100%. 

integral

Compute the integral of a trace.

Syntax

double integral(trace, from, to)

trace The real valued trace
from The beginning x value of the interval (optional, default: first point)
to The last x value of the interval (optional, default: final point)

Description

The integral function computes and returns the definite integral of the input trace over the interval [from, to]. The x coordinate of the first occurrence of the maximum y value within the interval [from, to].
**intersect**

Finds the coordinates of the intersection of two traces.

**Syntax**

```
Trace intersect(Trace1, Trace2, number, from, to)
```

- **trace1**: The first trace.
- **trace2**: The second trace.
- **number**: Find the nth intersection of the two traces (optional, default: 0), where 0=all, 1=first, 2=second, etc.
- **from**: The beginning x value of the interval (optional, default: the largest of the two trace's first point coordinates)
- **to**: The last x value of the interval (optional, default: the smallest of the two trace's final point coordinates)

**Description**

The intersect function will find the requested intersections of the two input traces within the interval [from, to]. The traces are interpolated with a cubic spline, which is evaluated to precisely find the intersections.

**Overload**

```
Trace intersect(Trace1, y, number, from, to) where y is the y value of a horizontal line
```
maximum

Constructs a trace which is the point-by-point maximum of two traces.

Syntax

**Trace maximum**(trace1, trace2, from, to)

**trace1**
- The first trace

**trace2**
- The second trace

**from**
- The beginning x value of the interval (optional, default:first point)

**to**
- The last x value of the interval (optional, default:final point)

Description

Compute a trace, each point of which is the maximum of trace1 and trace2 for interval [from,to]. The number of points in the final trace is equal to the number of unique x coordinates in either trace1 or trace2.
minimum

Constructs a trace which is the point-by-point minimum of two traces.

Syntax

```
Trace minimum(trace1, trace2, from, to)
```

- `trace1`: The first trace.
- `trace2`: The second trace.
- `from`: The beginning x value of the interval (optional, default:first point of trace)
- `to`: The last x value of the interval (optional, default:final point of trace)

Description

Compute a trace, each point of which is the minimum of `trace1` and `trace2` for interval [from,to]. The number of points in the final trace is equal to the number of unique x coordinates in either `trace1` or `trace2`.

`ymin(a)` and `ymax(a)` are used for computing the min/max value of a trace. For example, `measure calc ymin(trace(Out:V,temp=20))`. `min(a,b)` and `max(a,b)` are available for computing the min/max of 2 scalar arguments, or the element-by-element values of traces.
nextedge

Finds the initial point (t0 point) of the next leading or trailing waveform edge.

Syntax

```plaintext
double nextedge(trace, x, threshold, edge)
```

- **trace**: The real valued trace
- **x**: Search for an edge after this x coordinate
- **threshold**: Threshold y value for determining an edge (optional, default: computed value)
- **edge**: Find a certain type of edge (optional, default: 0)
  - 1 = rising edge
  - -1 = falling edge
  - 0 = either edge

Description

Returns the initial x coordinate of the next edge. If there are no more edges greater than the current x coordinate, the final x coordinate of the trace will be returned. (See also “previousextreme” on page 152.)
nextextreme

Finds the next point of a trace which is a local minima or maxima.

Syntax

double nextextreme(trace, x)

trace
   The real valued trace
x
   Search for an extreme point after this x coordinate

Description

Returns the x coordinate of the next point that is an inflection point. An inflection point is determined from a change in slope and a segment height that exceeds 1% of the trace y range. If there are no more extremes beyond the current x coordinate, then the final x coordinate of the trace will be returned. (See also “previousextreme” on page 152.)
nextpoint

Finds the next point of a trace, beyond a specified current point.

Syntax

double nextpoint(trace, x))

trace The real valued trace
x Search for a point after this x coordinate

Description

Returns the x coordinate of the next point. If there are no more points beyond the current point, or if the current point is beyond the final point of the trace, then the x coordinate of the final point in the trace will be returned. (See also “previouspoint” on page 153.)
overshoot

Measures the overshoot of a wave pulse. Overshoot is the maximum value exceeding the steady state top of a pulse (topline).

Syntax

**Vector overshoot**(trace, topline, threshold, number, from, to)

- **trace**: The real valued trace
- **topline**: The topline y value of the trace (optional, default: computed using the topline() function)
- **threshold**: Threshold y value for determining an edge (optional, default: computed value)
- **number**: measure nth wave overshoot, where 0=all, 1=first, 2=second, etc.
- **from**: The beginning x value of the interval (optional, default: first point)
- **to**: The last x value of the interval (optional, default: final point)

Description

Compute the overshoot value for the selected wave(s) within the interval [from,to].
**Parametrictrace**

Creates a new trace from x values of one trace and the y values of another.

**Syntax**

*Trace parametrictrace*(traceX, traceY)

*traceX*  
The trace that is referenced for x values.

*traceY*  
The trace that is referenced for y values.

**Description**

In this variation of the tracedefine function, the x values from the first input trace become the resulting trace's x coordinates, and the y values from the second input trace are interpolated to determine the resulting trace's values. Points that are outside the domain-in-common of x and y are ignored.
period

Measures the period of a wave.

Syntax

**Trace period**(trace, threshold, edge, number, from, to, versus, chart, chartbook)

- **trace**: The real valued trace
- **threshold**: Threshold y value for determining an edge (optional, default: computed value)
- **edge**: Bin measurement with a certain edge of the wave (optional, default: 0)
  - 1 = rising edge
  - -1 = falling edge
  - 0 = either edge
- **number**: Measure the nth wave period, where 0=all, 1=first, 2=second, etc.
- **from**: The beginning x value of the interval (optional, default: first point)
- **to**: The last x value of the interval (optional, default: final point)
- **versus**: When a simulation contains sweeps over one or more variables, the “versus” parameter is used to select one of these as the independent variable against which the measurement is taken. This can be used to create a new trace, for plotting a measured result against a sweep parameter.
- **chart**: The name of the chart.
- **chartbook**: The name of the chartbook.

Description

Compute the period of the selected wave(s) within the interval [from, to].
periodjitter

Compute the RMS jitter of a periodic waveform; |Tn - Tmean|

Syntax

**Trace periodjitter**(periods, reference, versus, chart, chartbook)

- **periods**
  Trace of period versus time, as computed by period.

- **reference**
  The reference period value, where the default is the mean.

- **versus**
  When a simulation contains sweeps over one or more variables, the “versus” parameter is used to select one of these as the independent variable against which the measurement is taken. This can be used to create a new trace, for plotting a measured result against a sweep parameter.

- **chart**
  The name of the chart.

- **chartbook**
  The name of the chartbook.
phasemargin

Compute the phase margin of an AC trace.

Syntax

double phasemargin(trace, from, to)

*trace*  
The real valued trace

*from*  
The beginning x value of the interval (optional, default:first point)

*to*  
The last x value of the interval (optional, default:final point)
**previousegedge**

Find the threshold cross point X coordinate of the previous edge from the given X coordinate.

**Syntax**

```plaintext
double previousegedge(x, trace, threshold, edge, versus, chart, chartbook)
```

- **x**: The location of interest.
- **trace**: The real valued trace
- **threshold**: The threshold cross value for determining edges.
- **edge**: The edge type, where 1=rising, -1=falling, 0=either.
- **versus**: When a simulation contains sweeps over one or more variables, the “versus” parameter is used to select one of these as the independent variable against which the measurement is taken. This can be used to create a new trace, for plotting a measured result against a sweep parameter.
- **chart**: The name of the chart.
- **chartbook**: The name of the chartbook.
previousextreme

Syntax

double previousextreme (trace, after)

*trace*  
The real valued trace

*x*  
Search for an extreme point before this x coordinate (optional, default: after the final point)

Description

Returns the x coordinate of the previous point that is an inflection point. An inflection point is determined from a change in slope and a segment height that exceeds 1% of the trace y range.

If there are no previous extremes before the current x coordinate, then the first x coordinate of the trace will be returned. See also “nextextreme” on page 144.
**previouspoint**

Finds the previous point of a trace, before a specified current point.

**Syntax**

```plaintext
double previouspoint(trace, x)
```

- **trace**: The real valued trace
- **x**: Search for a point before this x coordinate (optional, default: after the final point)

**Description**

Returns the x coordinate of the previous point.

If there are no more points before the current point, or if the current point is before the first point of the trace, then the x coordinate of the first point in the trace will be returned. See also “nextpoint” on page 145.
**pulsewidth**

Measures the pulse width of a wave.

**Syntax**

`Trace pulsewidth(trace, threshold, number, from, to)`

**trace**  
The real valued trace

**threshold**  
Threshold y value for determining an edge (optional, default: computed value)

**edge**  
Begin measurement with a certain edge of the wave (optional, default: 0)  
- 1 = rising edge  
- -1 = falling edge  
- 0 = either edge

**number**  
Measure the nth wave pulse width, where 0=all, 1=first, 2=second, etc.

**from**  
The beginning x value of the interval (optional, default: first point)

**to**  
The last x value of the interval (optional, default: final point)

**Description**

Compute the pulse width of the selected wave(s) within the interval [from,to]. The pulse width measurement begins when the specified edge crosses the threshold value, and ends when the threshold is again crossed in the same edge direction.
**risetime**

Measures the risetime (T_f) of a trace's rising edge or edges.

**Syntax**

```plaintext
Vector risetime(trace, lower_threshold, upper_threshold, threshold, number, from, to)
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>trace</code></td>
<td>The real valued trace</td>
</tr>
<tr>
<td><code>lower_threshold</code></td>
<td>Lower threshold y value (optional, default: 10%)</td>
</tr>
<tr>
<td><code>upper_threshold</code></td>
<td>Upper threshold y value (optional, default: 90%)</td>
</tr>
<tr>
<td><code>threshold</code></td>
<td>Threshold y value for determining an edge (optional, default: computed value)</td>
</tr>
<tr>
<td><code>number</code></td>
<td>Measure nth edge risetime, where 0=all, 1=first, 2=second, etc.</td>
</tr>
<tr>
<td><code>from</code></td>
<td>The beginning x value of the interval (optional, default: first point)</td>
</tr>
<tr>
<td><code>to</code></td>
<td>The last x value of the interval (optional, default: final point)</td>
</tr>
</tbody>
</table>

**Description**

Compute the risetime, in x axis units, of the selected trace edge(s). The default lower and upper threshold settings provide a 10% - 90% typical measurement.
rms

Find the root mean square of trace values across the range.

Syntax

double amax(trace, from, to)

trace \hspace{1cm} The real valued trace
from \hspace{1cm} The beginning x value of the interval (optional, default:first point)
to \hspace{1cm} The last x value of the interval (optional, default:final point)

Description

Compute the square root of the average of each of the values of the y range squared.
**sample**

Resamples a trace using a given sampling rate.

**Syntax**

**Trace sample**(trace, delta, from, to)

- **trace** The real valued trace
- **delta** the target delta x value for the new trace
- **from** The beginning x value of the interval (optional, default:first point)
- **to** The last x value of the interval (optional, default:final point)

**Description**

A new trace is constructed from the input trace, resampled with a fixed delta x point spacing. Point locations are computed from a cubic spline interpolant of the input trace.

**Overload**

**Trace sample**(trace, sweep, n, from, to)

- **sweep** The type of sweep function
  - LIN - linear sweep sampling
  - OCT - logarithmic sweep by octave
  - DEC - logarithmic sweep by decade
- **n** the number of samples
  - total points for LIN sampling
  - points per octave for OCT sampling
  - points per decade for DEC sampling

**Trace sample**(trace, vector coordinates, from, to)

- **coordinates** The x coordinates where sampling is requested.
settletime

Measures the settletime for a wave to reach its final steady state.

Syntax

Vector settletime(trace, value, tolerance, number, from, to)

- **trace**: The real valued trace
- **value**: The final steady state y value of interest
- **tolerance**: The full range of variation allowed in the settled wave region
- **number**: Measure nth wave settle time, where 0=all, 1=first, 2=second, etc.
- **from**: The beginning x value of the interval (optional, default:first point)
- **to**: The last x value of the interval (optional, default:final point)

Output

- **To**: The threshold cross point where the wave begins. It is the starting point of the settle time measurement.
- **Ts**: The settle time. At time T0 + Ts, the wave is settled so as to be within the error bound: Value-Tolerance <= Y <= Value+Tolerance.

Description

Compute the settle time, in x axis units, of the selected trace wave(s). The wave is 'settled' when consecutive peaks and troughs are within the bounds of value ± tolerance.
slewrate

Measures the maximum rate of change (slew rate) of a wave edge.

Syntax

**Vector settletime**(trace, value, tolerance, number, from, to)

- **trace**  The real valued trace
- **lower_threshold**  Lower threshold y value (optional, default: 10%)
- **upper_threshold**  Upper threshold y value (optional, default: 90%)
- **threshold**  Threshold y value for determining an edge (optional, default: computed value)
- **edge**  Begin measurement with a certain edge of the wave (optional, default: 0), where 1 = rising edge, -1= falling edge and 0 = either edge.
- **number**  Measure nth wave slewrate(optional, default: 0), where 0=all, 1=first, 2=second, etc.
- **from**  The beginning x value of the interval (optional, default: first point)
- **to**  The last x value of the interval (optional, default: final point)

Description

Compute the settle time, in x axis units, 0
smooth

Applies a moving-window average to the trace for smoothing.

Syntax

**Trace smooth**(trace, width, samples, function, from, to)

- **trace** The real valued trace
- **width** x range of the moving window (optional, default: 2x average delta x)
- **samples** The sampling density across the range (optional, default: 5)
- **function** The windowing function (optional, default: 1, triangular)
- **from** The beginning x value of the interval (optional, default:first point)
- **to** The last x value of the interval (optional, default:final point)

Description

Smooths the input trace by applying a moving window function.

The available window functions are:

- 0 Rectangular
- 1 triangular
- 2 Bartman
- 3 Hann
- 4 Hamming
- 5 Blackman
- 6 Harris
- 7 Gaussian
- 8 Kaiser
- 9 Welch
**stddev**

Calculates the standard deviation of a waveform. This measurement is intended for statistical (discrete) data such as histograms.

Syntax

**Trace compare**(trace, from, to, versus)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>trace</td>
<td>The trace name or expression</td>
</tr>
<tr>
<td>from</td>
<td>Start of measurement range.</td>
</tr>
<tr>
<td>to</td>
<td>End of measurement range.</td>
</tr>
<tr>
<td>versus</td>
<td>New independent variable</td>
</tr>
</tbody>
</table>

Description

Calculated using the following equation:

\[
stddev = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (x_i - \bar{x})^2}
\]
threshold

Estimates the trace edge threshold value.

Syntax

\texttt{double threshold}(\texttt{trace}, \texttt{bins}, \texttt{from}, \texttt{to})

- \texttt{trace} \quad \text{The real valued trace}
- \texttt{bins} \quad \text{The number of bins to use in the histogram (optional, default:100)}
- \texttt{from} \quad \text{The beginning x value of the interval (optional, default:first point)}
- \texttt{to} \quad \text{The last x value of the interval (optional, default:final point)}

Description

The \textit{threshold} function makes an estimate of the edge threshold value for a trace within the interval [\texttt{from},\texttt{to}].
topline

Computes the topline measure of a trace.

**Syntax**

```plaintext
double topline(trace, bins, threshold, from, to)
```

- **trace**: The real valued trace
- **bins**: The number of bins in the histogram (optional, default: 100)
- **threshold**: Threshold value estimate of edge centerlines (optional, default: computed)
- **from**: The beginning x value of the interval (optional, default: first point)
- **to**: The last x value of the interval (optional, default: final point)

**Description**

Returns the topline y value of the trace within the interval [from, to]. The topline is an estimate of the steady state value of a wave top, and is found by taking the center point of the value distribution curve for the top of the wave. This is used to find the upper value of a periodic waveform such as a digital signal that has a lot of noise or overshoots and undershoots.

Baseline and topline values are calculated by using the probability density histogram method, where the waveform is sampled at equally spaced time intervals and a histogram is created from the signal's y values.

The bin above the signal midpoint with the most samples will define the topline and the bin below the midpoint with the most samples will define the baseline. In W-Edit, the histogram is created as a weighted point value calculation, where the segment length between points is the weight, so it is similar to an equal spaced points algorithm but uses all the data points of the signal.
trace

Provides a syntactically unambiguous method of referring to a trace by name, and optionally selecting a subset of its component curves.

Syntax

**Trace trace**(name, qualifier0, qualifier1, …)

<table>
<thead>
<tr>
<th>name</th>
<th>The name of the trace.</th>
</tr>
</thead>
<tbody>
<tr>
<td>qualifier_n</td>
<td>A list of qualifiers that limit which curves are referenced</td>
</tr>
</tbody>
</table>

Description

Provides an alternate way to refer to a trace and optionally load a subset of its constituent curves. For instance, to reference just the 2nd .alter values of a trace, you could use: trace(out:v,alter=2).

Since the trace expressions use the comma as the primary delimiter and equal sign as the secondary delimiter, using either of these characters in an alter name will interfere with parsing of the subsequent trace expression. Instead of commas in the alter names in .inc files, you can the @ character. Do not use commas or equal sign characters in .alter names.
tracedefine

Generates a trace from 2 vectors, x and y.

Syntax

```
Trace tracedefine(x,y)
```

- **x**: The x value vector
- **y**: The y value vector

Description

Creates a new trace from 2 input vectors containing the x and y values.
units

Prints a list of units the calculator can use.
undershoot

Measures the undershoot of a wave pulse.

Syntax

**Vector undershoot**(trace, baseline, threshold, number, from, to)

- **trace**: The real valued trace
- **baseline**: The baseline y value of the trace (optional, default: computed using the baseline ( ) function)
- **threshold**: Threshold y value for determining an edge (optional, default: computed value)
- **number**: measure nth wave undershoot, where 0=all, 1=first, 2=second, etc.
- **from**: The beginning x value of the interval (optional, default: first point)
- **to**: The last x value of the interval (optional, default: final point)

Description

Compute the undershoot value for the selected wave(s) within the interval [from, to].
**xmax**

Finds the maximum x coordinate of a trace.

**Syntax**

`double xmax(trace, versus, chart, chartbook)`

- **trace**: The trace or expression.
- **versus**: When a simulation contains sweeps over one or more variables, the “versus” parameter is used to select one of these as the independent variable against which the measurement is taken. This can be used to create a new trace, for plotting a measured result against a sweep parameter.
- **chart**: The name of the chart.
- **chartbook**: The name of the chartbook.

**Description**

Returns the maximum x value of a trace.
xmin

Finds the minimum x coordinate of a trace.

Syntax

double xmin(trace, versus, chart, chartbook)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>trace</td>
<td>The trace or expression</td>
</tr>
<tr>
<td>versus</td>
<td>When a simulation contains sweeps over one or more variables, the “versus” parameter is used to select one of these as the independent variable against which the measurement is taken. This can be used to create a new trace, for plotting a measured result against a sweep parameter.</td>
</tr>
<tr>
<td>chart</td>
<td>The name of the chart.</td>
</tr>
<tr>
<td>chartbook</td>
<td>The name of the chartbook.</td>
</tr>
</tbody>
</table>

Description

Returns the minimum x value of a trace.
**xscale**

Creates a trace with scaled x values.

**Syntax**

Vector xscale(trace,scale,from, to, chart, chartbook)

- trace: The real valued trace
- scale: The scale that the existing x values will be multiplied by.
- from: The beginning x value of the interval (optional, default:first point)
- to: The last x value of the interval (optional, default:final point)
- chart: The name of the chart.
- chartbook: The name of the chartbook.

**Description**

Takes the x coordinates of the original trace and multiplies each by the value of the scale parameter. Constructs and returns a new trace using the new x coordinates.
**xshift**

Creates a trace with shifted x values.

**Syntax**

Vector $xscale$($trace, scale, from, to$)

- **trace**: The real valued trace
- **offset**: The offset of the new x values.
- **from**: The beginning x value of the interval (optional, default:first point)
- **to**: The last x value of the interval (optional, default:final point)

**Description**

Takes the x coordinates of the original trace and adds to each by the value of the offset parameter. Constructs and returns a new trace using the new x coordinates.
xsteps

Generates a trace that has values which are the x step sizes of the input trace.

Syntax

\textbf{Trace xsteps}(trace, from, to)

\textbf{trace} \hfill The real valued trace
\textbf{from} \hfill The beginning x value of the interval (optional, default: first point)
\textbf{to} \hfill The last x value of the interval (optional, default: final point)

Description

The xsteps function is used for computing the step sizes of an input trace.

\[ y'[n] = x[n+1] - x[n] \text{ for each } n \]

The x coordinates will be the same as the input trace coordinates, except that the last point location is removed, since it has no associated step size.
xval

Finds all x coordinates where the trace has a value of y.

Syntax

Vector xval(trace, y, from, to)

\textit{trace} \quad \text{The trace of interest}
\textit{y} \quad \text{The data value of interest}
\textit{from} \quad \text{The beginning x value of the interval (optional, default:first point)}
\textit{to} \quad \text{The last x value of the interval (optional, default:final point)}

Description

The \textit{xval} function forms a vector containing all x coordinates within the interval [from,to] where the value is y. Cubic spline interpolation is used.
xvalues

Extracts the x coordinates of the data points in a trace.

Syntax

Vector xvalues(trace, from, to)

- **trace**: The trace of interest
- **from**: The beginning x value of the interval (optional, default: first point)
- **to**: The last x value of the interval (optional, default: final point)

Description

The xvalues function forms a vector containing all x coordinates within the interval [from,to]. Depending upon the actual data points of the input trace, the resulting vector may not contain the from and to endpoint values.
yield

Calculates the ratio of the number of data points between the y-axis levels upper and lower relative to the total number of data points.

Syntax

**Trace yield**\((\text{trace}, \text{lower threshold}, \text{upper threshold}, \text{from}, \text{to}, \text{versus})\)

- **trace**: The trace name or expression
- **lower threshold**: The lower threshold limit
- **upper threshold**: The upper threshold limit
- **lower peak**: An optional override of the data minimum
- **upper peak**: An optional override of the data maximum
- **from**: Start of measurement range.
- **to**: End of measurement range.
- **versus**: New independent variable
**ymax**

Finds the maximum y value of a trace.

**Syntax**

```plaintext
double ymax(trace, from, to)
```

- **trace**
  - The real valued trace

- **from**
  - The beginning x value of the interval (optional, default: first point)

- **to**
  - The last x value of the interval (optional, default: final point)

**Description**

Returns the maximum value of a trace within the interval [from, to]. Only the actual data point values are processed, without interpolation.
**ymin**

Finds the minimum y value of a trace.

**Syntax**

`double ymin(trace, from, to)`

- `trace` The real valued trace
- `from` The beginning x value of the interval (optional, default:first point)
- `to` The last x value of the interval (optional, default:final point)

**Description**

Returns the minimum value of a trace within the interval [from,to]. Only the actual data point values are processed, without interpolation.
**ysteps**

Generates a trace that has values which are the y value deltas for each step of the input trace.

**Syntax**

```
Trace ysteps(trace, from, to)
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>trace</code></td>
<td>The real valued trace</td>
</tr>
<tr>
<td><code>from</code></td>
<td>The beginning x value of the interval (optional, default:first point)</td>
</tr>
<tr>
<td><code>to</code></td>
<td>The last x value of the interval (optional, default:final point)</td>
</tr>
</tbody>
</table>

**Description**

The `ysteps` function is used for computing the value changes of an input trace.

\[ y'[n] = y[n+1] - y[n] \quad \text{for each} \quad n \]

The x coordinates will be the same as the input trace coordinates, except that the last point location is removed, since it has no associated step size.
yval

Computes the value of a trace at a given coordinate.

Syntax

\[
\text{double yval}(\text{trace}, \text{x}, \text{from}, \text{to})
\]

- \text{trace} \quad \text{The real valued trace of interest}
- \text{x} \quad \text{The coordinate of interest}
- \text{from} \quad \text{The beginning x value of the interval (optional, default:first point)}
- \text{to} \quad \text{The last x value of the interval (optional, default:final point)}

Description

The yval function uses interpolation to compute the y value of a trace at a given x coordinate.
yvalues

Extracts data point values from a trace.

Syntax

Vector yvalues(trace, from, to)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>trace</td>
<td>The real valued trace</td>
</tr>
<tr>
<td>from</td>
<td>The beginning x value of the interval (optional, default:first point)</td>
</tr>
<tr>
<td>to</td>
<td>The last x value of the interval (optional, default:final point)</td>
</tr>
</tbody>
</table>

Description

The yvalues function forms a vector containing all y values within the interval [from,to]. Depending upon the actual data points of the input trace, the resulting vector may not contain the from and to endpoint values.
The Waveform Calculator

To open the calculator, choose View > Activate Waveform Calculator or pick Activate Waveform Calculator from the Trace navigator right-click menu.

The waveform calculator can generate output as measurement results, new waveform traces, text output files or as new input equations. You can type expressions, cursor names, marker names and trace names into the equation editor directly, use the built-in constants and functions, or construct them by dragging and dropping from the Trace navigator browser or the chart window. You can also update an existing arithmetic trace with the current expression in the calculator.

**Trace Nav**
Inserts the trace name selected in the Trace navigator into the expression field.

**Chart**
Inserts the fully qualified trace name of the trace selected in the chart window into the expression field.

**Schematic**
Inserts specific trace values from an S-Edit probe in the expression field. See “Probing from S-Edit” on page 187.
Chapter 5: Waveform Calculator

Memories

Use these operations to save and retrieve Waveform Calculator expressions.

§ **Store**—opens a dialog where you can name the current expression to store it for later use for the duration of the active session only.

§ **Recall**—opens a list from which you can retrieve into the calculator window any of the stored expressions from the current session.

§ **Delete**—opens a list in which you can highlight a stored expression in order to delete it.

§ **Save**—creates and saves a text file.

§ **Load**—opens a saved text file in the command window.

Constants

Opens a list from which you can select and enter a constant. See “Constants” on page 102.

Measures

Opens the **Add Measure** dialog, a wizard from which you can choose arithmetic functions to insert in the calculator. See “Using the Function Wizard” on page 183.

Clr/Graph

Clears the active chart of any existing traces and adds the curve defined by the expression in the calculator window to the chart window.

Graph

Calculates the expression in the Waveform Calculation window and displays the results as a curve in the chart window.

New Trace

Opens the **New Arithmetic Trace** dialog, where you can name and save an arithmetic trace and, optionally, graph it. See “New Arithmetic Trace” on page 187.

Edit Trace

When an arithmetic trace is selected in the Trace navigator, overwrites the expression defined by that trace name with the expression currently in the calculator.

**Note:** The Edit Trace function does not perform any checking—for example, if the calculator window is empty, the trace selected will acquire a blank definition.

Measure At

Opens the **Measure At** dialog. See “Measure At” on page 187.

Measure

Calculates the expression in the Waveform Calculation window and displays the results as data in the Command window.

Undo

Reverses the most recent edit operation.

Redo

Restores changes reversed with a previous **Undo** command.

Clear

Clears the entire contents of the expression field.

Backspace

Deletes one character to the left of the cursor in the expression window.

(Function buttons)

Provide shortcuts to commonly-used functions

(Keypad)

Provides an additional way to enter numerals and basic functions.
Naming Restrictions on Arithmetic Traces

Arithmetic traces have certain restrictions on their names. W-Edit will not complete an operation if these restrictions are violated.

- The following characters are not allowed in arithmetic trace names – a single parenthesis, comma, colon, square brackets, or double quotes, with these exceptions:
- A name may include parentheses only if they are in matched sets, i.e. a closed parenthesis for each open parenthesis.
- A name may contain commas, if they must be contained within parentheses – for example \(v(a,b)\).

Loading Traces into the Waveform Calculator

There are several ways to add a trace into an expression in the calculator.

1. You can right-click on a trace in the Trace navigator and choose Insert in Waveform Calculator from the context menu.
2. When a trace is selected in the Trace navigator, you can use the Trace Nav button in the calculator to insert the trace name into the expression.
3. If a trace is selected in a plot, you can press the Chart button in the calculator to insert the trace name into the expression.
4. Lastly, if you press the Schematic button in S-Edit, W-Edit inserts the trace name from the net or node you probed in S-Edit and also saves and names an arithmetic trace for the corresponding voltage, current or charge.

Using the Function Wizard

The Measures... button opens the Add Measure dialog, a wizard from which you can choose arithmetic functions to insert in the calculator.

When you pick a function from the Add Measure list, W-Edit populates the right side of the waveform calculator window with a a field for each of the arguments and flags used with the measurement. When you enter argument values in the fields, W-Edit inserts them in the expression maintaining proper syntax.

Cursor names can be used in range measurements (typically -from and -to). All -x and -y arguments can accept the names of vertical cursors or horizontal cursors, respectively, and marker names as well.

When a simulation contains sweeps over one or more variables, the -versus parameter is used to select one of these as the independent variable against which the measurement is taken. This can be used to create a new trace, for example, for plotting a measured result against a sweep parameter. For example:

```
chart new -analysis DC/Parametric -newwindow
trace define arithtrace3 [{measure ymax -trace Out:V -versus Cap} ]
trace add -name arithtrace3 -chartbook TransientAnalysis_RingVCO_TestBench
   -chart Chart3
```

For all functions, the optional flag -help displays a help screen for that particular function, or you can type help <keyword> into the command window.
Chapter 5: Waveform Calculator

The Waveform Calculator

- absolutejitter
  - Compute the absolute RMS jitter of a periodic wave.

- amax
  - Find the at maximum value of a trace.

- amin
  - Find the at minimum value of a trace.

- amplitude
  - Measure the peak-to-peak amplitude of a wave.

- average
  - Find the average value of a trace.

- bandwidth
  - Compute the bandwidth of an AC trace.

- baseline
  - Find the baseline of a trace.

- clip
  - Clip a trace to a new [from, to] range, with endpoint interpolation.
    Required arguments:
      - from <x-axis coordinate>: start of clip range
      - to <x-axis coordinate>: end of clip range
    Optional arguments:
      - trace <trace>: trace name or expression
      - chart <chartname>: name of chart
      - chartbook <chartbookname>: name of chartbook

- compare
  - Construct a trace that is a point-by-point comparison of two traces, where each y-value will be scaled in the negative if Y1 < y2, unscaled if Y1 == Y2, or positively scaled if y1 > Y2.
    Optional arguments:
      - trace1 <trace>: trace name or expression
      - trace2 <trace>: trace name or expression
      - scale <scale factor>: output values will be in the set [-scale, 0, scale]
      - from <x-axis coordinate>: start of measurement range
      - to <x-axis coordinate>: end of measurement range
      - chart <chartname>: name of chart
      - chartbook <chartbookname>: name of chartbook
### Waveform Calculator

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cross</td>
<td>Find the x coordinate of a crossing point through value y, optionally specifying that it begin with a rising, falling or either edge type and starting with the nth edge.</td>
</tr>
<tr>
<td>cursor</td>
<td>Returns the x position of a vertical cursor, the y position of a horizontal cursor, or the (x,y) position of a marker.</td>
</tr>
<tr>
<td>delay</td>
<td>Find the delay between two traces using the cross value for each trace, beginning with a rising, falling or either edge type and starting with the nth edge. You can optionally enter an x-value where the measurement will begin.</td>
</tr>
<tr>
<td>derivative</td>
<td>Find the derivative at each point of a trace.</td>
</tr>
<tr>
<td>derivative2</td>
<td>Find the second derivative at each point of a trace.</td>
</tr>
<tr>
<td>derivativeat</td>
<td>Find the first derivative of a trace at a given point.</td>
</tr>
<tr>
<td>derivative2at</td>
<td>Find the second derivative of a trace at a given point.</td>
</tr>
<tr>
<td>dutycycle</td>
<td>Measure the dutycycle of a trace.</td>
</tr>
<tr>
<td>error</td>
<td>Compute an error quantity for two traces, for one of these error terms: maximum, average or rms.</td>
</tr>
<tr>
<td>falltime</td>
<td>Find the falltime (Tf) of a wave.</td>
</tr>
<tr>
<td>fft</td>
<td>Calculates the Fast Fourier Transform.</td>
</tr>
<tr>
<td>fit</td>
<td>A linear regression measurement that returns coefficients and uncertainties as standard errors.</td>
</tr>
<tr>
<td>frequency</td>
<td>Find the frequency, in Hertz, of a wave.</td>
</tr>
<tr>
<td>gainmargin</td>
<td>Compute the gain margin of an AC trace.</td>
</tr>
<tr>
<td>groupdelay</td>
<td>Compute the group delay of an AC trace.</td>
</tr>
<tr>
<td>histogram</td>
<td>Generates a histogram of a trace, showing the wave's amplitude probability distribution. Optional arguments:</td>
</tr>
<tr>
<td>integral</td>
<td>Compute the integral of a trace.</td>
</tr>
<tr>
<td>intersect</td>
<td>Find the x coordinate of the nth intersection of two traces.</td>
</tr>
<tr>
<td>maximum</td>
<td>Construct a trace that is the point-by-point maximum of two traces.</td>
</tr>
<tr>
<td>minimum</td>
<td>Construct a trace that is the point-by-point minimum of two traces.</td>
</tr>
<tr>
<td>nextedge</td>
<td>Find the initial x coordinate of the next edge from the given x.</td>
</tr>
<tr>
<td>nextextreme</td>
<td>Find the next extreme from the given x coordinate.</td>
</tr>
<tr>
<td>nextpoint</td>
<td>Find the next data point from the given x coordinate.</td>
</tr>
<tr>
<td>overshoot</td>
<td>Find the overshoot (maximum value above the topline) of a trace.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>period</td>
<td>Find the period of a wave.</td>
</tr>
<tr>
<td>periodjitter</td>
<td>Finds the jitter time interval error measurements</td>
</tr>
<tr>
<td>phasemargin</td>
<td>Compute the phase margin of an AC trace.</td>
</tr>
<tr>
<td>previousedge</td>
<td>Finds the previous edge of a trace.</td>
</tr>
<tr>
<td>previousextreme</td>
<td>Find the extreme previous to the given x coordinate.</td>
</tr>
<tr>
<td>previouspoint</td>
<td>Find the data point of a trace previous to the given x coordinate.</td>
</tr>
<tr>
<td>pulsewidth</td>
<td>Find the pulse width of a wave.</td>
</tr>
<tr>
<td>risetime</td>
<td>Find the risetime (Tf) of a wave. Optionally you can specify for one or more or all of the rising edges, a threshold cross value, and with a lower and upper threshold level entered as a percentage.</td>
</tr>
<tr>
<td>rms</td>
<td>Compute the root mean square (RMS) of trace values across a given range.</td>
</tr>
<tr>
<td>sample</td>
<td>Resample a trace using a given sampling rate.</td>
</tr>
<tr>
<td>settletime</td>
<td>Measure the settletime for a wave to reach its final value</td>
</tr>
<tr>
<td>slewrate</td>
<td>Find the maximum rate of change (slew rate) of a selected wave edge.</td>
</tr>
<tr>
<td>smooth</td>
<td>Smooth a trace by applying one of the following moving average functions: Bartman, Hann, Hamming, Blackman, Harris, Gaussian, Kaiser or Welch.</td>
</tr>
<tr>
<td>threshold</td>
<td>Estimates the trace edge threshold value.</td>
</tr>
<tr>
<td>topline</td>
<td>Find the topline of a trace. (The topline is an estimate of the steady state value of a wave top, and is approximately the average value of the top of a wave pulse.)</td>
</tr>
<tr>
<td>units</td>
<td>Prints a list of the units the calculator can use.</td>
</tr>
<tr>
<td>undershoot</td>
<td>Find the minimum value below the baseline (undershoot) of a trace.</td>
</tr>
<tr>
<td>xmax</td>
<td>Finds the maximum x coordinate of a trace.</td>
</tr>
<tr>
<td>xmin</td>
<td>Finds the minimum x coordinate of a trace.</td>
</tr>
<tr>
<td>xscale</td>
<td>Multiply the x coordinates of a trace by a scale factor.</td>
</tr>
<tr>
<td>xshift</td>
<td>Shift the x coordinates of a trace.</td>
</tr>
<tr>
<td>xsteps</td>
<td>Generates a trace with the x step sizes of the input trace.</td>
</tr>
<tr>
<td>xval</td>
<td>Find all x coordinates where a trace passes through y.</td>
</tr>
<tr>
<td>xvalues</td>
<td>Finds the x coordinates of the data points in a trace.</td>
</tr>
<tr>
<td>ymax</td>
<td>Find the maximum x value of a trace. You can set a measurement range by entering a start and end x-axis value.</td>
</tr>
<tr>
<td>ymin</td>
<td>Find the minimum y value of a trace. You can optionally add a start and end x-axis value to set a measurement range.</td>
</tr>
<tr>
<td>ysteps</td>
<td>Generates a trace with the y step sizes of the input trace.</td>
</tr>
<tr>
<td>yval</td>
<td>Find the y value of a trace at the given x coordinate.</td>
</tr>
<tr>
<td>yvalues</td>
<td>Finds the y coordinates of the data points in a trace.</td>
</tr>
</tbody>
</table>
**Probing from S-Edit**

You can create an arithmetic trace in W-Edit by probing in S-Edit.

[1] The design corresponding to the active simulation must be open in S-Edit.

[2] Press **Schematic** to bring S-Edit into focus and place it in probe mode with the voltage probe selected. The S-Edit button in the Taskbar will flash orange.

[3] When you probe a pin or net in S-Edit, W-Edit will come into focus and the trace name corresponding to the device, net or pin you probed in S-Edit is inserted in the calculator window. If you then press Graph or Clr/Graph, W-Edit will create an arithmetic trace for the active expression and add that new trace to the chart window.

**New Arithmetic Trace**

Adds an arithmetic trace name to the Trace navigator.

![New Arithmetic Trace](image)

**Name**

Enter a name for the arithmetic trace. This name appears in the Trace navigator. The W-Edit default is “**arithtraceN**” where N increments by 1 beginning with 0.

**Plot in active chart**

When checked, adds a plot of the calculated trace to the active plot.

**Measure At**

Cursor names can be used in range measurements (typically -from and -to). Names of vertical cursors can be used whenever an x-position is required, horizontal cursor names can be used whenever a y-position is required, and marker names can be used for either.
All -x and -y arguments can take vertical cursors or horizontal cursors, respectively, and marker names as well.

### Measure At

**Enter one or more values of the independent variable, separated by spaces.**

**Enter start, end, and step or sample count values to measure over a range.**

- **Start:** 
- **End:**

- **Linear --step:**
- **Dec --samples / decade:**
- **OCT --samples per octave:**

- **Output as Tcl list**
- **Copy results to clipboard**

**OK** **Cancel**

(entry field) Use this field to enter independent variable values to be used as measurement points.

**Start** Enter a start value, with units of measurement.

**End** Enter an end value, with units of measurement.

**Linear --step** Enter a values that will be used as the step increment to select values.

**Dec --samples / decade** Enter the number of measurements per decade to sample.

**OCT --samples per octave** Enter the number of measurements per octave to sample.

**Output as Tcl list** Check this box to output measurements to the Command window in TCL format.

Please see “Exporting Chart Data and Images to the Clipboard,” below.

**Copy results to Clipboard** Check this box to copy chart data to the Windows clipboard.

When selected, this option disables the regular output to the Command window.

Please see “Exporting Chart Data and Images to the Clipboard,” below.

### Exporting Chart Data and Images to the Clipboard

Chart data can be copied to the Windows clipboard, from which it can be pasted to an external program. When executed, **Copy results to clipboard** runs the command "calc -clipboard", which will only operate when there is a singly-selected curve. The **calc -clipboard** command can also be issued directly from the command line.
The bitmap formats supported are regular and enhanced metafiles, PNG, BMP and JPG, plus SVG format when written to a file.

The interaction between the **Output as Tcl list** and **Copy results to clipboard** options is as follows:

<table>
<thead>
<tr>
<th>Output as Tcl list</th>
<th>Copy results to clipboard</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>unchecked</td>
<td>unchecked</td>
<td>Returns a text string with results formatted into columns. (calc)</td>
</tr>
<tr>
<td>checked</td>
<td>unchecked</td>
<td>Returns a tcl list with results formatted as list elements. (calc -tcl)</td>
</tr>
<tr>
<td>unchecked</td>
<td>checked</td>
<td>Returns nothing; copies text results into the clipboard. (calc -clipboard)</td>
</tr>
<tr>
<td>checked</td>
<td>checked</td>
<td>Returns a tcl list and copies text results into the clipboard. (calc -tcl -clipboard)</td>
</tr>
</tbody>
</table>

**Importing Data and Plotting it in W-Edit**

The tcl command **trace define** allows you to import data from both data files and from the clipboard.

*Trace define -file <filename>* reads the data from a file. If there are text headers above the data, they are used to name the new traces. If there are multiple columns of data, the first is taken to be the x-values. All columns must contain the same number of data points. If there is only one column, a trace is defined using the index as the x-value. If no trace name is found or specified with the *-name* option, the file name is taken as the trace name. If multiple traces are found, W-Edit uses filename.1, filename.2, etc.

*Trace define -clipboard* reads the data from the clipboard. The same naming rules as above are implemented, except of course the filename is not used to name the traces.

*Trace define -units, -xunits, -yunits* and *calc -units, -xunits, -yunits* each allow you to define units for an expression.
WaveTool extends the functionality of T-Spice Pro by providing post-processing capabilities for other SPICE simulators and for output files from earlier versions of Tanner Tools.

With WaveTool, you can convert Berkeley SPICE solution files to the T-Spice file format, which can then, in turn, be viewed using W-Edit. Any SPICE simulator files that are in the Berkeley SPICE raw file format can be read by WaveTool and translated into a format that W-Edit can read. WaveTool can also read and process HSPICE™ text files containing transient, AC and DC simulation results.

You can also use WaveTool to open .dat and .out files in order to convert older Tanner Tool files so they are compatible with newer versions of W-Edit.

In brief, the WaveTool utility provides the following capabilities:

- Reads files that are in Berkeley SPICE (raw file) format, T-Spice text format, or HSPICE text format
- Writes output files in T-Spice text format, which can subsequently be read in by W-Edit
- Can combine multiple input files into a single output file

### Launching WaveTool

To launch WaveTool, use the Start menu to navigate to Tanner EDA. From the folder for the product version you are using, choose WaveTool from the list of installed tools.
**Wavetool Toolbar Controls**

The WaveTool toolbar provides buttons for the Input Files area:

- **Insert** opens a new, editable line where you can manually enter a file name, which allows you to use wildcard expansion. When you enter a name containing the * wildcard character and press Enter, WaveTool automatically expands your entry to all filenames which meet the naming specification. The list will be populated with this full set of filenames, in alphabetic order. If no filenames match the typed text, WaveTool just deletes the new input field when you press Enter.

This can be useful, for example, if you have a series of HSPICE output files from a single simulation of many analysis types and alter blocks.

- **Open** launches a file open dialog so you can browse to and select a file to add to the Input files list.

- **Delete** removes the highlighted file from the Input files list.

The **Up** and **Down** buttons are provided because the order of the input files can be important – the output file will contain the input files in the sequence in which they are listed.
Wavetool Fields

Each input file is tagged with a file ID, which is a single letter from A to Z, used to identify input data values.

Lists the files that will be translated. Note that WaveTool will attempt to concatenate all files in this pane, in the order in which they are listed, during the conversion process.

(The concatenation is intended for H-SPICE files. For other input formats it is best to have only one input file open.)
Chapter 6: WaveTool

Launching WaveTool

Output -
File type

This pull-down lets you choose from the available output file types. Options are:

- T-Spice or W-Edit (.tsim)
- Legacy T-Spice text (.out)
- Legacy T-Spice binary (.dat)
- Tab delimited text (.txt)

Output -
File name

Enter or browse to the file name of your choice.

Always prompt before overwriting output file

When checked, prompts before overwriting an existing file.

Clear Log

Clears the log pane.

Run

Initiates file translation.

W

Launches W-Edit.

Exit

Closes WaveTool.
Note that several conventional shortcut key combinations in W-Edit (for example, Ctrl+F for “find”) launch S-Edit panes. You can clear these keyboard bindings to S-Edit with the “Reset All” button using a right-click in the toolbar area to open the “Customize” dialog, but this will reset all other customized hotkeys as well.

### File Menu

<table>
<thead>
<tr>
<th>Command</th>
<th>Shortcut and link to a description</th>
</tr>
</thead>
</table>
| File > New > | See “Creating a Chartbook” on page 49  
New Chartbook  
New File  |
| Ctrl+N See “Standard Toolbar” on page 16  |
| File > Open > | See “Opening a Chartbook” on page 49  
Open Chartbook  
Open File  
Execute Script  |
| See “Creating a Chartbook” on page 49  
See “Running TCL Scripts” on page 96  |
| File > Image > | See “Copying or Printing a Chart Image” on page 59  
Copy to clipboard  
Save to file  |
| File > Save > | Ctrl+S See “Saving a Chartbook” on page 50  
Save Chartbook \{filename\}  
Save all chartbooks  
Save Chartbook \{filename\} as...  |
| See “Standard Toolbar” on page 16  
See “Project Setup Folder Location” on page 39  |
| File > Close > | See “Closing a Chartbook” on page 49  
Close Chartbook \{filename\}  
Close All Chartbooks  
Close File  |
| File > | See “Exporting Chart Data” on page 58  |
| File > Print Preview | See “Copying or Printing a Chart Image” on page 59  |
| File > Print | Ctrl+P See “Copying or Printing a Chart Image” on page 59  |
| File > Recent Files | “Language Selection and Simulation Retention” on page 33  |
| File > Recent Scripts | See “Running TCL Scripts” on page 96  |
| File > Exit | See “Exiting W-Edit” on page 41  |
Edit Menu

<table>
<thead>
<tr>
<th>Command</th>
<th>Shortcut and link to a description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit &gt; Undo</td>
<td>Ctrl+Z, Alt+Back See “Standard Toolbar” on page 16</td>
</tr>
<tr>
<td>Edit &gt; Redo</td>
<td>Ctrl+Y See “Standard Toolbar” on page 16</td>
</tr>
<tr>
<td>Edit &gt; Cut</td>
<td>Ctrl+X, Delete See “Standard Toolbar” on page 16</td>
</tr>
<tr>
<td>Edit &gt; Copy</td>
<td>Ctrl+C, Insert See “Standard Toolbar” on page 16</td>
</tr>
<tr>
<td>Edit &gt; Paste</td>
<td>Ctrl+V, Shift+Insert See “Standard Toolbar” on page 16</td>
</tr>
<tr>
<td>Edit &gt; Clear</td>
<td>Delete See “Standard Toolbar” on page 16</td>
</tr>
<tr>
<td>Edit &gt; Select</td>
<td>See “Selecting and Deselecting Chart Objects” on page 57</td>
</tr>
<tr>
<td></td>
<td>Ctrl+A</td>
</tr>
<tr>
<td>Edit &gt; Deselect</td>
<td>See “Selecting and Deselecting Chart Objects” on page 57</td>
</tr>
<tr>
<td></td>
<td>Alt+A</td>
</tr>
</tbody>
</table>

View Menu

<table>
<thead>
<tr>
<th>Command</th>
<th>Shortcut and link to a description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View &gt; Fit</td>
<td>Home See “Panning and Zooming in a Chart” on page 82</td>
</tr>
<tr>
<td>View &gt; Zoom</td>
<td>See “Panning and Zooming in a Chart” on page 82</td>
</tr>
<tr>
<td>Mouse Zoom</td>
<td>Z See “Mouse Wheel Options” on page 34</td>
</tr>
<tr>
<td>Zoom In - Horizontal</td>
<td>+</td>
</tr>
<tr>
<td>Zoom out - Horizontal</td>
<td>-</td>
</tr>
<tr>
<td>Zoom In - Vertical</td>
<td>Shift+Num+</td>
</tr>
<tr>
<td>Zoom out - Vertical</td>
<td>Shift+Num-</td>
</tr>
<tr>
<td>View &gt; Pan</td>
<td>See “Panning and Zooming in a Chart” on page 82</td>
</tr>
<tr>
<td>Pan Left</td>
<td>←</td>
</tr>
<tr>
<td>Pan Right</td>
<td>→</td>
</tr>
<tr>
<td>Pan Up</td>
<td>↑</td>
</tr>
<tr>
<td>Pan Down</td>
<td>↓</td>
</tr>
<tr>
<td>View &gt; Simulations Navigator</td>
<td>See “Using the Simulations Navigator” on page 50</td>
</tr>
<tr>
<td>View &gt; Traces Navigator</td>
<td>See “Using the Trace Navigator” on page 76</td>
</tr>
<tr>
<td>View &gt; Activate Traces Navigator</td>
<td>See “Using the Trace Navigator” on page 76</td>
</tr>
<tr>
<td>View &gt; Chart Parameters View</td>
<td>See “Using the Variation Navigator and Chart Parameters” on page 52</td>
</tr>
<tr>
<td>View &gt; Activate Chart Parameters View</td>
<td>See “Using the Variation Navigator and Chart Parameters” on page 52</td>
</tr>
</tbody>
</table>
### Draw Menu

<table>
<thead>
<tr>
<th>Command</th>
<th>Shortcut and link to a description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>View</strong> &gt; Cursor Table</td>
<td>See “The Cursor Table” on page 86</td>
</tr>
<tr>
<td><strong>View</strong> &gt; Activate Cursor Table</td>
<td>See “The Cursor Table” on page 86</td>
</tr>
<tr>
<td><strong>View</strong> &gt; Waveform Calculator</td>
<td>See “The Waveform Calculator” on page 181</td>
</tr>
<tr>
<td><strong>View</strong> &gt; Activate Waveform Calculator</td>
<td>See “The Waveform Calculator” on page 181</td>
</tr>
<tr>
<td><strong>View</strong> &gt; Properties</td>
<td>See “Properties Navigator” on page 30</td>
</tr>
<tr>
<td><strong>View</strong> &gt; Status Bar</td>
<td>See “Status Bar” on page 23</td>
</tr>
<tr>
<td><strong>View</strong> &gt; Command Window</td>
<td>See “Command Window” on page 24</td>
</tr>
<tr>
<td><strong>View</strong> &gt; Activate Command Window</td>
<td>See “Command Window” on page 24</td>
</tr>
<tr>
<td><strong>View</strong> &gt; Hide Docked Toolbars</td>
<td>See “Instantly Hiding Everything But the Chart Window” on page 32</td>
</tr>
<tr>
<td><strong>View</strong> &gt; Toolbars &gt; Standard</td>
<td>See “Toolbars” on page 16</td>
</tr>
<tr>
<td></td>
<td>Draw</td>
</tr>
<tr>
<td></td>
<td>Zoom and Pan</td>
</tr>
<tr>
<td></td>
<td>Locator</td>
</tr>
<tr>
<td></td>
<td>Mouse Buttons</td>
</tr>
<tr>
<td></td>
<td>Tanner Info</td>
</tr>
<tr>
<td></td>
<td>Customize</td>
</tr>
<tr>
<td><strong>View</strong> &gt; Redraw</td>
<td>Space See “Chart Window” on page 24</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Shortcut and link to a description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Draw</strong> &gt; Select</td>
<td>See “The Draw Toolbar” on page 84</td>
</tr>
<tr>
<td><strong>Draw</strong> &gt; Text Label</td>
<td>See “The Draw Toolbar” on page 84</td>
</tr>
<tr>
<td><strong>Draw</strong> &gt; Vertical Cursors</td>
<td>See “The Draw Toolbar” on page 84</td>
</tr>
<tr>
<td><strong>Draw</strong> &gt; Draw Vertical Cursor</td>
<td>See “The Draw Toolbar” on page 84</td>
</tr>
<tr>
<td><strong>Draw</strong> &gt; Draw Horizontal Cursor</td>
<td>See “The Draw Toolbar” on page 84</td>
</tr>
<tr>
<td><strong>Draw</strong> &gt; Marker</td>
<td>See “The Draw Toolbar” on page 84</td>
</tr>
<tr>
<td><strong>Draw</strong> &gt; Labeled Marker</td>
<td>See “The Draw Toolbar” on page 84</td>
</tr>
<tr>
<td><strong>Draw</strong> &gt; Find Maximum</td>
<td>See “The Draw Toolbar” on page 84</td>
</tr>
<tr>
<td><strong>Draw</strong> &gt; Find Minimum</td>
<td>See “The Draw Toolbar” on page 84</td>
</tr>
<tr>
<td><strong>Draw</strong> &gt; Find Previous Data Point</td>
<td>See “The Draw Toolbar” on page 84</td>
</tr>
<tr>
<td><strong>Draw</strong> &gt; Find Next Data Point</td>
<td>See “The Draw Toolbar” on page 84</td>
</tr>
<tr>
<td><strong>Draw</strong> &gt; Find Previous Extreme</td>
<td>See “The Draw Toolbar” on page 84</td>
</tr>
<tr>
<td><strong>Draw</strong> &gt; Find Next Extreme</td>
<td>See “The Draw Toolbar” on page 84</td>
</tr>
<tr>
<td><strong>Draw</strong> &gt; Find Previous Edge</td>
<td>See “The Draw Toolbar” on page 84</td>
</tr>
<tr>
<td><strong>Draw</strong> &gt; Find Next Edge</td>
<td>See “The Draw Toolbar” on page 84</td>
</tr>
</tbody>
</table>
# Chart Menu

<table>
<thead>
<tr>
<th>Command</th>
<th>Shortcut and link to a description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chart &gt; New Chart</td>
<td>See “Creating a Chartbook” on page 49</td>
</tr>
<tr>
<td>Chart &gt; Rename Chart</td>
<td>See “Renaming a Chart” on page 58</td>
</tr>
<tr>
<td>Chart &gt; Duplicate Chart</td>
<td>See “Duplicating a Chart” on page 58</td>
</tr>
<tr>
<td>Chart &gt; Close Chart</td>
<td>See “Closing a Chart” on page 60</td>
</tr>
<tr>
<td>Chart &gt; Close All Charts</td>
<td>See “Closing a Chartbook” on page 49</td>
</tr>
<tr>
<td>Chart &gt; Chart Properties</td>
<td>See “Setting Individual Chart Display” on page 48</td>
</tr>
<tr>
<td>Chart &gt; Chart Text</td>
<td>See “Setting Text Size in a Chart” on page 55</td>
</tr>
<tr>
<td>Chart &gt; Plot</td>
<td>See “Adding a Plot to a Chart” on page 65</td>
</tr>
<tr>
<td>New Plot</td>
<td>See “Removing a Plot from a Chart” on page 66</td>
</tr>
<tr>
<td>Delete ...</td>
<td>See “Removing Empty Plots” on page 66</td>
</tr>
<tr>
<td>Delete Empty Plots</td>
<td>See “Moving a Plot” on page 65</td>
</tr>
<tr>
<td>Move...</td>
<td>Shift+Up arrow</td>
</tr>
<tr>
<td>Move Plot Up</td>
<td>Shift+Down arrow</td>
</tr>
<tr>
<td>Move Plot Down</td>
<td>See “Moving a Plot” on page 65 or “Right-Click Menu for Plots, Charts and Traces” on page 68</td>
</tr>
<tr>
<td>Chart &gt; Move Curves</td>
<td>See “Stacked Plots” on page 67</td>
</tr>
<tr>
<td>Move Curves Up</td>
<td>Shift+Up arrow</td>
</tr>
<tr>
<td>Move Curves Down</td>
<td>Shift+Down arrow</td>
</tr>
<tr>
<td>Chart &gt; Expand Traces</td>
<td>See “Expanding and Collapsing Traces” on page 80</td>
</tr>
<tr>
<td>Chart &gt; Collapse Traces</td>
<td>See “Expanding and Collapsing Traces” on page 80</td>
</tr>
</tbody>
</table>

# Setup Menu

<table>
<thead>
<tr>
<th>Command</th>
<th>Shortcut and link to a description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setup &gt; General</td>
<td>See “Language Selection and Simulation Retention” on page 33</td>
</tr>
<tr>
<td>Setup &gt; Chart Styles</td>
<td>See “Setting Default Chart Display” on page 47</td>
</tr>
<tr>
<td>Setup &gt; Chart Text</td>
<td>See “Setting Text Size in a Chart” on page 55</td>
</tr>
<tr>
<td>Setup &gt; Trace Styles</td>
<td>See “Display Settings for Traces” on page 72</td>
</tr>
<tr>
<td>Setup &gt; Selection Styles</td>
<td>See “Setting the Appearance of Selected and Unselected Objects” on page 75</td>
</tr>
<tr>
<td>Setup &gt; Mouse</td>
<td>See “Mouse Wheel Options” on page 34</td>
</tr>
<tr>
<td>Setup &gt; Text Editor and Styles</td>
<td>See “Text File Display Options” on page 35</td>
</tr>
</tbody>
</table>
Window Menu

<table>
<thead>
<tr>
<th>Command</th>
<th>Shortcut and link to a description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Window &gt; Dockable</td>
<td>See “Docking and Floating Windows in W-Edit” on page 32</td>
</tr>
<tr>
<td>Window &gt; Auto Hide</td>
<td></td>
</tr>
<tr>
<td>Window &gt; Hide</td>
<td></td>
</tr>
<tr>
<td>Window &gt; Floating</td>
<td>See “Docking and Floating Windows in W-Edit” on page 32</td>
</tr>
<tr>
<td>Window &gt; Close</td>
<td></td>
</tr>
<tr>
<td>Window &gt; Cascade</td>
<td></td>
</tr>
<tr>
<td>Window &gt; New Window</td>
<td></td>
</tr>
<tr>
<td>Window &gt; Tile Horizontally</td>
<td></td>
</tr>
<tr>
<td>Window &gt; Tile Vertically</td>
<td></td>
</tr>
<tr>
<td>Window &gt; Arrange Icons</td>
<td></td>
</tr>
<tr>
<td>Window &gt; Close All Except Active</td>
<td></td>
</tr>
<tr>
<td>Window &gt; Theme</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Windows XP</td>
</tr>
<tr>
<td></td>
<td>Office 2000</td>
</tr>
<tr>
<td></td>
<td>Office 2003</td>
</tr>
<tr>
<td></td>
<td>etc.</td>
</tr>
<tr>
<td>Window &gt; (recently opened files)</td>
<td></td>
</tr>
</tbody>
</table>

Help Menu

<table>
<thead>
<tr>
<th>Command</th>
<th>Shortcut and link to a description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Help &gt; Waveform Viewer Manual</td>
<td>See “W-Edit Documentation” on page 43</td>
</tr>
<tr>
<td>Help &gt; Release Notes</td>
<td>See “W-Edit Documentation” on page 43</td>
</tr>
<tr>
<td>Help &gt; Tutorial</td>
<td>See “Help and Diagnostics” on page 43</td>
</tr>
<tr>
<td>Help &gt; Setup Examples and Tutorial</td>
<td>See “Help and Diagnostics” on page 43</td>
</tr>
<tr>
<td>Help &gt; Commuter Licenses</td>
<td>See “W-Edit Product Support and Diagnostics” on page 44</td>
</tr>
<tr>
<td>Obtain Commuter Licenses</td>
<td></td>
</tr>
<tr>
<td>Return Commuter Licenses</td>
<td></td>
</tr>
<tr>
<td>Commuter License Status</td>
<td></td>
</tr>
<tr>
<td>Help &gt; Support</td>
<td>See “W-Edit Product Support and Diagnostics” on page 44</td>
</tr>
<tr>
<td>Support Diagnostics</td>
<td></td>
</tr>
<tr>
<td>Support on the Web</td>
<td></td>
</tr>
<tr>
<td>Contact Tanner EDA</td>
<td></td>
</tr>
<tr>
<td>Frequently Asked Questions, online</td>
<td></td>
</tr>
<tr>
<td>Help &gt; About W-Edit</td>
<td>See “W-Edit Product Support and Diagnostics” on page 44</td>
</tr>
</tbody>
</table>
Index

SYMBOLS

.cbk files
  opening, 49
.dat wdb files
  converting, 190
.out wdb files
  converting, 190

A

.alter command, 69, 164
arithmetic traces, 100
arranging windows, docking, 32
axes
  assigning coordinate systems, 56, 66, 67
  labeling, 54
  separating charts by, 80

B

Berkeley SPICE files
  converting to .tsim format, 190
buttons
  adding to toolbars, 19
  creating customized for toolbar, 19
  renaming, 19
  TCL functions for, 19

C

Caps Lock indicator, 23
.cbk files, 50
cells
  top-level, 77
chart
  copying an image of, 59, 60
  exporting data from, 85, 94
Chart >
  Cursors >
    Horizontal Bars, 88
    Markers, 88
    Vertical Bars, 87
  chart data

  copying to clipboard, 188
  chart properties
    modifying, 30, 57
  chartbooks, 9
    definition of, 45
    opening, 49
  charts
    defined, 45
    measurement bars and markers, 83
    properties, 30
    titles, 49
Command window
  as a log, 24, 95
  excluding messages from, 28
  filtering content, 28
  for executing scripts, 24
  formatting display, 27
  toggling display, 24
commands
  adding to toolbars, 19
  creating customized, 19
  renaming, 19
  TCL functions for, 19
coordinate systems, 56, 66, 67
Copy data to clipboard, 85, 94
copying a chart, 59, 60
copying chart data to clipboard, 188
cursor bars, 84
cursors, 83
  horizontal bars, 88
  markers, 88
  vertical bars, 87
curves
  definition of, 46
Customize > Keyboard, 16
Customize > Commands, 19
Customize > Options, 21
Customize > Toolbars, 18

D

data files
  traces, 69
dependent variable
  separating charts by, 54, 80
deselecting objects, 57
Index

E

design windows
  reuse behavior, 33
device parameters
  coupled transmission line
    length, 70
    lumps, 70
    lump type, 70
dialog, 14
display area, 24
documentation conventions, 12

eye charts, 57
eye diagrams, 62

F

Fade out, 76
file formats
  text files, 36

H

help
  status bar display of, 23
hidden charts
  status bar display, 23
histograms, 61
horizontal cursor bars, 88
HSPICE, 190
HSPICE compatible functions, 105

I

icon, W-Edit, 14, 190, 192
in stacked display mode, 68

K

keyboard shortcuts
  adding or changing, 20
keyword groups in text files, 37

L

labeled markers
  selecting, 57
  snapping to cursor bar, 91, 93

  labels
    axis, 54
    traces, 46
language
  changing, 33
launch, 14
launching W-Edit, 14
launching W-Edit from S-Edit, 14
launching W-Edit from T-Spice, 14
launching W-Edit using a probe command in S-Edit, 14
L-Edit
  features, 12
length device parameter
  coupled transmission line, 70
libraries
  loading, 99
  loading simulations, 49
Locator toolbar, 17
log files
  displaying, 25
  filtering, 28
  where saved, 26
lumps device parameter
  coupled transmission line, 70
lumptype device parameter
  coupled transmission line, 70

M

markers, 83, 88
mathematical functions
  HSPICE compatible, 105
Measure toolbar, 17
measurement
  horizontal cursor bars, 88
  markers, 88
  vertical cursor bars, 87
menu bar, 16
mouse
  scrolling with wheel, 34
Mouse buttons toolbar, 17
multiple document interface, 24
multiple simulation runs
  labeling, 46

N

Num Lock indicator, 23
Index

O

open.design, 99
order in which traces are displayed, 82
order of plots in a chart, 80

P

panning
  with the mouse wheel, 34
parameter sweeps, 69
  labeling, 46
plots
  definition of, 45
  order in chart, 80
  stacked display, 67
product version
  displaying, 44
project setup folder, 38
  location, 39
properties
  chart, 30, 57
  system, 30, 57
  trace, 69
  user, 30, 57

R

redrawing the chart window display, 24
refreshing the display, 24
restoring an operation, 182

S

Scroll Lock indicator, 23
scrolling
  with the mouse wheel, 34
selected state
  chart objects, 75
  labeled markers, 92
  trace display, 57
selecting objects, 57
  traces, 82
setup options
  keyword groups, 37
  language, 33
  mouse wheel, 34
  project setup folder, 38
  reusing design windows, 33
  saving local settings, 39
  text file extensions, 36
  text file updates, 34
  user preferences folder, 38
simulation commands
  .alter, 69, 164
  .step, 69, 164
simulations
  definition of, 45
  opening, 50
Standard toolbar, 16
starting W-Edit, 14
startup folder
  for button functions, 19
  running scripts automatically, 98
  running scripts on startup, 41
swept data, 69
  labels for, 46
system information
  displaying, 44

t

tcl command
  trace define, 189
TCL files
  commands used in S-Edit, 95
  editing, 24, 97
  executing, 95, 96
  filtering events to log, 20, 28
  log file, 25
  order of execution, 98
  running, 24
  running automatically, 98
  source scripts, 97
  writing, 95
text editor
  local and remote file updates, 34
text files
  extensions, 36
  external updates, 34
  files formats, 36
  keyword groups, 37
text labels, 68
title bar, 16
toolbars, 16
  adding, 18
  creating customized, 18
  hiding, 18
  locator, 17
  measurement, 17
  mouse buttons, 17
  renaming, 18
  resetting to defaults, 18
  standard, 16
  zoom and pan, 17, 82
top-level cells, 77
trace families, 46
traces
  brightness, 73
  color of in navigator list, 78
  colors, 73
  definition of, 45
  deselecting all, 57
  display settings, 73
  exporting data from, 85, 94
  filtering by wildcard, 77
  filtering display of, 79
  line type, 73
  multiple simulation runs, 46
  order of, 82
  point type, 73
  properties, 69, ??–70
  selecting, 57
  selecting all, 57
  separating charts by, 80
  separating charts into families, 54, 69, 80
  showing and hiding, 79, 82
  trace names, 69

Y
  y-axis
    separating charts by, 80

Z
  Zoom and Pan toolbar, 82
  zooming
    with the mouse wheel, 34
    Zoom and Pan toolbar, 17

U
  undoing an operation, 182
  user interface
    described, 15
    display area, 24
    menu bar, 16
    title bar, 16
    toolbars, 16
user preferences folder, 38, 99
  location, 40

V
  verbosity, 28, 97
  vertical cursor bars, 87
  View > Activate Command Window
    (shortcut backquote '), 24
  View > Status Bar, 23
  View > Toolbars, 16
  Vista operating system
    startup script locations, 98

W
  W-Edit
    program icon, 14, 190, 192
  wildcards, 77
Credits

Software Development

Seyran Avanesyan
Barry Dyne
Dan’l Leviton
Kurt Van Ness

David Cardoze
Nikita Jorniak
Massimo Sivilotti
Nicolas Williams

Quality Assurance

Luba Gromova

Documentation

Judy Bergstresser
Ken Van de Houten

Barry Dyne
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
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 agrees to 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 competitors, 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](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 or (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.”

7.2. THE WARRANTIES SET FORTH IN THIS SECTION 7 ARE EXCLUSIVE. NEITHER MENTOR GRAPHICS NOR ITS LICENSORS MAKE ANY OTHER WARRANTIES EXPRESS, IMPLIED OR STATUTORY, WITH RESPECT TO PRODUCTS PROVIDED UNDER THIS AGREEMENT. MENTOR GRAPHICS AND ITS LICENSORS SPECIFICALLY DISCLAIM ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OF INTELLECTUAL PROPERTY.

8. 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 8 SHALL SURVIVE THE TERMINATION OF THIS AGREEMENT.

9. 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 ANY DAMAGES RESULTING FROM 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. 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 10 SHALL SURVIVE THE TERMINATION OF THIS AGREEMENT.

11. INFRINGEMENT.

11.1. Mentor Graphics will defend or settle, at its option and expense, any action brought against Customer in the United States, Canada, Japan, or member state of the European Union which alleges that any standard, generally supported Product acquired by Customer hereunder infringes a patent or copyright or misappropriates a trade secret in such jurisdiction. Mentor Graphics will bear all costs and damages finally awarded against Customer that are attributable to such action. Customer understands and agrees that as conditions to Mentor Graphics’ obligations under this section Customer must: (a) notify Mentor Graphics promptly in writing of the action; (b) provide Mentor Graphics all reasonable information and assistance to settle or defend the action; and (c) grant Mentor Graphics sole authority and control of the defense or settlement of the action.

11.2. If a claim is made under Subsection 11.1 Mentor Graphics may, at its option and expense: (a) replace or modify the Product so that it becomes noninfringing; (b) procure for Customer the right to continue using the Product; or (c) require the return of the Product and refund to Customer any purchase price or license fee paid, less a reasonable allowance for use.

11.3. Mentor Graphics has no liability to Customer if the action is based upon: (a) the combination of Software or hardware with any product not furnished by Mentor Graphics; (b) the modification of the Product other than by Mentor Graphics; (c) the use of other than a current unaltered release of Software; (d) the use of the Product as part of an infringing process; (e) a product that Customer makes, uses, or sells; (f) any Beta Code or Product provided at no charge; (g) any software provided by Mentor Graphics’ licensors who do not provide such indemnification to Mentor Graphics’ customers; or (h) infringement by Customer that is deemed willful. In the case of (h), Customer shall reimburse Mentor Graphics for its reasonable attorney fees and other costs related to the action.

11.4. THIS SECTION 11 IS SUBJECT TO SECTION 8 ABOVE AND STATES THE ENTIRE LIABILITY OF MENTOR GRAPHICS AND ITS LICENSORS, AND CUSTOMER’S SOLE AND EXCLUSIVE REMEDY, FOR DEFENSE, SETTLEMENT AND DAMAGES, WITH RESPECT TO ANY ALLEGED PATENT OR COPYRIGHT INFRINGEMENT OR TRADE SECRET MISAPPROPRIATION BY ANY PRODUCT PROVIDED UNDER THIS AGREEMENT.

12. TERMINATION AND EFFECT OF TERMINATION.

12.1. If a Software license was provided for limited term use, such license will automatically terminate at the end of the authorized term. Mentor Graphics may terminate this Agreement and/or any license granted under 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