# Contents

## Preface
- What WaveScan Does .............................. 16
- Supported Data Formats .......................... 16
- Related Documents .................................. 16
- Typographic and Syntax Conventions .............. 17

## 1 Running WaveScan .................................. 19
### Starting WaveScan ................................ 20
- Starting WaveScan from ADE ...................... 20
- Starting the Standalone WaveScan ............... 20
### Exiting WaveScan ................................. 21
### Saving a Session ................................. 22
### Restoring a Session .............................. 22

## 2 Accessing Data .................................. 23
### Opening the Results Browser ..................... 24
### Choosing a Data Directory ......................... 25
### Selecting Signals ................................ 26
- Selecting Multiple Signals ....................... 26
### Filtering Signals ................................ 27
### Plotting Signals ................................ 28
- About the Signal Destination ..................... 28
- About the Graph Type .............................. 30
### Creating Graphs .................................. 32
### Data Ranging .................................... 32
### Plotting Parametric Swept Data ................... 35
### Plotting Y Versus Y for Two Signals ............. 37
### Plotting the Difference of Two Signals .......... 40
Creating Tables ................................................................. 42
Using the Calculator ......................................................... 43
Reloading Data ................................................................. 43
Menu Bar and Toolbar ......................................................... 43
  Menu Bar ................................................................. 44
  Toolbar ................................................................. 46
Graph Type ................................................................. 47
Trace Modifier .............................................................. 47
Signal Destination .......................................................... 48
Location ................................................................. 48
Status Bar ................................................................. 48

3
Working with Graphs ....................................................... 51
Opening the Graph Window ................................................ 53
Customizing a Graph ......................................................... 55
  Changing the Graph Layout ................................................ 55
  Changing the Graph Colors ............................................... 57
Handling Graph Objects .................................................... 58
  Zooming a Graph ........................................................ 59
  Panning a Graph ......................................................... 62
Editing Graph Attributes .................................................. 63
  Working With Traces ...................................................... 65
  Editing Axis Attributes ................................................... 75
  Editing Scale Attributes for Circular Graphs ......................... 80
  Working with Cursors .................................................... 82
  Working with Labels ..................................................... 89
  Working with Markers .................................................... 93
  Working With Buses ..................................................... 102
Saving and Loading Traces ................................................ 103
Freezing Graphs .............................................................. 105
Reloading Graphs ........................................................... 105
Printing Graphs .............................................................. 105
Saving Graphs ............................................................... 109
  Saving a Graph in XML Format ......................................... 110
5

Working with Tables

Opening the Report Table Window

Viewing Types of Tables

Scalar Data
Swept Data

Formatting Tables

Formatting Column Cells
Formatting a Column Header

Transposing Rows and Columns

Moving Columns

Resizing Columns

Selecting Columns

Hiding and Revealing Columns

To hide a column.
To reveal hidden columns.

Sorting Columns

Printing Tables

Saving Tables

Saving Tables in XML Format
Saving Tables in Text Format

Opening Saved Tables

Menu Bar and Toolbar
### A

**WaveScan Variables**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Application Variables</strong></td>
<td>192</td>
</tr>
<tr>
<td>statefile</td>
<td>192</td>
</tr>
<tr>
<td>readstate</td>
<td>193</td>
</tr>
<tr>
<td>writestate</td>
<td>194</td>
</tr>
<tr>
<td>graphtemplatefile</td>
<td>195</td>
</tr>
<tr>
<td>exitdialog</td>
<td>196</td>
</tr>
<tr>
<td>envdirectory</td>
<td>197</td>
</tr>
<tr>
<td>readDefaultVarFileOnStartup</td>
<td>198</td>
</tr>
<tr>
<td>writeDefaultVarFileOnExit</td>
<td>199</td>
</tr>
<tr>
<td>defaultVarFileName</td>
<td>200</td>
</tr>
<tr>
<td>printer</td>
<td>201</td>
</tr>
<tr>
<td><strong>Results Browser Variables</strong></td>
<td>202</td>
</tr>
<tr>
<td>historyLength</td>
<td>202</td>
</tr>
<tr>
<td>plotstyle</td>
<td>203</td>
</tr>
<tr>
<td>dataDirHome</td>
<td>204</td>
</tr>
<tr>
<td><strong>Filter Variable</strong></td>
<td>205</td>
</tr>
<tr>
<td>textFilterType</td>
<td>205</td>
</tr>
<tr>
<td><strong>Graph Frame Variables</strong></td>
<td>206</td>
</tr>
<tr>
<td>width</td>
<td>206</td>
</tr>
<tr>
<td>height</td>
<td>207</td>
</tr>
<tr>
<td>autoTraceSelect</td>
<td>208</td>
</tr>
<tr>
<td>rightMouseZoom</td>
<td>209</td>
</tr>
<tr>
<td><strong>Rectangular Graph Variables</strong></td>
<td>210</td>
</tr>
<tr>
<td>background</td>
<td>210</td>
</tr>
<tr>
<td>foreground</td>
<td>212</td>
</tr>
<tr>
<td>visibleDigitalRows</td>
<td>213</td>
</tr>
<tr>
<td>visibleStripChartRows</td>
<td>214</td>
</tr>
</tbody>
</table>
Circular Graph Variables .................................................. 215
  background ................................................................. 215
  foreground ................................................................. 217
Graph Variables ............................................................. 218
  visibleLegendRows ...................................................... 218
  fileSuffix ................................................................. 219
  subTitle ................................................................. 220
  displayDate ............................................................. 221
  defaultSubtitle ......................................................... 222
  cursorOn ................................................................. 223
  majGridsOn .............................................................. 224
  minGridsOn .............................................................. 225
  majorForeground ....................................................... 226
  minorForeground ....................................................... 227
  snapOn ................................................................. 228
  fontName ................................................................. 229
  fontStyle ................................................................. 230
  fontSize ................................................................. 231
  stripChartOn ........................................................... 232
Axis Variables .............................................................. 233
  depAxis logScale ....................................................... 233
  depAxis forceOrigin .................................................... 234
  indepAxis logScale ..................................................... 235
  indepAxis forceOrigin ................................................ 236
  foreground ............................................................ 237
  significantDigits ..................................................... 238
Circular Scale Variables .................................................. 239
  polarForeground ....................................................... 239
  impedForeground ...................................................... 241
  admittForeground ...................................................... 242
  significantDigits ..................................................... 243
  cursorScale ............................................................ 244
Trace Variables .......................................................... 245
  type ................................................................. 245
  lineThickness ......................................................... 246
  lineStyle ............................................................. 247
symbolsOn ................................................................. 248
symbolStyle ............................................................... 249
symbolCount .............................................................. 250
dToAConversion .......................................................... 251
dToAUnit ................................................................. 252
modifier ................................................................. 253

**Marker Variables** .................................................... 254
foreground .............................................................. 254
arrowOn ................................................................. 256
fontName ................................................................. 257
fontStyle ................................................................. 258
fontSize ................................................................. 259

**Graph Label Variables** .............................................. 260
foreground .............................................................. 260
direction ................................................................. 262
significantDigits .................................................... 263

**Calculator Variables** ................................................ 264
rpnMode ................................................................. 264
reportVarErrors ....................................................... 265
stackSize ............................................................... 266
xLocation ............................................................... 267
yLocation ............................................................... 268
width ................................................................. 269
height ................................................................. 270
significantDigits .................................................... 271
notation ............................................................... 272

**Table Variables** ..................................................... 273
width ................................................................. 273
height ................................................................. 274
fileSuffix ............................................................... 275
stringColumnWidth .................................................. 276
cellPadding ............................................................ 277
maxColumnWidth ...................................................... 278
notation ............................................................... 279
location ............................................................... 280
significantDigits .................................................... 281
## B
### Shell Environment Variables
- `CDS_WAVESCAN_JRE_DIR` .................................................. 284
- `CDS_WAVESCAN_MAXHEAP` .................................................. 285

## C
### Accelerator Keys
- Graph Window ................................................................. 287
- Calculator ................................................................. 288

## D
### Calculator Functions
- `1/x` ................................................................. 292
- `10**x` ............................................................. 293
- `abs` ................................................................. 294
- `acos` ................................................................. 295
- `acosh` .............................................................. 296
- `angle` ................................................................. 297
- `argmax` ............................................................... 298
- `argmin` ............................................................... 299
- `asin` ................................................................. 300
- `asinh` ............................................................... 301
- `atan` ................................................................. 302
- `atanh` ............................................................... 303
- `average` ............................................................. 304
- `b1f` ................................................................. 305
- `bw (bandwidth)` ......................................................... 306
- `ceil` ................................................................. 308
- `cfft` ................................................................. 309
- `clip` ................................................................. 310
- `compression` .......................................................... 312
- `compressionVRI` ....................................................... 313
- `conj` ................................................................. 314
- `convolve` ............................................................ 315
<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>cos</td>
<td>316</td>
</tr>
<tr>
<td>cosh</td>
<td>317</td>
</tr>
<tr>
<td>cplx</td>
<td>318</td>
</tr>
<tr>
<td>cross</td>
<td>319</td>
</tr>
<tr>
<td>crosscorr</td>
<td>321</td>
</tr>
<tr>
<td>crosses</td>
<td>322</td>
</tr>
<tr>
<td>d2r (degrees-to-radians)</td>
<td>324</td>
</tr>
<tr>
<td>db10</td>
<td>325</td>
</tr>
<tr>
<td>db20</td>
<td>326</td>
</tr>
<tr>
<td>dBm</td>
<td>327</td>
</tr>
<tr>
<td>delay</td>
<td>328</td>
</tr>
<tr>
<td>deltax</td>
<td>329</td>
</tr>
<tr>
<td>deriv</td>
<td>331</td>
</tr>
<tr>
<td>dft (Discrete Fourier Transform)</td>
<td>332</td>
</tr>
<tr>
<td>dftbb</td>
<td>335</td>
</tr>
<tr>
<td>dutycycle</td>
<td>337</td>
</tr>
<tr>
<td>dutycycles</td>
<td>338</td>
</tr>
<tr>
<td>exp</td>
<td>340</td>
</tr>
<tr>
<td>eyeDiagram</td>
<td>341</td>
</tr>
<tr>
<td>falltime</td>
<td>342</td>
</tr>
<tr>
<td>fft</td>
<td>344</td>
</tr>
<tr>
<td>flip</td>
<td>346</td>
</tr>
<tr>
<td>floor</td>
<td>347</td>
</tr>
<tr>
<td>foo</td>
<td>348</td>
</tr>
<tr>
<td>fourEval</td>
<td>349</td>
</tr>
<tr>
<td>freq</td>
<td>350</td>
</tr>
<tr>
<td>freq_jitter</td>
<td>352</td>
</tr>
<tr>
<td>ga</td>
<td>354</td>
</tr>
<tr>
<td>gac_freq</td>
<td>355</td>
</tr>
<tr>
<td>gac_gain</td>
<td>356</td>
</tr>
<tr>
<td>gainBwProd</td>
<td>357</td>
</tr>
<tr>
<td>gainmargin</td>
<td>358</td>
</tr>
<tr>
<td>getAsciiWave</td>
<td>359</td>
</tr>
<tr>
<td>gmax</td>
<td>360</td>
</tr>
<tr>
<td>gmin</td>
<td>361</td>
</tr>
<tr>
<td>gmsq</td>
<td>362</td>
</tr>
<tr>
<td>Function</td>
<td>Page</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------</td>
</tr>
<tr>
<td>gp</td>
<td>363</td>
</tr>
<tr>
<td>gpc_freq</td>
<td>364</td>
</tr>
<tr>
<td>gpc_gain</td>
<td>365</td>
</tr>
<tr>
<td>groupdelay</td>
<td>366</td>
</tr>
<tr>
<td>gt</td>
<td>368</td>
</tr>
<tr>
<td>gumx</td>
<td>369</td>
</tr>
<tr>
<td>harmonic</td>
<td>370</td>
</tr>
<tr>
<td>harmonicFreq</td>
<td>371</td>
</tr>
<tr>
<td>histo</td>
<td>372</td>
</tr>
<tr>
<td>ifft</td>
<td>373</td>
</tr>
<tr>
<td>integ</td>
<td>375</td>
</tr>
<tr>
<td>im</td>
<td>377</td>
</tr>
<tr>
<td>imag</td>
<td>378</td>
</tr>
<tr>
<td>int</td>
<td>379</td>
</tr>
<tr>
<td>integ</td>
<td>380</td>
</tr>
<tr>
<td>ipn</td>
<td>381</td>
</tr>
<tr>
<td>ipnVRI</td>
<td>382</td>
</tr>
<tr>
<td>kf</td>
<td>383</td>
</tr>
<tr>
<td>ln</td>
<td>384</td>
</tr>
<tr>
<td>log10</td>
<td>385</td>
</tr>
<tr>
<td>lsb (Load Stability Circles)</td>
<td>386</td>
</tr>
<tr>
<td>lshift (Left Shift)</td>
<td>387</td>
</tr>
<tr>
<td>mag</td>
<td>388</td>
</tr>
<tr>
<td>max</td>
<td>389</td>
</tr>
<tr>
<td>min</td>
<td>390</td>
</tr>
<tr>
<td>mod</td>
<td>391</td>
</tr>
<tr>
<td>movingavg</td>
<td>392</td>
</tr>
<tr>
<td>nc_freq (Noise Circles - Sweep Frequency)</td>
<td>393</td>
</tr>
<tr>
<td>nc_gain (Noise Circles - Sweep Level)</td>
<td>394</td>
</tr>
<tr>
<td>nf</td>
<td>395</td>
</tr>
<tr>
<td>nfmin</td>
<td>396</td>
</tr>
<tr>
<td>overshoot</td>
<td>397</td>
</tr>
<tr>
<td>period_jitter</td>
<td>399</td>
</tr>
<tr>
<td>phase</td>
<td>400</td>
</tr>
<tr>
<td>phaseMargin</td>
<td>401</td>
</tr>
<tr>
<td>phaseNoise</td>
<td>402</td>
</tr>
<tr>
<td>Function</td>
<td>Page</td>
</tr>
<tr>
<td>-------------------</td>
<td>------</td>
</tr>
<tr>
<td>pow</td>
<td>403</td>
</tr>
<tr>
<td>pp (peak-to-peak)</td>
<td>404</td>
</tr>
<tr>
<td>psd (Power Spectral Density)</td>
<td>405</td>
</tr>
<tr>
<td>psdbb (Power Spectral Density Baseband)</td>
<td>407</td>
</tr>
<tr>
<td>pzbode</td>
<td>408</td>
</tr>
<tr>
<td>pzfilter</td>
<td>409</td>
</tr>
<tr>
<td>r2d (radians-to-degrees)</td>
<td>411</td>
</tr>
<tr>
<td>re</td>
<td>412</td>
</tr>
<tr>
<td>real</td>
<td>413</td>
</tr>
<tr>
<td>risetime</td>
<td>414</td>
</tr>
<tr>
<td>rms (root-mean-square)</td>
<td>417</td>
</tr>
<tr>
<td>rmsNoise</td>
<td>418</td>
</tr>
<tr>
<td>rn</td>
<td>419</td>
</tr>
<tr>
<td>Rn</td>
<td>420</td>
</tr>
<tr>
<td>root</td>
<td>421</td>
</tr>
<tr>
<td>round</td>
<td>422</td>
</tr>
<tr>
<td>s11</td>
<td>423</td>
</tr>
<tr>
<td>s12</td>
<td>424</td>
</tr>
<tr>
<td>s21</td>
<td>425</td>
</tr>
<tr>
<td>s22</td>
<td>426</td>
</tr>
<tr>
<td>sample</td>
<td>427</td>
</tr>
<tr>
<td>settlingTime</td>
<td>429</td>
</tr>
<tr>
<td>sign</td>
<td>431</td>
</tr>
<tr>
<td>sin</td>
<td>432</td>
</tr>
<tr>
<td>sinh</td>
<td>433</td>
</tr>
<tr>
<td>slewrate</td>
<td>434</td>
</tr>
<tr>
<td>snr</td>
<td>435</td>
</tr>
<tr>
<td>spectralPower</td>
<td>437</td>
</tr>
<tr>
<td>sqrt</td>
<td>438</td>
</tr>
<tr>
<td>ssb</td>
<td>439</td>
</tr>
<tr>
<td>stathisto</td>
<td>440</td>
</tr>
<tr>
<td>stddev</td>
<td>441</td>
</tr>
<tr>
<td>tan</td>
<td>442</td>
</tr>
<tr>
<td>tangent (Tangent Line)</td>
<td>443</td>
</tr>
<tr>
<td>tanh</td>
<td>444</td>
</tr>
<tr>
<td>thd (Total Harmonic Distortion)</td>
<td>445</td>
</tr>
</tbody>
</table>
WaveScan User Guide

trim ................................................................................................................. 447
value ............................................................................................................... 449
window ......................................................................................................... 450
x**2 ............................................................................................................. 458
xmax .............................................................................................................. 459
xmin .............................................................................................................. 460
xval ............................................................................................................... 461
y**x ............................................................................................................... 462
ymax .............................................................................................................. 463
ymin .............................................................................................................. 464
yval ............................................................................................................... 465

E
Constants ..................................................................................................... 467

Index ............................................................................................................. 469
Preface

WaveScan is an analog and mixed-signal waveform display tool. This user guide describes WaveScan and explains how to make the best use of it.

This preface discusses the following:

- “What WaveScan Does” on page 16
- “Supported Data Formats” on page 16
- “Related Documents” on page 16
- “Typographic and Syntax Conventions” on page 17
What WaveScan Does

WaveScan helps you analyze the data generated by your simulator.

WaveScan consists of the following components:

- Results Browser displays simulation data in the hierarchical arrangement of your design.
- Graph tool offers features that simplify the processing of your signal data.
- Calculator provides an extensive expression building capability that addresses the needs of a wide variety of analysis types.
- Table tool displays scalar data.

Supported Data Formats

WaveScan can interpret the following data formats.

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSF</td>
<td>Format created by Virtuoso® Spectre Circuit Simulator (including the Virtuoso® Spectre RF Simulation Option) and other simulators integrated into the Virtuoso® Analog Design Environment.</td>
</tr>
<tr>
<td>WSF</td>
<td>Format created by SpectreVerilog. WaveScan supports only digital WSF.</td>
</tr>
<tr>
<td>SST2</td>
<td>Format created by Virtuoso® AMS Simulator. Spectre can also create this format.</td>
</tr>
</tbody>
</table>

Related Documents

For information about related products, consult the sources listed below.

- *Virtuoso Analog Design Environment User Guide*
- *Virtuoso Spectre Circuit Simulator Reference*
- *Virtuoso Spectre Circuit Simulator User Guide*
Typographic and Syntax Conventions

Special typographical conventions distinguish certain kinds of text in this document.

- Boldface words represent elements of the syntax that must be used exactly as presented. Such items include keywords, operators, and punctuation marks. For example,
  
  statefile

- Variables are set in italic font,
  
  maxDirectories

- Vertical bars indicate alternatives. You can choose to use any one of the items separated by the bars. For example,
  
  -graphattributesfile "mygraph"
  | -readstatefile "true | false"

- Square brackets denote optional arguments. For example,
  
  wavescan [-expr MDL]
Running WaveScan

You can start WaveScan in one of the following ways:

- From within the Virtuoso® Analog Design Environment (ADE).
- In the standalone SKILL or Measurement Description Language (MDL) mode.

This chapter describes both of the above modes in the following topics:

- Starting WaveScan on page 20
- Exiting WaveScan on page 21
- Restoring a Session on page 22
- Exiting WaveScan on page 21
# Starting WaveScan

## Starting WaveScan from ADE

- From the CIW, choose *Tools – Analog Environment – Results Browser.*

## Starting the Standalone WaveScan

To start WaveScan, type the following in an xterm window:

```
wavescan
[-expr skill | mdl
  -graphattributesfile "mygraph"
  -readstatefile "true | false"
  -statefile "mystatefile.xml"
  -writestatefile "true | false"
  -V
  -W
  -datadir]
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h</code> or <code>-help</code></td>
<td>Displays information on how to run WaveScan.</td>
</tr>
<tr>
<td>`-expr skill</td>
<td>mdl`</td>
</tr>
<tr>
<td><code>-graphattributesfile</code></td>
<td>Specifies the graph attributes file to be used. The default value is none.</td>
</tr>
<tr>
<td>`-readstatefile&quot;true</td>
<td>false&quot;`</td>
</tr>
<tr>
<td><code>-statefile&quot;mystatefile.xml&quot;</code></td>
<td>Specifies the state file to be used. The default state file is <code>wavescan.xml</code>.</td>
</tr>
<tr>
<td><code>-writestatefalse</code></td>
<td>Specifies whether WaveScan should write a state file when it exits. The default value is true.</td>
</tr>
<tr>
<td><code>-V</code></td>
<td>Displays the version number for WaveScan.</td>
</tr>
<tr>
<td><code>-W</code></td>
<td>Displays the sub-version number for WaveScan.</td>
</tr>
<tr>
<td><code>-datadir</code></td>
<td>Specifies the data directory to be opened on startup.</td>
</tr>
</tbody>
</table>
The first time you start WaveScan, the Results Browser window appears. Then, if you use the 
`-statefile` option, the windows specified in the saved state file open.

### Exiting WaveScan

To exit ADE WaveScan,

➤ From the Graph Window, choose *File* – *Exit*.

All WaveScan windows are closed, but WaveScan keeps running in the background.

To exit standalone MDL WaveScan,

1. Do one of the following:
   - From the Calculator window, choose *Window* – *Exit*.
   - From the Results Browser, Graph Window, or Report Table window, choose *File* – *Exit*.

**Note:** If the Graph Display, Report Table, or Results Browser is the only WaveScan window open, you can also choose *File* – *Close*. If the Calculator is the only WaveScan window open, you can also choose *Window* – *Close*.

The Save Session dialog box appears.

![Save Session dialog box](image)

Clicking the *Save* checkbox saves the session. For more information, see “Saving a Session” on page 22.

2. Click *OK*. 
Saving a Session

When you save a session, the current state of the application is saved. When you reload a saved session, the same windows and settings as in the saved session come up. For more information on the graph attributes that are saved, see “Saving Graphs” on page 109.

To save your session,

1. From the Results Browser window, choose File – Save Session.
2. Click the Save checkbox if you want to save the session.
3. Type the name of the state file if you want to rename it.
4. Click OK.

Restoring a Session

To restore a previously saved session, type the following in an xterm window:

```
wavescan -statefile mystatefile.xml
```

where `mystatefile` is the name of your state file.
Accessing Data

The Results Browser displays signals in a hierarchical arrangement that corresponds to the hierarchy of your design, making it easy for you to locate and manage simulation data.

This chapter discusses the following topics:

- Opening the Results Browser on page 24
- Choosing a Data Directory on page 25
- Selecting Signals on page 26
- Filtering Signals on page 27
- Plotting Signals on page 28
- Creating Tables on page 42
- Using the Calculator on page 43
- Reloading Data on page 43
- Menu Bar and Toolbar on page 43
Opening the Results Browser

To open the Results Browser window from the Virtuoso® Analog Design Environment (ADE),
➤ From the CIW, choose Tools – Analog Environment – Results Browser.

To open the standalone Results Browser window,
➤ Type the following in a terminal Browser window:

    wavescan -expr skill | mdl &

The Results Browser window appears. For information about the parts of the window, click the cross-references.

In ADE, if you run a simulation and then open WaveScan, the Results Browser displays the simulation results directory. Otherwise, the Results Browser opens up blank.

In standalone WaveScan, if you specify a data directory through the WaveScan -datadir command, the directory is displayed. If you type the wavescan & command in the data
directory that you last opened in the previous session, the directory is displayed. Otherwise, the Results Browser opens up blank.

For information on menu and toolbar options, see “Menu Bar and Toolbar” on page 43.

**Note:** When you start WaveScan from a directory that contains a state file, the windows specified in the state file open. For more information, see “Starting WaveScan” on page 20.

### Choosing a Data Directory

To choose a data directory,

➢ Do one of the following:

- In the Results Browser window, type the path in the *Location* field and press the Enter key.

- In the Results Browser window, select a path from the *Location* drop-down. The drop down displays paths to previously opened data directories.

- Follow these steps:

  a. From the Results Browser window, choose *File – Open Results*.

  The Choose Data Directory dialog box appears.

Directories containing data that can be interpreted by WaveScan have the icon next to them.
b. Do one of the following:

- In the Directory name field, type the path to the data directory you want to display.
- Choose the directory you want to display by clicking on it.

c. Click OK.

Selecting Signals

The data from the directory you select appears in the Results Browser window. The left panel displays the data directory, and the right panel displays the associated datasets.

To select a signal,

1. Double-click on a directory in the left panel of the Results Browser.

   The associated datasets appear in the right-hand list view.

   **Note:** The icon next to a directory in either list box indicates that the directory contains subdirectories. You can double-click on the directory to expand (display the subdirectories) or collapse (hide the subdirectories) it.

2. Double-click on the appropriate dataset.

   The associated signals appear in the right-hand list view.

3. Select the desired signal. If the selected signal has a scalar value, you can press the Alt key to display the value. Right-click on the signal to plot it, send it to the calculator buffer, or create a table. For more information on

   - Graphs, see Chapter 3, “Working with Graphs”
   - The calculator, see Chapter 4, “Working with the Calculator”
   - Tables, see Chapter 5, “Working with Tables”

Selecting Multiple Signals

To select a consecutive set of signals,

1. Click on a signal.

2. Shift-click on the final signal in the list.

To select non-consecutive signals,
1. Click on a signal.

2. Ctrl-click on all the other signals.

**Filtering Signals**

You can filter signals by type or by name.

To filter signals,

1. In the Results Browser window, select a dataset.

2. From the *Filter* cyclic field, choose one of the following:

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>All</em></td>
<td>Displays all signals</td>
</tr>
<tr>
<td><em>Logic</em></td>
<td>Displays logic (digital) signals</td>
</tr>
<tr>
<td><em>LogicBus</em></td>
<td>Displays logicbus (digital bus) signals</td>
</tr>
<tr>
<td><em>V</em></td>
<td>Displays voltage signals</td>
</tr>
<tr>
<td><em>I</em></td>
<td>Displays current signals</td>
</tr>
<tr>
<td><em>W</em></td>
<td>Displays power signals</td>
</tr>
</tbody>
</table>

The available choices depend on the selected dataset. For example, the available choices for an analog dataset are *All*, *V*, and *I*.

3. Click ![filter icon] and type the filter pattern.

   WaveScan supports both regular expression syntax and shell syntax

   For example, assume you want to display all voltage signals whose names begin with *net*. First, select *V* from the *Filter* cyclic field. Then do one of the following:

   - If you are using regular expression syntax, type `net.*` to display voltage signals starting with *net*.
   - If you are using shell syntax, type `net*` to display voltage signals starting with *net*. For more information, see “textFilterType” on page 205.

4. Click ![execute button].

   The signals meeting the filtering criteria you specified are displayed.
Plotting Signals

This section describes the available destinations and graph types for your signal. It then describes how to use these options while creating a graph.

About the Signal Destination

If you are creating a graph for the first time, it is displayed in a new graph window. If you have a graph window already open, you need to specify the destination for the new signal. This section describes the available destination choices. For information on how to specify the signal destination, see step 2 in “Creating Graphs” on page 32.

Appending to a Graph

You can add a signal to the selected graph. If the trace shares the same unit, it is assigned to the same Y axis. Otherwise, it is assigned to a new Y axis. If the selected graph window already has four Y axes, or if the units do not match, WaveScan plots the signal to a new subwindow.
The following figure shows the trace for the \texttt{V2:p} signal appended to the graph containing the trace for the \texttt{out} signal.

Note that the Y axis unit for \texttt{out} is V and the Y axis unit for \texttt{V2:p} is mA. WaveScan assigns the trace for \texttt{V2:p} to a new Y axis.

**Replacing a Signal**

You can replace a signal or group of signals in a graph window or subwindow.
Adding a New Subwindow

You can create a new graph in a subwindow within the selected Graph Window. The following figure shows the graph for the $V_2 : p$ signal created as a subwindow.

Adding a New Window

You can plot a signal to a new graph window.

About the Graph Type

Different types of graphs are suited for different kinds of data. WaveScan supports rectangular, polar, admittance, impedance, and real versus imaginary graphs. Transient data can be plotted only on a rectangular graph but complex data, such as AC data, can be plotted in various ways.
This section describes the available graph types. For information on how to specify the graph type, see step 3 in “Creating Graphs” on page 32.

Default

The default graph type is based on the data type information in the simulator data file. For example, the default graph type for transient data is rectangular.

Rectangular Graphs

Transient and DC sweep data are always plotted on a rectangular graph. You can also plot portions of complex data on a rectangular graph by selecting the modifier (real, imaginary, magnitude, or phase). WaveScan plots the selected modifier against frequency.

Polar Graphs

Polar graphs represent data with respect to a polar coordinate system rather than a rectangular coordinate system. A polar graph plots a point at a given radial distance along a ray that creates a given angle with the positive X axis.

The following example illustrates how you can plot a point (45 degrees, 1).

Admittance and Impedance Graphs

Admittance and impedance graphs are a direct graphical representation, in the complex plane, of the complex reflection coefficient. They reveal the complex impedance anywhere along a line.

The center of the chart is always a perfect match. This normally represents 50 ohms but can be any impedance line you want — it is normalized to 1.0 units. Everything is scaled relative
to the unit you pick. The nature of impedance is that of a real or resistive portion, and an imaginary, or reactive portion, combined Pythagorean style.

There are four goalposts spaced 90 degrees apart graphically and 45 degrees apart electrically. Two goalposts are resistive, one a short and the other an open. These are the left and right respective sides. The top and bottom posts are reactive, either inductive, or capacitive. Every point in between represents the various combinations resulting from a mismatched condition.

You can display either impedance or admittance grids within the smith chart you create — the grids are mirror images of each other.

**Real Vs Imag**

Real Vs Imag graphs plot the real part against the imaginary part. These graphs are available only for AC data.

**Creating Graphs**

To create a graph,

1. From the Results Browser window, select the signal you want to plot.

2. Choose Settings – Plot Style to specify the signal destination. For more information, see “About the Signal Destination” on page 28.

3. Choose Settings – Graph Type and specify the graph type.

4. Do one of the following:
   - Middle-click or double-click on the selected signal.
   - Click .

The Graph Window appears. If a graph window is already open and you selected Append in Step 2, WaveScan plots the trace in the active window or subwindow. For more information, see Chapter 3, “Working with Graphs.”

**Data Ranging**

The data ranging feature in WaveScan makes it easy to use a very large dataset efficiently by opening just the portion of the dataset that you need. You can specify a particular time range in a transient analysis, and then open the dataset and plot signals over just that range.
Plotting a Signal Over a Time Range

You can plot transient data over a time range.

To specify the time range for a transient dataset,

1. In the Results Browser window, select a dataset.
2. Choose Settings – Select Data.

   The Data Selection dialog box appears.

   ![Data Selection Dialog Box]

   The Start and End fields display the whole range of the signal.

3. In the Start field, type the time at which you want the plot to begin.
   If you want to display the time for the first data point in the signal, click default.

4. In the End field, type the time at which you want the plot to end.
   If you want to display the time for the last data point in the signal, click default.

5. Click OK.

6. Click .

   The Graph Window appears with a plot for the specified time range.
An Example of a Time Range Sweep

The following figure represents a signal from a transient dataset.

The following figure represents the same signal plotted over a time range of 60ns to 240ns.
You cannot zoom out or pan this graph to read anything before 60ns and after 240ns. To display the whole graph, follow steps 1 through 6 again, using default values in the Start and End fields in the Sweep Selection dialog box.

**Plotting Parametric Swept Data**

A parametric analysis sweeps a parameter or group of parameters and executes one or more analyses for each combination of parameters. Wavescan lets you post-process the resulting data efficiently.

To plot a part of parametric swept data,

1. In the Results Browser window, select a parametric signal.
2. Choose *Settings – Select Data*.
   
   The Sweep Selection dialog box appears, displaying a list of inner sweep variables.
3. Select the values which you want to plot and click *OK*.
4. Click ![plot](image).
   
   The Graph Window appears with plots for all the combinations of sweep variables.

**An Example of Plotting Data from a Parametric Sweep**

To plot selected curves from parametric data,

1. In the Results Browser window, use the tutorial data and double-click on the `sweepTemp_tran_sweep-sweep` folder. For more information on tutorial data, see
2. Choose *Settings – Select Data*. 
The Data Selection dialog box appears. The *Start* and *End* fields display the whole range of the signal and all temperature and vdd values are selected by default.

3. In the *temp* listbox, click 25.0, hold down the Ctrl key and click 75.0 and 125.0.
4. In the *vdd* (inner seep variable) listbox, click 4.5, press the Shift key and click 5.1.

The Data Selection dialog box now looks as shown in the figure below.

5. Click *OK*.
6. In the Results Browser window, right-click on the out signal and choose *New Win*.
The Graph Window appears with traces for the parametric family. Each trace in the family is annotated by a sweep path describing each parameter – value pair.

You can bring these traces to the calculator and apply the `max` function to them. Plotting `max V(out)` results in the maximum value of each trace on the Y axis, and the `vdd` values used in the sweep on the X axis. Each trace corresponds to a different temp value. For more information on the calculator, see Chapter 4, "Working with the Calculator."

**Plotting Y Versus Y for Two Signals**

This section describes how to plot the Y-axis values of one signal versus the Y axis values of another signal. You can plot Y vs Y to measure input offset voltage, which displays the offset between the input of the circuit and the output.

To plot Y versus Y,

1. In the Results Browser window, select a signal.
2. Click `Leod`.
3. Select the second signal.
The Graph Window appears with a trace for Y vs Y.

**Note:** To cancel the procedure, press *Esc* before step 3.
The following figures illustrate the `out` and `in_m` signals from the tutorial data.
When you plot Y vs Y for these signals, the result is a diagonal line because the signals are identical.

**Plotting the Difference of Two Signals**

To plot the difference of two signals,

1. In the Results Browser window, select a signal.
2. Click .
3. Select the second signal.

The Graph Window appears with a plot of the difference between the signals.

**Note:** To cancel the procedure, press Esc before step 3.
The following figures illustrate an example.
Creating Tables

This section describes how to display simulation data in a tabular format.

To create a table,

1. In the Results Browser window, select the dataset, signals, or signal you want to display in a table.

2. Do one of the following:

   - Choose Tools – Table.
   - Right-click on the selected dataset or signal and choose Create Table from the pop-up menu.

   The Report Table window appears. For more information, see Chapter 5, “Working with Tables.”

**Note:** You cannot append to a table. To create a table for multiple signals, select them together in the Results Browser window.
Using the Calculator

To open the calculator,

➤ In the Results Browser window, choose Tools – Calculator.

The Calculator window appears. For more information, see Chapter 4, “Working with the Calculator.”

Reloading Data

While resimulating your design, you can

■ Update the data directory. The Results Browser re-reads the data. You can thus look at results as the simulation is running. All new graphs you create display the updated data, which helps you in monitoring long simulations. Existing graphs are not updated.

■ Reload the data directory. The Results Browser re-reads the data and updates all associated graphs except frozen graphs. For more information on freezing graphs, see “Freezing Graphs” on page 105.

To update a data directory,

■ In the Results Browser window, choose the data directory from the Location pull-down.

■ Click in either panel.

■ Choose File – Open to open the data directory again.

■ Click on a toolbar icon or menu option.

To reload a data directory and update all associated graphs,

➤ Select the data directory and choose File – Reload.

Note: If the Graph Window displays a graph of data that does not exist after the data directory is reloaded, the graph is erased unless it is frozen.

Menu Bar and Toolbar

This section describes the menu and toolbar of the Results Browser window.
# Menu Bar

For a description of the menu commands, see the table below.

<table>
<thead>
<tr>
<th>Item</th>
<th>Usage and Description</th>
<th>For More Information, See</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>File</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Open Results</td>
<td>Opens the specified data directory.</td>
<td>“Choosing a Data Directory” on page 25.</td>
</tr>
<tr>
<td>Open Graph</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Open Graph as Plot</td>
<td>Opens the specified graph.</td>
<td>“Opening Saved Graphs” on page 112.</td>
</tr>
<tr>
<td>Open Graph as Template</td>
<td>Opens the specified graph as a template.</td>
<td>“Creating Graph Templates” on page 113.</td>
</tr>
<tr>
<td>Open Table</td>
<td>Opens the specified table.</td>
<td>“Opening Saved Tables” on page 181.</td>
</tr>
<tr>
<td>Clear</td>
<td>Clears the selected dataset from the Results Browser window.</td>
<td></td>
</tr>
<tr>
<td>Reload</td>
<td>Updates the selected data directory and graphs that are not frozen.</td>
<td>“Reloading Data” on page 43.</td>
</tr>
<tr>
<td>Save Session</td>
<td>Saves the current session.</td>
<td>“Saving a Session” on page 22.</td>
</tr>
<tr>
<td>Close</td>
<td>Closes the Results Browser window. If this is the only WaveScan window open, Close exits WaveScan.</td>
<td></td>
</tr>
<tr>
<td>Exit</td>
<td>Available only in the standalone WaveScan. Closes all windows and exits WaveScan.</td>
<td>“Exiting WaveScan” on page 21.</td>
</tr>
</tbody>
</table>
## WaveScan User Guide
### Accessing Data

<table>
<thead>
<tr>
<th>Item</th>
<th>Usage and Description</th>
<th>For More Information, See</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Settings</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Select Data</strong></td>
<td>Opens the Data Selection dialog box.</td>
<td>“Plotting Parametric Swept Data” on page 35.</td>
</tr>
<tr>
<td><strong>Plot Style</strong></td>
<td>Sets the graph destination.</td>
<td>“About the Signal Destination” on page 28.</td>
</tr>
<tr>
<td>Append</td>
<td>Adds the signal to the selected graph.</td>
<td></td>
</tr>
<tr>
<td>Replace</td>
<td>Replaces the selected graph with the new one.</td>
<td></td>
</tr>
<tr>
<td>New SubWin</td>
<td>Plots the signal in a new subwindow.</td>
<td></td>
</tr>
<tr>
<td>New Win</td>
<td>Plots the signal in a new window.</td>
<td></td>
</tr>
<tr>
<td><strong>Graph Type</strong></td>
<td></td>
<td>“About the Graph Type” on page 30.</td>
</tr>
<tr>
<td>Default</td>
<td>Plots the signal to the default graph type.</td>
<td></td>
</tr>
<tr>
<td>Rectangular</td>
<td>Plots the signal to a rectangular graph.</td>
<td></td>
</tr>
<tr>
<td>Polar</td>
<td>Plots the signal to a polar graph.</td>
<td></td>
</tr>
<tr>
<td>Impedance</td>
<td>Plots the signal to an impedance graph.</td>
<td></td>
</tr>
<tr>
<td>Admittance</td>
<td>Plots the signal to an admittance graph.</td>
<td></td>
</tr>
<tr>
<td>Real vs Imag</td>
<td>Plots real versus imaginary.</td>
<td></td>
</tr>
</tbody>
</table>
### Toolbar

For a description of the toolbar, see the table below.

<table>
<thead>
<tr>
<th>Item</th>
<th>Usage and Description</th>
<th>For More Information, See</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Open</strong></td>
<td>Opens the Choose Data Directory dialog box.</td>
<td>“Choosing a Data Directory” on page 25.</td>
</tr>
<tr>
<td><strong>Plot Signal</strong></td>
<td>Plots the selected signal or signals. This icon is enabled only if one or more signals is selected in the Results Browser window.</td>
<td>“Plotting Signals” on page 28.</td>
</tr>
<tr>
<td><strong>Signal to calculator</strong></td>
<td>Opens the Calculator window with the selected signal displayed in the buffer.</td>
<td>“Using the Calculator” on page 43.</td>
</tr>
<tr>
<td><strong>Difference of Two Signals</strong></td>
<td>Plots the difference of the selected signals. This icon is enabled only if a signal is selected in the Results Browser window.</td>
<td>“Plotting the Difference of Two Signals” on page 40.</td>
</tr>
<tr>
<td><strong>Y vs Y for two signals</strong></td>
<td>Plots the Y axis values of one signal versus the Y axis values of another signal at the same time points. This icon is enabled only if a signal is selected in the Results Browser window.</td>
<td>“Plotting Y Versus Y for Two Signals” on page 37.</td>
</tr>
</tbody>
</table>
Graph Type

This pull-down, which is available only for AC data, specifies the graph type for plotting the signal.

<table>
<thead>
<tr>
<th>Item</th>
<th>Usage and Description</th>
<th>For More Information, See</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>Plots the signal to the default graph type.</td>
<td></td>
</tr>
<tr>
<td>Rectangular</td>
<td>Plots the selected part of the AC signal (see “Trace Modifier” on page 47) to a rectangular graph.</td>
<td></td>
</tr>
<tr>
<td>Polar</td>
<td>Plots the signal to a polar graph.</td>
<td></td>
</tr>
<tr>
<td>Impedance</td>
<td>Plots the signal to an impedance graph.</td>
<td></td>
</tr>
<tr>
<td>Admittance</td>
<td>Plots the signal to an admittance graph.</td>
<td></td>
</tr>
<tr>
<td>Real vs Imag</td>
<td>Plots real versus imaginary.</td>
<td></td>
</tr>
</tbody>
</table>

Trace Modifier

This pull-down, which is available only for AC data, specifies the X axis scale used to plot data on a rectangular graph.

You can change the default trace modifier through the modifier variable in the .cdsenv file.

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mag</td>
<td>Plots magnitude versus frequency.</td>
</tr>
<tr>
<td>Phase</td>
<td>Plots phase versus frequency.</td>
</tr>
</tbody>
</table>
WaveScan User Guide
Accessing Data

### Signal Destination

<table>
<thead>
<tr>
<th>Item</th>
<th>Usage and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Append</td>
<td>Adds the signal to the selected graph.</td>
</tr>
<tr>
<td>Replace</td>
<td>Replaces the selected graph with the new one.</td>
</tr>
<tr>
<td>New SubWin</td>
<td>Plots the signal in a new subwindow.</td>
</tr>
<tr>
<td>New Win</td>
<td>Plots the signal in a new window.</td>
</tr>
</tbody>
</table>

### Location

<table>
<thead>
<tr>
<th>Item</th>
<th>Usage and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Lets you type the path to the data directory you want to open.</td>
</tr>
<tr>
<td></td>
<td>Displays a pull-down containing paths to the last 20 data directories opened in WaveScan.</td>
</tr>
</tbody>
</table>

### Status Bar

Displays warning and errors or prompts you for further action. For example, when you click 

![](image)

the status bar displays the following message:

> Please select the second signal. Press <ESC> to cancel.
Working with Graphs

WaveScan helps you in representing your simulation data graphically. You can customize your graphs and use markers and labels to annotate them.

This chapter discusses the following topics:

- Opening the Graph Window on page 53
- Customizing a Graph on page 55
  - Changing the Graph Layout on page 55
  - Handling Graph Objects on page 58
  - Zooming a Graph on page 59
  - Panning a Graph on page 62
  - Editing Graph Attributes on page 63
  - Working With Traces on page 65
  - Editing Axis Attributes on page 75
  - Editing Scale Attributes for Circular Graphs on page 80
  - Working with Cursors on page 82
  - Working with Labels on page 89
  - Working with Markers on page 93
  - Working With Buses on page 102
- Saving and Loading Traces on page 103
- Freezing Graphs on page 105
- Printing Graphs on page 105
- Saving Graphs on page 109
WaveScan User Guide
Working with Graphs

- Opening Saved Graphs on page 112
- Creating Graph Templates on page 113
- Menu Bar and Toolbar on page 115
Opening the Graph Window

To open the Graph Window, do one of the following:

■ If you are in Virtuoso® Analog Design Environment (ADE) WaveScan,
  a. From the CIW, choose *Tools – Analog Environment – Waveform*.

■ If you are in the standalone WaveScan (SKILL or MDL),
  a. From the Results Browser window, select the signal you want to plot.
  b. Middle-click or double-click on the selected signal.

**Note:** In the SKILL mode, if the Results Browser window is closed, you can use the `drbBrowserFormCB` function (see the *Virtuoso Analog Design Environment SKILL Language Reference*) to open the Results Browser and the `awvCreatePlotWindow` function to open the Graph Window.
The Graph Window appears. For information about the parts of the window, click the cross references.

For digital traces, if the trace name is too long, it is truncated. To see the full name, place the cursor over the name and press the Alt key.

For information on menu and toolbar options, see “Menu Bar and Toolbar” on page 115.

For more information on creating a graph, see Plotting Signals on page 28.

To see the available accelerator keys, choose Help – Accelerator. For more information, see Appendix C, “Accelerator Keys.”
Customizing a Graph

Once you create a graph, you can customize it to help you analyze your data.

Changing the Graph Layout

A Graph Window may have several subwindows. You can change the way subwindows are displayed within the Graph Window.

To change the graph layout,

➤ Choose one of the following options:

- **Graph – Layout – Auto**
  The default graph layout is determined by the aspect ratio. If the window is wider than it is tall, the graphs are displayed side by side; if the window is taller than it is wide, graphs are displayed one below the other.

- **Graph – Layout – Vertical**
  The following figure displays subwindows arranged vertically.
- **Graph – Layout – Horizontal**

The following figure displays subwindows arranged horizontally.

- **Graph – Layout – Card**

This makes each subwindow the size of the Graph Window and stacks the subwindows one on top of the other. Thus, only one subwindow is displayed at a time. There is a small
numbered box at the top right corner of each subwindow which identifies the subwindow. Select the box for the subwindow you want to display.

Highlighted numbered box indicates the current graph

![Graph Window](image)

### Changing the Graph Colors

The default color scheme for the Graph Window is

- black in Virtuoso® Analog Design Environment (ADE) WaveScan and standalone SKILL WaveScan
- white in standalone MDL WaveScan

To change the color scheme,

> In the Graph Window, choose **Graph – Color Schemes** and select the appropriate color.

The current graph’s background color is changed to the color you specified. All graphs you create (in this as well as subsequent sessions) will have the background color you specified.
Handling Graph Objects

This section describes how you can select, hide, reveal, and delete a graph and its components (traces, axes, markers, labels).

Selecting Objects

You can select a graph, trace, axis, marker, or label, and change its attributes.

You can click on graph, axis, marker, or label to select it. You can select multiple objects by holding down the Control key and clicking on each of them.

If the autoTraceSelect variable is set to true (the default value) in the .cdsenv file, WaveScan selects the trace nearest to the cursor. This is especially useful if you have multiple traces on a graph. If the autoTraceSelect variable is set to false, you must click on a trace to select it.

Hiding Objects

You can hide labels, markers, legends, traces, axes and graphs. For example, you can Concentrate on a particular trace in a graph by hiding the other traces.

To hide an object,

1. Select the object.
2. Choose Edit – Hide.
   WaveScan hides the selected object.

Revealing Objects

You can reveal all hidden objects.

To reveal an object,

1. Choose Edit – Reveal.
   A list of hidden objects is displayed.
2. Choose the object you want to reveal.
   WaveScan reveals the specified object.
Swapping Objects

You can swap traces in a strip chart or digital graph, two dependent axes within a graph that has multiple Y-axes, or graphs within the same window.

To swap objects,
1. Select an object.
2. Choose *Edit – Swap*.
3. Choose the second object.

WaveScan swaps the first object with the second object.

Deleting Objects

You can delete labels, markers, legends, traces, and graphs. You cannot delete the last graph in the Graph Window.

To delete an object,
1. Select an object.
2. Choose *Edit – Delete*, or press the *Del* key.

WaveScan deletes the selected object.

Undoing Actions

You can undo most menu selections.

To undo an action,

➤ Choose *Edit – Undo*.

WaveScan undoes your most recent action.

Zooming a Graph

Do one of the following to zoom a graph,

■ Click, drag and release the right mouse button.

The corners of the box from where you click to where you release will be the size of the new graph.
Note: If you set the rightMouseZoom variable in the .cdsenv file to false, use the middle-mouse button to zoom.

Choose one of the following options from the Zoom menu:

- **Zoom**
  Click to begin the box. Drag till the zoom box is the size you want the new graph to be, and then release the mouse button.

- **X-Zoom**
  Displays the ← cursor. Your first click defines the left X axis edge of the zoom, and the release of the mouse button defines the right X axis edge. The graph is zoomed such that these two values are the end points of the X axis.

- **Y-Zoom**
  Displays the ↓ cursor. Your first click defines the top Y axis edge of the zoom, and the release of the mouse button defines the bottom Y axis edge. The graph is zoomed such that these two values are the end points of the Y axis.

- **Unzoom**
  Incrementally reverses a series of zooms.

- **Fit**
  Returns the graph to the unzoomed state.

- **Zoom In**
Zooms in with respect to the mid-point of the graph. The zoomed graph is half the size of the original graph.
Zooms out with respect to the mid-point of the graph. The zoomed graph is double the size of the original.

**Note:** In a digital graph, zoom affects the X (time) dimension only.

**Panning a Graph**

To pan a graph,

➤ Select one of the following options from the *Pan* menu

  🟠 *Pan Right*
The figure below illustrates how WaveScan pans a graph.

![Graph panned right](image)

Thus if the X axis of a graph goes from 0 to 200, pan right moves it from 40 to 240.

- **Pan Left**
- **Pan Up**
- **Pan Down**

**Note:** In a digital graph, pan affects the X (time) dimension only.

### Editing Graph Attributes

The default graph attributes are controlled by the values assigned to variables in the `.cdsenv` file. For more information, see Appendix A, “WaveScan Variables.”

To edit graph attributes,

1. Double-click in the Graph Window.
The Graph Attributes dialog box appears.

2. In the Title field, type the graph title or select Default to display the default title. See the figure in “Working with Labels” on page 89 to see a graph title.

   Selecting Default disables the Title field.

3. In the SubTitle field, type the graph subtitle or select Default to display the sub-default title.

   Selecting Default disables the SubTitle field.

4. In the Legend Rows field, specify the number of legend rows to be displayed in the Graph Window. A scrollable list is displayed after this number is exceeded.

   When you exit WaveScan, this gets reset to the value of the VisibleLegendRows variable in the .cdsenv file.

5. Select the Show checkbox to display legends for traces in the Graph Window.

6. In the Foreground/Background fields, select the foreground and background color for the graph. The foreground color controls the color of the graph title and subtitle, date, and axes as well as axes labels.

7. In the Major Grids field, select Show to display major grids and select the grids color.

8. In the Minor Grids field, select Show to display minor grids and select the grids color.

9. Click OK.
Working With Traces

You can drag and drop traces from a:

- graph window to another. If you drag an ac trace to a transient graph, the X-axis label displays a series of question marks to indicate that the resultant axis has no value.
- subwindow to another
- digital graph to an analog graph. WaveScan creates a clone of the digital trace in the analog graph.
- strip to another

While you are dragging the trace, its name appears in the cursor as below:

`--< net10 -->`

You can select a trace and choose one of the following menu options in the Graph Window to manipulate traces:

- **Trace – New Graph – Copy New Window**
- **Trace – New Graph – Move New Window**
- **Trace – New Graph – Copy New SubWindow**
- **Trace – New Graph – Move New SubWindow**

This is especially useful when you want to study a single trace from a set of parametric leaf waveforms.

If you did not select a trace before choosing one of the above options:

- **SKILL** WaveScan creates an empty new graph window or subwindow.
- **Standalone MDL** WaveScan does nothing.

Displaying Traces in Strips

In a Graph Window containing multiple traces, you can display each trace in an individual strip. The strips are aligned vertically. Each strip has its own Y axis and shares the X axis with other strips.

You can display traces in strips only in rectangular graphs.

To display traces in strips,
Choose **Axis – Strips**.

Each trace is displayed in a separate strip with its own Y axis.

The following figure shows a graph window containing multiple traces.
The following figure shows each trace displayed in a strip.

You can drag and drop traces between strips. If you release the trace

- in a strip, WaveScan adds the trace to the strip in an analog graph
- outside a strip, WaveScan puts the trace in a new strip below the strip closest to the release point
Adjusting Graph Area

In digital and mixed-signal graphs, you can adjust size of the digital row by dragging the handle just below the last visible digital trace.

Each digital trace is compressed or expanded to adjust to the new row size. Analog graphs in the same Graph window are also adjusted.

Displaying Symbols

To display symbols on data points on a trace, do one of the following:

➤ Select a trace and choose Trace – Symbols On.

You can control the type of symbol, and the number of data points to be identified by symbols. The symbol used for the trace is displayed next to the trace name. For more information, see “Editing Trace Attributes” on page 71.
The following figure illustrates how two traces can be distinguished by using symbols to display the data points for each of them.

Assigning a Trace to an Axis

In a window or subwindow containing multiple traces, you can assign a trace to

- a new Y axis. WaveScan re-scales the new Y axis.
- an existing Y axis. If the units do not match, WaveScan puts in a series of question marks in the axis label to indicate that the resultant axis has no physical value.

The maximum number of Y axes in a window or subwindow is four. After this, if you assign a trace to a new Y axis, WaveScan automatically puts that trace in a new subwindow.

To assign a trace to an axis,

1. Select a trace.
2. Choose Trace – Assign to Axis and select an axis. The menu displays the existing axes as well as a new one.

The trace is assigned to the specified Y axis.
Example of Assigning a Trace to an Axis

Example 1

The left graph in the following figure shows two traces on the same axis. The signal \texttt{net35} appears flat. The right graph shows \texttt{net35} assigned to a new axis.
Example 2

The left graph in the following figure shows two traces on two Y axes – one is voltage and the other is current. The right graph shows both traces assigned to the same axis. Since the units do not match, the Y axis label is “???” (V?A).

Editing Trace Attributes

The default trace attributes are controlled by the values assigned to variables in the .cdsenv file. For more information, see Appendix A, “WaveScan Variables.”

To edit trace attributes for an analog trace,

1. Double-click on a trace.
The Trace Attributes dialog box appears.

2. In the **Name** field, type the name for the trace or select the **Default** checkbox to display the default name. See the figure in “Working with Labels” on page 89 to see how the trace name is displayed.

Selecting **Default** disables the **Name** field.

**Note:** The **Name** and **Default** fields are grayed out if more than one trace is selected.

3. In the **Strip Chart Visible Rows** field, specify the maximum number of strip graphs to be displayed in the Graph Window. A scrollable list is displayed after this number is exceeded. This field is disabled if there is only one trace in the Graph Window.

4. In the **Type/Style** fields, do the following:
   
   a. Specify whether you want the trace to be represented by a **Line**, **Points**, **Bars**, **Spectral**, or a **Histogram**.
   
   b. Specify whether you want the trace style to be **Solid**, **Dashed**, **Dotted**, or **DotDashed**.
The following figure displays these styles.

![Graph styles](image)

c. Specify whether you want the trace to be **Fine**, **Medium**, or **Bold**.

5. Specify whether you want data points to be displayed on the trace as symbols of **Point**, **Dot**, **Plus**, **Square**, **Box**, **UpArrow**, **DownArrow**, **X**, or **Circle**.

6. Select the **Show** checkbox to display data points on the trace, and specify the number of points to be displayed.

7. In the **Foreground** field, select the foreground color for the trace.

8. In the **Cursor X/Y Offset** fields, specify the offset for the X axis and Y axis, if desired. The trace cursor is displayed relative to these offsets. The X offset is added to the horizontal location of a point on the trace while the Y offset is added to the vertical location of the point.

9. Click **OK**.

To edit trace attributes for a digital trace,

1. Double-click on a trace.
The Trace Attributes dialog box appears.

2. In the **Name** field, type the name for the trace or select the **Default** checkbox to display the default name. See the figure in “Working with Labels” on page 89 to see how the trace name is displayed.

   Selecting **Default** disables the **Name** field.

   **Note:** The **Name** and **Default** fields are grayed out if more than one trace is selected.

3. In the **Digital Visible Rows** field, type the maximum number of digital rows to be displayed in the Graph Window for mixed-signal graphs. The digital signals are displayed in a scrollable list after this number is exceeded.

   This field does not work for digital-only graphs – the Graph Window displays the maximum number of digital traces possible. For example, if you add 10 digital traces to a graph, all of them are displayed in the Graph Window irrespective of the value of the **Digital Visible Rows** field. If you now add an analog trace to the graph, only three (default value of the **Digital Visible Rows** field) digital traces are visible on the graph. You can look at the rest of the digital traces by scrolling down.

4. In the **Style** fields, do the following:

   a. Specify whether you want the trace style to be **Solid, Dashed, Dotted, or DotDashed**.

   b. Specify whether you want the trace to be **Fine, Medium, or Bold**.

5. In the **Foreground** field, select a color for the trace.
6. In the *Cursor X/Y Offset* fields, specify the offset for the X axis and Y axis, if desired. The coordinates of the trace cursor are displayed relative to these offsets. The X offset is added to the horizontal location of a point on the trace while the Y offset is added to the vertical location of the point.

7. In the *D to A max/unit* fields, type the maximum scaling and unit for the Y axis.

   If you want to superimpose a digital trace on an analog graph for comparison, these fields set the value and unit for the logical high so that the digital trace can be scaled on the analog graph. In the following example, logical high has been set to 500mV.

8. Click *OK*.

**Editing Axis Attributes**

The default axis attributes are controlled by the values assigned to variables in the *.cdsenv* file. For more information, see Appendix A, “WaveScan Variables.”

To edit axis attributes,

1. Double-click on an axis.
The Axis Attributes dialog box appears. It resembles the picture shown below if your trace is from a DC swept or parametric non-transient dataset.

If you selected the Y-axis, the *Eye diagram Interval*, *Eye On*, and *Sweep Var* fields do not appear.

2. In the *Label* field, type the label title or select *Default* to display the default axis label. Selecting *Default* disables the *Label* field. WaveScan reads the default axis label from the simulation data file.

3. In the *Scaling* field, choose one of the following:
   - *Auto* lets WaveScan select the scale limits and axes divisions.
   - *Min-Max* lets you set the beginning and ending axis values.
   - *Manual* lets you set the beginning and ending axis values as well as the major and minor divisions.
The left graph in the following figure is auto-scaled and the right is manually scaled.

4. Select Log to display the axis in logarithmic scale.

5. Select Origin to force the selected axis to contain the origin.

6. For eye diagrams, do the following:
   a. Select Eye On.
      
      The Scaling field is set to Min-Max so that you can set the starting and ending point for the eye diagram.

   b. In the Eye diagram Interval field, type the interval.

   These fields appear only if you are editing the X-axis attributes.

7. In the Min/Max fields, type the minimum and maximum values for the selected axis. You can specify this only if Scaling is Min/Max or Manual.

8. In the Major/Minor Divisions field, specify the major and minor divisions for the axis. You can specify this only if Scaling is Manual.

9. In the Significant Digits field, type the number of significant digits for the axis. Select Default to display the default number of significant digits. Selecting Default disables the Significant Digits field.

   By default, WaveScan displays the number of significant digits required to display all labels individually for an axis.

10. In the Foreground field, select the foreground color for the axis.

11. When editing the X-axis attributes, do the following, as relevant:
If the trace you selected was from a non-parametric swept analog dataset, the *Plot Vs* field displays all signals in your dataset. Choose the signal whose Y-axis you want to plot versus the Y-axis of the current graph. This has the same effect as plotting Y Versus Y for two signals from the Results Browser window.

If the trace you selected was from a DC swept or parametric non-transient dataset, the *Sweep Var* field displays all swept variables. Choose the sweep variable whose values you want on the X-axis of the current graph. See the example after this procedure for more explanation.

You can swap sweep variables only if

- the swept dataset is non-transient.
- the innermost sweep variable has less than 200 data points.

If the trace you selected was from an AC dataset, the *Plot Vs* field displays all signals in your dataset. Choose the signal whose Y-axis you want to plot versus the Y-axis of the current graph. Choose the X-axis modifier for the graph from the cyclic field next to the *Plot Vs* field.

12. Click *OK*.

**Example for Swapping Sweeps**

1. In the Results Browser window, open the `sweeptran.raw` folder. For information on how to access this dataset, see *WaveScan Tutorial*.

2. Select the `out` signal and click *Signal to calculator* ( ).
   The Calculator window opens. `V(out)` appears in the buffer.

3. Click `max`.
   
   `max(V(out))` appears in the buffer.

4. Click *Plot Signal* ( ).
The Graph Window appears with a plot of the expression. This plot represents the maximum values of each waveform (Y-axis) for each value of vdd (X-axis). Each of the five traces represents a different swept temperature.

5. Double-click on the X-axis.

The Axis Attributes dialog box appears. The *Sweep Var* field displays both swept variables – *vdd* and *temp*. 
6. In the **Sweep Var** field, choose *temp* and click **Apply**. The Graph Window changes to

The X-axis now displays the temperature and each trace represents a different vdd value.

**Editing Scale Attributes for Circular Graphs**

The default scale attributes for circular graphs are controlled by the values assigned to variables in the `.cdsenv` file. For more information, see Appendix A, “WaveScan Variables.”

To edit scale attributes for a circular graph,

1. Double-click on a circular grid.
The Scale Attributes dialog box appears.

2. In the **Scaling** field, select one of the following:

- **Auto** scales the grids automatically.
- **Manual** lets you set the maximum radial limit, and the number of circular and radial grids.

The left figure displays a polar graph scaled automatically by WaveScan and the right figure shows the same graph scaled manually to a higher resolution.
3. In the *Manual* field, specify the radial scaling limit. You can specify this only if *Scaling* is *Manual*.

4. In the *Circ/Radius* field, specify the number of circular and radial grid divisions you want in the graph. You can specify these only if *Scaling* is *Manual*.

5. In the *Show* section, select the grids you want to be displayed on your graph.

6. In the *Foreground* section, select the grid color.

7. In the *Significant Digits* field, type the number of significant digits for the grids. Select *Default* to display the default number of significant digits. If you select *Default*, the *Significant Digits* field is disabled.

8. In the *Cursor Scale* field, select one of the following options to specify the scaling for cursor coordinate display:
   - *Real/Imaginary* or *Mag/Phase* for polar graphs.
   - *Impedance*, *Admittance*, or *Reflection* for admittance or impedance graphs.

9. Do one of the following:
   - Click *OK* to apply the changes and close the Scale Attributes dialog box.
   - Click *Apply* to apply the changes and keep the Scale Attributes dialog box open.

**Working with Cursors**

**Using a Trace Cursor**

The trace cursor is a moveable cursor that follows the contour of the trace. The X and Y coordinates of the trace cursor are displayed in the lower-left corner of the Graph Window.
The lower-right corner of the Graph Window displays the coordinates of the location of the system mouse cursor.

To enable the trace cursor,

➤ Choose **Trace – Trace Cursor**.

To move the trace cursor, do one of the following:
Move the mouse cursor. The trace cursor follows it. There is a small zone on both sides of the trace: the trace cursor moves only if the mouse cursor is within this zone. The following figure shows this.

Hold down the Control key while you press the Right or Left arrow key on your keyboard. The trace cursor moves to the next data point on the trace.

When there are multiple traces on your graph, the trace cursor

follows the mouse cursor across traces if the autoTraceSelect variable is set to true (the default value) in the .cdsenv file.

stays on the selected trace if the autoTraceSelect variable is set to false in the .cdsenv file.

You can select one of the following options from the Marker menu to control the behavior of the tracking cursor:

Graph – Snap Off

Tracking cursor follows the trace smoothly going between the data points.

Graph – Snap-to-Data

Tracking cursor snaps to data points on the trace.

Graph – Snap-to-Peak
Tracking cursor snaps to the trace peaks.

Using a Vertical Cursor

A vertical cursor is a moveable vertical grid line on your graph. The X and Y coordinates of its intersection with each trace on the graph are displayed next to the signal names in the legend area. You can drag the red triangle at the end of the line to move the cursor. As you move the cursor, the coordinates of the intersection points are dynamically updated.

To enable the vertical cursor,

➤ Choose Trace – Vert Cursor.

A vertical cursor appears on your graph.

For digital traces, the time, level, and strength appear next to the trace.
Using a Horizontal Cursor

A horizontal cursor is a moveable horizontal grid line on your graph. The coordinates of the first intersection point with each trace are displayed next to the signal name in the legend area. You can drag the red triangle at the end of the line to move the cursor.

To enable the horizontal cursor,

➤ Choose Trace – Horizontal Cursor.

A horizontal cursor appears on your graph.
Using a Delta Cursor

A delta cursor consists of two movable cursors marked by a red and blue triangle at the point at which they intersect the trace. You can click and drag these triangles to move the cursors. Each cursor consists of two lines – one intersecting the X-axis, and the other intersecting the Y-axis. The X and Y coordinates of the cursors are displayed in the lower left corner of the Graph Window and the $\Delta x$, $\Delta y$, and slope are displayed in the lower right corner.

To enable the delta cursor,

➤ Choose Trace – Delta Cursor.

A delta cursor appears on your graph.
When there are multiple traces on your graph, the delta cursor

- appears on the trace closest to the mouse cursor if the `autoTraceSelect` variable is set to true (the default value) in the `.cdsenv` file.

- appears on the selected trace (or the trace last added to the graph) if the `autoTraceSelect` variable is set to false in the `.cdsenv` file.
Working with Labels

The following figure displays the labels in a graph. You can edit all these labels.

The following sections describe how you can add and edit independent labels.

Adding a Label

This section describes how to add a label through the Graph menu or the Label field.

Adding a Label through the Graph Menu

1. In the Graph Window, choose Graph – Label – Edit.

   The Label Attributes dialog box appears.

   **Note:** If the Attributes dialog box is already open, you can choose Object – Label to display the Label Attributes dialog box.

2. Enter the desired information and click Add.
The label begins moving with your cursor.

3. Click where you want to place the label in the Graph Window.

For more information on the fields in the Label Attributes dialog box, see “Editing a Label” on page 90.

Adding a Label through the Label Field

1. If there is more than one graph in the Graph Window, select a graph by clicking on the index box in the top right corner of the Graph Window.

2. Type text for the label in the Label field.

   This field is available only if the graph window is selected.

3. Click  

   The label begins moving with your cursor.

4. Click where you want to place the label in the Graph Window.

Editing a Label

This section describes how to edit a label through the Graph menu or the Label field.

Editing a label through the Graph Menu

1. Double-click on a label.
The Label Attributes dialog box appears.

2. In the *String* field, enter the label text. WaveScan provides a number of variables (see table below) that can be embedded into the label text string. These variables are evaluated and the resultant values are inserted into the string when the marker is placed on the graph. This allows labels to easily reference the marker coordinates, trace slope, trace name, etc. or the result of an scalar expression.

The following table describes the available variables.
This text is displayed in the *Label* field on the Graph Window when you select the label.

3. In the *Expression* field, type an expression or click [enter] and select the memory (in ADE WaveScan or standalone SKILL WaveScan) or variable (in standalone MDL WaveScan) for the expression you want to assign to the label. For more information on creating memories, see *Working with Memories* on page 151. For more information on creating variables, see *Working with Variables* on page 156.

4. In the *Signif Digits* field, type the number of significant digits to be displayed on your marker label.

   Select *Default* to display the default number of significant digits. This disables the *Signif Digits* field.

5. In the *Font* fields, specify the font name, style, and size for the label.

6. In the *Foreground* field, specify the label color.

7. In the *Direction* field, specify the label direction.

   Label directions are depicted in the figure below. The end of the label touches the direction you specify.

```
   East  West  North  South
   ABC   ABC   ABC    ABC
```

8. Click *OK*.
Editing a Label through the Label Field

1. Select the label you want to edit.
   The label text appears in the Label field.
2. Type the new label text.
3. Click on the left of the Label field.
   WaveScan updates the selected label.

Moving a Label

Using this procedure, you can move independent labels; you cannot move a label attached to a marker.

To move a label,

1. Select the label that you want to move.
2. Do one of the following:
   - Drag the label to a new position and release the left mouse button.
   - Choose Edit – Move and click where you want to place the label.
   WaveScan moves the label moves to the new position.

Working with Markers

A marker attaches a description to a point on the trace. The default label for a marker displays the X and Y coordinates of its intersection with the trace. You can associate an expression with a marker label. For more information, see step 3 in “Editing a Marker” on page 100. The expression is evaluated when you place the marker on the graph and updated when you select File – Reload in the Results Browser. For more information, see Reloading Data on page 43.

Creating Markers

You can create trace, vertical, horizontal, and delta markers in the Graph Window.

Creating a Trace, Vertical, or Horizontal Marker by Specifying Coordinates

To create a marker by specifying coordinates,
1. Choose *Marker – Create*.

The Marker Attributes dialog box opens.

**Note:** If the Attributes dialog box is already open, choose *Create – Marker* to create a new marker.

2. In the *Label* field, do one of the following:

- Enter the text string for the marker. You may use one of the variables mentioned in the table below. These variables are evaluated and inserted into the string when you place or edit a marker. This allows labels to reference the marker coordinates, trace slope, trace name, etc. or the result of an scalar expression.

The following table describes the available variables.
Select Default to display the default label (for example, %X, %Y for a marker in the XY mode). Selecting Default disables the Label field.

3. In the X/Y fields, do one of the following:
   - Specify the X and Y coordinates of the point at which you want to place the marker. This helps you place the marker at an exact intersection point with the trace. For example, if you want to place a marker at 10ns (X-axis value): type 10 in the X coordinate field, and leave the Y coordinate field blank. WaveScan finds the corresponding Y coordinate and places the marker.
   - Select Use Cursor to place the marker at the cursor location. Selecting Use Cursor disables the X/Y fields.

4. Select the Trace checkbox to attach the marker to the closest interpolated point on the trace.

5. Select the Data Points checkbox to attach the marker to the closest data point on the trace.

6. In the Type field, specify whether you want to create a trace, vertical, or horizontal marker.

7. Select the Arrow checkbox to connect the label to the trace.

8. In the Font fields, specify the font name, style, and size for the label.

9. In the Foreground field, select the foreground color for the label.

10. Click Add.
The following figure shows a trace, vertical, and horizontal marker.

Creating a Trace, Vertical, or Horizontal Marker with the Mouse

To add a marker with the mouse, do one of the following:

To add a trace marker through the Marker menu, do one of the following:

- Choose Marker – Place – Trace Marker, Marker – Place – Vert Marker, or Marker – Place – Horiz Marker and click in the graph.

- Point the system cursor at the desired position on the graph and type m (for trace marker), v (for vertical marker), or h (for horizontal marker).

WaveScan places a marker on the trace nearest to the point you clicked. If there are multiple traces on your graph, WaveScan places the trace marker

- on the trace nearest to the point you clicked if the autoTraceSelect variable is set to true (the default value) in the .cdsenv file.

- on the trace last selected (or the trace last added to the graph) if the autoTraceSelect variable is set to false in the .cdsenv file.
Creating a Delta Marker

Delta markers are used to mark the difference between two points on a graph. They are used in association with trace markers; in order to place a delta marker you must have just placed a trace marker or have one selected. Delta markers may be moved or deleted independently of their trace markers.

You can move either end of a delta marker – the X and Y coordinates are updated correspondingly. You can use delta markers to measure delays or use them with the min/max functions to measure peak to peak values.

To place a delta marker,

1. Do one of the following:
   - Choose Marker – Place Marker – Trace Marker to place a marker.
   - Select an existing marker.

2. Choose Marker – Add Delta.

3. Do one of the following:
   - Click at a point on the trace.
   - Point the system cursor at the desired position on the graph and type \texttt{m} for a marker. Then point to the second location and type \texttt{a} to add a second marker.
A delta marker of XY type appears on the graph. The marker label displays the $\Delta x$ and $\Delta y$ values.

When there are multiple traces on your graph, the delta marker

- can show the difference between points on different traces if the `autoTraceSelect` variable is set to true (the default value) in the `.cdsenv` file.
- can show the difference between points on the same trace if the `autoTraceSelect` variable is set to false in the `.cdsenv` file.

**Changing the Marker Display Mode**

You can display all markers (trace, vertical, horizontal, and delta marker) in the XY, X, or Y mode.

To change the display mode for a marker,

1. Select a marker.
2. Choose *Marker – Display Type* and select one of the following:
The following figure displays a delta marker in each of the three modes.

<table>
<thead>
<tr>
<th>Selection</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>XY Mode</strong></td>
<td>If the selected marker is a trace, vertical, or horizontal marker, the label displays the X and Y coordinates of the point at which the marker intersects with the trace. If the selected marker is a delta marker, the label displays the Δx and Δy values.</td>
</tr>
<tr>
<td><strong>X Mode</strong></td>
<td>If the selected marker is a trace, vertical, or horizontal marker, the label displays the X coordinate of the point at which the marker intersects with the trace. If the selected marker is a delta marker, the label displays the Δx value.</td>
</tr>
<tr>
<td><strong>Y Mode</strong></td>
<td>If the selected marker is a trace, vertical, or horizontal marker, the label displays the Y coordinate of the point at which the marker intersects with the trace. If the selected marker is a delta marker, the label displays the Δy value.</td>
</tr>
</tbody>
</table>
Editing a Marker

The default marker attributes are controlled by the values assigned to variables in the .cdsenv file. For more information, see Appendix A, “WaveScan Variables.”

To edit a marker,

1. Do one of the following:
   - Double-click on a marker.
   - Select the marker you want to edit by clicking the + on the trace where the marker exists, or one of the two lines connecting the + symbol to the marker text. Then choose Marker – Edit.

Click on this to select the marker
The Marker Attributes dialog box appears.

2. Edit the required fields. For more information on the fields, see Creating a Trace, Vertical, or Horizontal Marker by Specifying Coordinates on page 93.

3. Click OK.

Moving a Marker

This section describes how to move a marker. To move independent labels, see “Moving a Label” on page 93.

To move a marker,

1. Select the marker that you want to move. For more information on selecting a marker, see Step 1 of “Editing a Marker” on page 100.

2. Do one of the following:

   - Drag the marker to a new position and release the left mouse button.
   - Choose Edit – Move and click where you want to place the marker.

WaveScan moves the marker to the new position.

When there are multiple traces on your graph, you can move a marker
• across traces if the autoTraceSelect variable is set to true (the default value) in the .cdsenv file.

• only along the trace on which the marker exists if the autoTraceSelect variable is set to false in the .cdsenv file.

Working With Buses

You can create a bus from digital signals or other buses; and you can expand a bus to view the component signals.

Creating a Bus

1. In the Graph Window, select the digital traces from which you want to create a bus.

   Select traces in order of significance (top to bottom) as shown in the figure below (most significant bit to least significant bit).

2. Choose Trace – Bus – Create.

   The selected traces form a bus. The bus replaces the selected signals.

Expanding a Bus

To expand a bus:

1. Select the bus you want to expand.

2. Choose Trace – Bus – Expand.

   WaveScan displays the individual traces in the selected bus.
Saving and Loading Traces

Saving a Trace

Saving a trace saves the data in ASCII format. You can bring existing ASCII data into WaveScan to compare it with simulated data, or use a saved trace to re-create an old graph.

To save a trace,

1. Select a trace and choose Trace – Save.

   The Save dialog box appears.

   ![Save dialog box](image)

2. In the Files of type field, specify whether you want to display all files or files with the .grf suffix. The suffix for graph files is specified by the filesuffix variable in the .cdsenv file.

3. In the Look in field, select the directory where you want to save the trace.

4. Do one of the following:

   - If you want to overwrite an existing trace file, select that trace file from the listbox below the Look in field.
   - In the File name field, type a name for the trace file you want to save.

5. Click Save.
WaveScan saves the trace.

**Loading a Trace**

You can load a saved trace in an existing graph.

To load a trace,

1. In the Graph Window, choose *Trace – Load*.

   The Open dialog box appears.

2. In the *Look in* field, select the directory from which you want to load the trace.

3. Do one of the following:
   - Select the trace file you want to load from the listbox below the *Look in* field.
   - In the *File name* field, type the name of the trace file you want to load.

4. Click *Open*.

   WaveScan adds the trace to the selected graph.
Freezing Graphs

You can freeze a graph to ensure that it does not change even when the simulation results in the data directory are reloaded.

To freeze a graph,

➢ In the Graph Window, choose Graph – Freeze On.

WaveScan freezes the graph.

Reloading Graphs

You can update the data in all the graphs in a Graph Window.

To reload a graph,

➢ In the Graph Window, choose File – Reload.

WaveScan updates the data.

Printing Graphs

You can print a selected graph or all the graphs in the selected window.

To print a graph,

1. In the Graph Window, choose File – Print.
The Print dialog box appears.

2. In the Print Service area, do one of the following:

- Select Print To File to print your graph to a file.
- In the Name field, type the name of the printer you want to use or choose the printer from the cyclic field.

WaveScan compiles the list of printers in the cyclic field by appending the printers specified in each of the following bullets.

- Printer specified through the `printer` variable in the `.cdsenv` file.
- Printer specified through the shell `PRINTER` or `LPDEST` environment variable.
- Printers listed in the `.printers` file (on Solaris machines only).
  
  For more information, type `man printers` in a Solaris terminal window.

- Printers displayed by the `lpc status` command (on Linux systems) or by the `lpstat -a` command (on all other systems).
  
  For more information on these commands, see your operating system manual.

- Postscript printers specified in the `.cdsplotinit` file.
For more information on the `.cdsplotinit` file, see Appendix A in the *Plotter Configuration User Guide*.

- Printer specified in the previous WaveScan session.

If you change the default printer name, WaveScan saves this new value as the default value.

The *Status*, *Type*, and *Info* fields and the Properties button are disabled.

3. In the *Print Subwindows* section, select one of the following:
   - *All Subwindows* to print all subwindows within the Graph Window on a single page.
   - *Selected* to print only the selected subwindows.
   - *Print All Strips* to print all traces (including traces scrolled off the screen).

   If you have multiple graph windows, there is no way to print all the graphs in all the windows. You need to choose *Print* in each window.

4. In the *Number of copies* field, enter the number of copies you want to print. The *Collate* field is disabled.

5. Click the Page Setup tab.

6. In the *Size* field, select the paper size. WaveScan takes the default value from the locale that the `LC_ALL` environment variable is set to on your machine. If you change the paper size, WaveScan saves this new value as the default value.
7. In the *Source* field, loading paper automatically is the only option currently available.

8. In the *Orientation* section, specify whether you want to print your graph in portrait or landscape mode.

9. In the *Scale Factor* field, specify the percentage by which you want to scale up or scale down the printed graph.

10. In the *Margin* section, type the *left*, *right*, *top*, and *bottom* margins for your printout.

11. Click the Appearance tab.

12. In the *Color Appearance* section, specify whether you want your graph to be printed in grayscale or color.

13. In the *Quality* section, *Normal* is the only option currently available.

14. In the *Sides* section, *One Side* is the only option currently available.

15. Select *Banner Page* if you want to print a banner page. Selecting the Banner Page option enables the Job Name and User Name fields. Type the *Job Name* and *User Name* you want to display on the banner page.
16. Click the Annotations tab.

17. Choose one of the following in the Trace Legends area:
   - Normal (Top) to print the trace legends above the analog traces and below the digital traces.
   - Extended (Bottom) to print the trace legends including information displayed in the trace pop-ups (signal name, dataset, and data directory) at the bottom of the page.
   - None to print the graph without any trace legends.

18. Select Auto style traces if you want each trace on the graph to be distinguished by a unique symbol and line style.

19. The Labels area is currently disabled.

20. Click Print.
   - If you specified a printer in Step 2, WaveScan prints the graph.
   - If you selected Print to File in Step 2, the Print To File dialog box appears. Type a file name for the graph and click OK.

### Saving Graphs

WaveScan allows you to save graphs in an XML or postScript format, or as an image file.
Saving a Graph in XML Format

By saving a graph in XML format, you can restore a graph from a previous session. A saved graph saves the following information:

- The location to the data (data directory, data set, and trace name); not the actual data. Therefore, if your simulation data changes between sessions, the graph reflects those changes.
- Most graph attributes such as grids, background and foreground color, labels, and markers.

To save a graph in the XML format,


   The Save Graph dialog box appears.

2. In the Files of type field, specify whether you want to display all files or files with the .grf suffix. The suffix for graph files is specified by the filesuffix variable in the .cdsenv file.

3. In the Look in field, select the directory where you want to save the graph.

4. Do one of the following:
   - If you want to overwrite an existing graph file, select that graph file from the listbox below the Look in field.
   - In the File name field, type a name for the graph file you want to save.

5. Click Save.
WaveScan saves the graph with the file suffix specified in the .cdsenv file.

**Saving a Graph in the PostScript Format**

You can save a graph in the postScript format. The postScript format is fully scalable and is therefore used in most UNIX printers.

To save a graph in the postScript format,

1. In the Graph Window, choose *File – Print*.
   
   The Print dialog box appears.

2. Select *Print To File*.
   
   The Print To File dialog box appears.

3. Type a file name for the graph and click *OK*.
   
   WaveScan saves your graph.

**Saving a Graph as an Image File**

You can save your graph as an image file. The image formats can be displayed in a wide variety of applications and environments. All file formats are saved in lossless (no compression) mode. While all are functionally equivalent binary storage formats, the size of a typical file may vary greatly according to the format chosen. For example, a simple graph file saved in PNG format is typically less than 100 KB while the same file in the TIFF or BMP formats may exceed 1MB.

You cannot open a graph saved as an image file in WaveScan.

To save a graph as an image file,

1. Choose *File – Save as Image*. 
The Save Image dialog box appears.

2. Select the image format you want to save the graph in.

   If you select Auto (based on file extension), WaveScan sets the output format based on the extension of the file name you type in. For example, if you type output.png, WaveScan saves the file in png format. The extensions that are supported are .png, .bmp, and .tiff (or .tif). On the other hand, if you select PNG (Best compressed) and type output.tiff, WaveScan saves the file as output.png.

3. In the File field, specify the name of the image file. The default name is snapshot.png. Click Browse to specify the directory where you want to save the file.

4. Click Save to save the file.

   WaveScan saves the graph file in the specified format.

**Opening Saved Graphs**

You cannot open graphs saved as image files in WaveScan.

To open a saved graph,

1. In the Results Browser window, choose File – Open Graph – Open Graph as Plot.
The Open Graph dialog box appears.

2. In the **Look In** field, select the directory from which you want to open a graph.

3. In the **Files of type** field, specify whether you want to display all files or files with the `.grf` suffix. The suffix for graph files is specified by the `filesuffix` variable in the `.cdsenv` file.

4. Do one of the following:
   - Select the graph file you want to open from the listbox below the **Look in** field.
   - In the **File name** field, type the name of the file you want to open.

5. Click **Open**.

WaveScan opens the specified graph file.

### Creating Graph Templates

When you create a new graph, WaveScan refers to the `.cdsenv` file to determine graph attributes such as the foreground and background colors, whether grids are on or off, font size etc.

You can open a saved graph as a template, or set the selected graph as the template and use this template as a starting point for new graphs. The new graphs would then have the same attributes as the template. The specified template is not applied to the current graph.

To set a saved graph as the template,
In the Results Browser window, choose File – Open Graph – Open Graph as Template.

Any new graph that you create uses the specified graph as the template.

To set the selected graph as the template,

In the Graph Window, select Graph – Template – Set Current.

Any new graph that you create uses the selected graph as the template. However, you can use this graph as the template only in the current session. To use it in subsequent sessions, save the graph and open it as a template.

To go back to the default template,

In the Graph Window, select Graph – Template – Set Default.

Any new graph that you create will have the attributes specified in the .cdsenv file. The attributes of the current graph do not change to reflect the default attributes.

For information about the default settings, see Appendix A, “WaveScan Variables.”
Menu Bar and Toolbar

For information on the accelerator keys you can use in this window, see Appendix C, “Accelerator Keys.”

Menu Bar

For a description of the menu commands, see the table below. All menu options work on the selected graph window or subwindow.
<table>
<thead>
<tr>
<th>Item</th>
<th>Usage and Description</th>
<th>For More Information, See</th>
</tr>
</thead>
<tbody>
<tr>
<td>File</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Open</td>
<td>Opens the Open Graph dialog box so that you can open a graph.</td>
<td>“Opening Saved Graphs” on page 112.</td>
</tr>
<tr>
<td>Save</td>
<td>Opens the Save Graph dialog box so that you can save a graph in XML format.</td>
<td>“Saving a Graph in XML Format” on page 110.</td>
</tr>
<tr>
<td>Save as Image</td>
<td>Opens the Save Image dialog box so that you can save a graph as an image file.</td>
<td>Saving a Graph as an Image File on page 111.</td>
</tr>
<tr>
<td>Reload</td>
<td>Updates data for all traces within all graphs in the Graph Window.</td>
<td></td>
</tr>
<tr>
<td>Print</td>
<td>Prints the graph.</td>
<td>“Printing Graphs” on page 105.</td>
</tr>
<tr>
<td>Save Session</td>
<td>Saves the session.</td>
<td>Saving a Session on page 22.</td>
</tr>
<tr>
<td>Close</td>
<td>Closes the Graph Window. If this is the only WaveScan window open, Close exits WaveScan.</td>
<td></td>
</tr>
<tr>
<td>Exit</td>
<td>Closes all windows and exits WaveScan in MDL WaveScan.</td>
<td>Exiting WaveScan on page 21.</td>
</tr>
<tr>
<td>Edit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Move</td>
<td>Moves the selected label or marker.</td>
<td>“Moving a Marker” on page 101 and “Moving a Label” on page 93.</td>
</tr>
<tr>
<td>Swap</td>
<td>Swaps two traces, dependent axes, or graphs.</td>
<td>“Swapping Objects” on page 59.</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes the selected label, marker, legend, trace, or graph.</td>
<td>“Deleting Objects” on page 59.</td>
</tr>
<tr>
<td>Hide</td>
<td>Hides the selected label, marker, legend, trace, axis, or graph.</td>
<td>“Hiding Objects” on page 58.</td>
</tr>
</tbody>
</table>
## WaveScan User Guide
### Working with Graphs

<table>
<thead>
<tr>
<th>Item</th>
<th>Usage and Description</th>
<th>For More Information, See</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reveal</td>
<td>Reveals the hidden selected label, marker, legend, trace, or graph axis.</td>
<td>“Revealing Objects” on page 58.</td>
</tr>
<tr>
<td>Undo</td>
<td>Undoes the most recent action.</td>
<td>“Undoing Actions” on page 59.</td>
</tr>
</tbody>
</table>

### Graph

<table>
<thead>
<tr>
<th>Item</th>
<th>Usage and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grids On</td>
<td>Displays and hides all grids (major and minor).</td>
</tr>
<tr>
<td>Layout</td>
<td>Controls the graph layout.</td>
</tr>
<tr>
<td>Auto</td>
<td>Displays subwindows according to the overall size of the Graph Window.</td>
</tr>
<tr>
<td>Vertical</td>
<td>Displays subwindows one below the other.</td>
</tr>
<tr>
<td>Horizontal</td>
<td>Displays subwindows side by side.</td>
</tr>
<tr>
<td>Card</td>
<td>Stacks the subwindows in the Graph Window.</td>
</tr>
<tr>
<td>Display Type</td>
<td>Controls the trace display.</td>
</tr>
<tr>
<td>Rectangular</td>
<td>Displays the trace as a rectangular graph.</td>
</tr>
<tr>
<td>Histogram</td>
<td>Displays the trace as a histogram.</td>
</tr>
<tr>
<td>RealVsImag</td>
<td>Plots the real part of the data against the imaginary part. This selection is available only for AC data.</td>
</tr>
<tr>
<td>Polar</td>
<td>Displays the trace as a polar graph.</td>
</tr>
<tr>
<td>Impedance</td>
<td>Displays the trace as an impedance graph.</td>
</tr>
<tr>
<td>Admittance</td>
<td>Displays the trace as an admittance graph.</td>
</tr>
</tbody>
</table>

### Font

<table>
<thead>
<tr>
<th>Item</th>
<th>Usage and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>Sets font size of graph to small. Affects the title, subtitle, and axes labels.</td>
</tr>
<tr>
<td>Medium</td>
<td>Sets font size of graph to medium. Affects the title, subtitle, and axes labels.</td>
</tr>
</tbody>
</table>
### Large
Sets font size of graph to large. Affects the title, subtitle, and axes labels.

### Label
- **Create**
  Opens the Label Attributes dialog box so that you can create a label.
  - “Adding a Label through the Graph Menu” on page 89.
- **Edit**
  Edits the selected label.
  - “Editing a Label” on page 90.

### Freeze On
Blocks the graph from data updates.
- “Freezing Graphs” on page 105.

### Show Toolbar
Displays the icons.
- “Toolbar” on page 125.

### Snap Off
Cursor follows system cursor.

### Snap-to-Data
Marker snaps to data points on the trace.

### Snap-to-Peaks
Marker snaps to trace peaks.

### Color Schemes
Controls the color scheme of the graph.
- **Default**
  Displays the graph on a white background.
- **Grey**
  Displays the graph on a grey background.
- **Black**
  Displays the graph on a black background.
- “Creating Graph Templates” on page 113.

### Template
- **Set Default**
  Resets the graph template to reflect the .cdsenv file.
- **Set Current**
  Sets the current graph attributes as the default.
- **Load**
  Opens the specified graph as the template.
- **Edit**
  Opens the Graph Attributes dialog box.
  - “Editing Graph Attributes” on page 63.

### Axis
## Working with Graphs

<table>
<thead>
<tr>
<th>Item</th>
<th>Usage and Description</th>
<th>For More Information, See</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Major Grids On</strong></td>
<td>Displays major grids for the X or Y axis, depending on the selected axis. This option is available only if an axis is selected.</td>
<td></td>
</tr>
<tr>
<td><strong>Minor Grids On</strong></td>
<td>Displays minor grids for the X or Y axis, depending on the selected axis. This option is available only if an axis is selected.</td>
<td></td>
</tr>
<tr>
<td><strong>Log</strong></td>
<td>Displays the logarithmic scale for the selected axis. This option is available only if an axis is selected.</td>
<td></td>
</tr>
<tr>
<td><strong>Strip</strong></td>
<td>Displays each trace in an individual strip.</td>
<td></td>
</tr>
<tr>
<td><strong>Edit</strong></td>
<td>Opens the Axis Attributes dialog box. This option is available only if an axis is selected.</td>
<td>“Editing Axis Attributes” on page 75.</td>
</tr>
<tr>
<td>Item</td>
<td>Usage and Description</td>
<td>For More Information, See</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>Trace</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Trace*
## WaveScan User Guide
### Working with Graphs

<table>
<thead>
<tr>
<th>Item</th>
<th>Usage and Description</th>
<th>For More Information, See</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Symbols On</strong></td>
<td>Displays symbols on the individual data points on the trace. This option is available only if a trace is selected.</td>
<td>“Displaying Symbols” on page 68</td>
</tr>
<tr>
<td><strong>Assign to Axis</strong></td>
<td>Assigns the selected trace to a new Y axis. This option is available only if a trace is selected.</td>
<td>“Assigning a Trace to an Axis” on page 69</td>
</tr>
<tr>
<td><strong>New Graph</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Copy New Window</td>
<td>Copies the selected trace to a new graph window.</td>
<td></td>
</tr>
<tr>
<td>Move New Window</td>
<td>Moves the selected trace to a new graph window.</td>
<td></td>
</tr>
<tr>
<td>Copy New SubWindow</td>
<td>Copies the selected trace to a new subwindow.</td>
<td></td>
</tr>
<tr>
<td>Move New SubWindow</td>
<td>Moves the selected trace to a new subwindow.</td>
<td></td>
</tr>
<tr>
<td><strong>Bus</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create</td>
<td>Creates a bus from selected digital traces.</td>
<td>“Creating a Bus” on page 102</td>
</tr>
<tr>
<td>Expand</td>
<td>Expands a bus to its component signals. This option is available only if a bus of digital signals is selected.</td>
<td>“Expanding a Bus” on page 102</td>
</tr>
<tr>
<td>Trace Cursor</td>
<td>Enables and disables the tracking cursor.</td>
<td>“Using a Trace Cursor” on page 82.</td>
</tr>
<tr>
<td>Vert Cursor</td>
<td>Turns the vertical cursor on and off.</td>
<td>“Using a Vertical Cursor” on page 85.</td>
</tr>
<tr>
<td>Horiz Cursor</td>
<td>Turns the horizontal cursor on and off.</td>
<td>“Using a Horizontal Cursor” on page 86.</td>
</tr>
<tr>
<td>Delta Cursor</td>
<td>Turns the delta cursor on and off.</td>
<td>Using a Delta Cursor on page 87.</td>
</tr>
<tr>
<td>Cut</td>
<td>Cuts the selected trace. This option is available only if a trace is selected.</td>
<td></td>
</tr>
<tr>
<td>Copy</td>
<td>Copies the selected trace. This option is available only if a trace is selected.</td>
<td></td>
</tr>
<tr>
<td>Item</td>
<td>Usage and Description</td>
<td>For More Information, See</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------------------------------------------------------------------</td>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>Load</td>
<td>Opens the Load dialog box so you can add a trace to the selected graph.</td>
<td>“Loading a Trace” on page 104.</td>
</tr>
<tr>
<td>Save</td>
<td>Opens the Save dialog box so you can save the trace in ASCII format.</td>
<td>“Saving a Trace” on page 103.</td>
</tr>
<tr>
<td>Edit</td>
<td>Opens the Trace Attributes dialog box. This option is available only if a trace is selected.</td>
<td>“Editing Trace Attributes” on page 71.</td>
</tr>
<tr>
<td>Select All</td>
<td>Selects all the traces in the Graph Window or the selected subwindow.</td>
<td></td>
</tr>
<tr>
<td>Marker</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Place</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trace Marker</td>
<td>Places a trace marker on the trace.</td>
<td>Creating a Trace, Vertical, or Horizontal Marker with the Mouse on page 96.</td>
</tr>
<tr>
<td>Vert Marker</td>
<td>Places a vertical marker on the graph.</td>
<td>Creating a Trace, Vertical, or Horizontal Marker with the Mouse on page 96.</td>
</tr>
<tr>
<td>Horiz Marker</td>
<td>Places a horizontal marker on the graph.</td>
<td>Creating a Trace, Vertical, or Horizontal Marker with the Mouse on page 96.</td>
</tr>
<tr>
<td>Add Delta</td>
<td>Adds a marker displaying the distance between two points.</td>
<td>“Creating a Delta Marker” on page 97.</td>
</tr>
<tr>
<td>Display Type</td>
<td></td>
<td>“Changing the Marker Display Mode” on page 98.</td>
</tr>
<tr>
<td>Item</td>
<td>Usage and Description</td>
<td>For More Information, See</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>XY Delta</td>
<td>Marker label displays the $\Delta x$ and $\Delta y$ values.</td>
<td></td>
</tr>
<tr>
<td>X Delta</td>
<td>Marker label displays the $\Delta x$ value.</td>
<td></td>
</tr>
<tr>
<td>Y Delta</td>
<td>Marker label displays the $\Delta y$ value.</td>
<td></td>
</tr>
<tr>
<td>Attach to Trace</td>
<td>Marker is attached to the selected trace.</td>
<td></td>
</tr>
<tr>
<td>Find Max</td>
<td>Moves marker to the maximum point on the selected trace.</td>
<td></td>
</tr>
<tr>
<td>Find Min</td>
<td>Moves marker to the minimum point on the selected trace.</td>
<td></td>
</tr>
<tr>
<td>Create</td>
<td>Opens the Marker Attributes dialog box so that you can create a new marker.</td>
<td></td>
</tr>
<tr>
<td>Edit</td>
<td>Opens the Marker Attributes dialog box so that you can edit a marker. This option is available only when a marker is selected.</td>
<td>“Editing a Marker” on page 100.</td>
</tr>
<tr>
<td>Select All</td>
<td>Selects all markers in the Graph Window or the selected subwindow.</td>
<td></td>
</tr>
<tr>
<td>Zoom</td>
<td></td>
<td>“Zooming a Graph” on page 59.</td>
</tr>
</tbody>
</table>
WaveScan User Guide
Working with Graphs

<table>
<thead>
<tr>
<th>Item</th>
<th>Usage and Description</th>
<th>For More Information, See</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zoom</td>
<td>Zooms the graph.</td>
<td></td>
</tr>
<tr>
<td>X-Zoom</td>
<td>Zooms the graph along the X axis.</td>
<td></td>
</tr>
<tr>
<td>Y-Zoom</td>
<td>Zooms the graph along the Y axis.</td>
<td></td>
</tr>
<tr>
<td>Unzoom</td>
<td>Incrementally reverses a zoom action. This is especially useful when you zoom a graph multiple times.</td>
<td></td>
</tr>
<tr>
<td>Fit</td>
<td>Returns the graph to the original size.</td>
<td></td>
</tr>
<tr>
<td>Zoom In</td>
<td>Zooms into the graph.</td>
<td></td>
</tr>
<tr>
<td>Zoom Out</td>
<td>Zooms out of the graph.</td>
<td></td>
</tr>
<tr>
<td>Pan</td>
<td></td>
<td>“Panning a Graph” on page 62.</td>
</tr>
<tr>
<td>Pan Right</td>
<td>Displays the section of the graph to the right of the currently visible area.</td>
<td></td>
</tr>
<tr>
<td>Pan Left</td>
<td>Displays the section of the graph to the left of the currently visible area.</td>
<td></td>
</tr>
<tr>
<td>Pan Up</td>
<td>Displays the section of the graph above the currently visible area.</td>
<td></td>
</tr>
<tr>
<td>Pan Down</td>
<td>Displays the section of the graph below the currently visible area.</td>
<td></td>
</tr>
<tr>
<td>Tools</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Browser</td>
<td>Opens the Results Browser window.</td>
<td></td>
</tr>
<tr>
<td>Calculator</td>
<td>Opens the Calculator.</td>
<td>Chapter 4, “Working with the Calculator.”</td>
</tr>
<tr>
<td>Help</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Help</td>
<td>Displays this document, the WaveScan User Guide.</td>
<td>Appendix C, “Accelerator Keys.”</td>
</tr>
<tr>
<td>Accelerator Keys</td>
<td>Displays the accelerator keys for menu options.</td>
<td></td>
</tr>
</tbody>
</table>
Toolbar

For a description of the toolbar, see the table below.
## WaveScan User Guide

### Working with Graphs

<table>
<thead>
<tr>
<th>Item</th>
<th>Usage and Description</th>
<th>For More Information, See</th>
</tr>
</thead>
<tbody>
<tr>
<td>Print</td>
<td>Prints the graph.</td>
<td>Printing Graphs on page 105.</td>
</tr>
<tr>
<td>Undo</td>
<td>Undoes the most recent action.</td>
<td>“Undoing Actions” on page 59.</td>
</tr>
<tr>
<td>Grids On/Off</td>
<td>Displays or hides grids.</td>
<td></td>
</tr>
<tr>
<td>Strip Chart Mode</td>
<td>Displays each trace in a separate subwindow.</td>
<td>“Handling Graph Objects” on page 58.</td>
</tr>
<tr>
<td>Stack Graphs</td>
<td>Stacks the subwindows.</td>
<td>“Changing the Graph Layout” on page 55.</td>
</tr>
<tr>
<td>New SubWindow</td>
<td>In SKILL WaveScan, creates a new subwindow. If a trace is selected, it is moved to the</td>
<td></td>
</tr>
<tr>
<td></td>
<td>new subwindow. In standalone WaveScan, moves the selected trace to a new subwindow.</td>
<td></td>
</tr>
<tr>
<td>New Window</td>
<td>In SKILL WaveScan, creates a new graph window. If a trace is selected, it is moved to</td>
<td></td>
</tr>
<tr>
<td></td>
<td>the new window. In standalone WaveScan, moves the selected trace to a new window.</td>
<td></td>
</tr>
<tr>
<td>Expand bus</td>
<td>Expands the selected bus of digital signals. This icon is available only if a bus is</td>
<td>Expanding a Bus on page 102.</td>
</tr>
<tr>
<td></td>
<td>selected.</td>
<td></td>
</tr>
<tr>
<td>Create bus</td>
<td>Creates a bus from selected digital traces. This icon is available only if more than</td>
<td>“Creating a Bus” on page 102.</td>
</tr>
<tr>
<td></td>
<td>one digital trace is selected.</td>
<td></td>
</tr>
</tbody>
</table>

## Label Area
WaveScan User Guide
Working with Graphs

<table>
<thead>
<tr>
<th>Item</th>
<th>Usage and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place New Label</td>
<td>Attaches the text entered in the Label field to the cursor. Click where you want to place the label. For more information, see “Adding a Label through the Label Field” on page 90.</td>
</tr>
<tr>
<td>Label field</td>
<td>Displays the text for the selected label.</td>
</tr>
</tbody>
</table>

**Status Bar**

Displays the following:

- Warnings and error messages.
- Prompts for further action. For example, when you type text for a label in the Label field, and click , the status bar displays the following message:

  ![Left mouse click to place label.]

- Static information. For example, if you select a trace, the status bar displays the name of the trace.
Working with the Calculator

The WaveScan calculator helps you perform computations on the data generated by your simulator.

This chapter discusses the following

- “About the Calculator” on page 130
  - “Opening the Calculator” on page 130
  - “Features of the Calculator” on page 132
- “Selecting Signals” on page 137
- “Building Expressions” on page 139
- “Using the Calculator Functions” on page 145
- “Entering Constants” on page 149
- “Printing Expressions” on page 149
- “Working with Memories” on page 151
- “Working with Variables” on page 156
- “Menu Bar and Toolbar” on page 161
About the Calculator

This section describes how to open the calculator from the Virtuoso Analog Design Environment (ADE) and in the standalone mode as well as its main features. In this chapter, SKILL mode refers to both the Virtuoso® Analog Design Environment (ADE) WaveScan and standalone SKILL calculators unless otherwise specified. MDL mode refers to the standalone MDL mode.

Opening the Calculator

This section talks about the various ways in which you can open the calculator.

Opening the Calculator from ADE

➤ From the CIW, choose Tools – Analog Environment – Calculator.

The Calculator window appears. For information about the parts of the window, click the cross references.

For information on menu and toolbar options, see “Menu Bar and Toolbar” on page 161.
Opening the Standalone SKILL Calculator

1. Start WaveScan by typing the following in a terminal window:
   
   ```
   wavescan -expr skill &
   ```
   
   The Results Browser, which is the main WaveScan window, appears.

2. (Optional) In the Results Browser window, select a signal.


   **Note:** You cannot perform simple arithmetic operations in the calculator if you open it without selecting a dataset or signal.

   **Note:** If the Results Browser window is closed, you can use the `drbBrowserFormCB` function (see Chapter 16 in the *Virtuoso Analog Design Environment SKILL Language Reference*) to open the Results Browser and the `calCalculatorFormCB` function (see Chapter 22 in the *Virtuoso Analog Design Environment SKILL Language Reference*) to open the calculator.

   The Calculator window appears. If you selected a signal, it appears in the buffer. For information about the parts of the window, click the cross references.

For information on menu and toolbar options, see “Menu Bar and Toolbar” on page 161.
Opening the Standalone MDL Calculator

1. Open the Results Browser by typing the following in a terminal window:
   
   ```bash
   wavescan [-expr mdl &]
   ```

2. (Optional) In the Results Browser window, select a signal.


   The Calculator window appears. If you selected a signal, it appears in the buffer. For information about the parts of the window, click the cross references.

For information on menu and toolbar options, see “Menu Bar and Toolbar” on page 161.

Features of the Calculator

This section describes the main features of the calculator.

Menu bar

For information, see “Menu Bar” on page 162.
DataSet Name

Displays the active data directory and dataset. You build expressions within the context of the data directory and dataset selected in the Results Browser.

In the ADE WaveScan calculator, the *DataSet Name* is determined by the current results directory in ADE. You can create expressions containing signals from multiple data directories.

In the standalone MDL calculator

- You can add a signal from a different dataset to the expression in the buffer – the dataset specifier is displayed with the signal name. The following expression illustrates this. The dataset specifier \((ac-ac)\) is displayed before the signal name \((net35)\).

\[ V(net10) + V(ac-ac->net35) \]

- You cannot add a signal from a different data directory to the expression in the buffer

To change the active data directory,

1. In the Results Browser window, select the data directory you want to open. For more information, see “Choosing a Data Directory” on page 25.
2. Choose *Tools – Calculator.*

WaveScan updates the dataset label and clears the calculator buffer.

Selection buttons

Family

The *Family* checkbox is available only when the *Select Mode* checkbox is selected.

When the *Family* checkbox is selected and you select a trace from a set of parametric leaf waveforms, WaveScan enters an expression for the entire family in the calculator buffer.

Select Mode

Enables you to select signals – any signal you select is displayed in the calculator buffer. You can select signals from the

- Results Browser
- Graph Window
### Schematic window (only in ADE WaveScan)

### Buffer
Displays expressions or results.

### Plot Destination
Displays the available destinations for plotting the result of your expression.
- **Append** – adds the result to the selected Graph Display window
- **Replace** – replaces the selected graph (or subwindow) with the result of your expression
- **New Subwin** – plots the result in a new subwindow
- **New Win** – plots the result in a new window

For more information, see “About the Signal Destination” on page 28.

### Static Panel
Displays numeric keys, operator keys, and special keys such as Clst, Enter, and Eval. The special keys displayed depend on whether the calculator is in the RPN or algebraic mode.

Most keys and functions are available in both modes. The following keys are unique to either the RPN or algebraic mode.

<table>
<thead>
<tr>
<th>Key</th>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clst</td>
<td>RPN</td>
<td>Clears the calculator stack.</td>
</tr>
<tr>
<td>Enter</td>
<td>RPN</td>
<td>Adds expression to the stack.</td>
</tr>
<tr>
<td>(</td>
<td>Algebraic</td>
<td>Adds opening parenthesis.</td>
</tr>
<tr>
<td>)</td>
<td>Algebraic</td>
<td>Adds closing parenthesis</td>
</tr>
</tbody>
</table>

### Function Panel
Displays the list of available functions. The functions displayed depend on the selected category. When you select a single-parameter function, it is displayed in the calculator buffer. When you select a multi-parameter function, the parameter panel for the function appears.
For example, when you select `slewRate`, the function panel displays the parameter list for `slewrate`. This is shown in the figure below (the figure displays the standalone calculator).

![Standalone Calculator](image)

Default values for each optional field are displayed. If you want to return to the default values for the parameters after over-writing them, click *Defaults*. Click >>> to display the rest of the parameters for `slewrate`. 
If you stretch the calculator vertically so that all the parameters are displayed at once, the <<< and >>> are grayed out. This is shown in the figure below.

After you fill in the parameter list for the selected function, you can do one of the following:

- Click OK to create the expression, add it to the calculator buffer, and re-display the list of built-in functions in the function panel.
- Click Cancel to cancel the function selection and re-display the list of built-in functions in the function panel.
- Click Apply to create the expression, add it to the buffer, and keep the parameter list for the same function open.

For a description of the functions in the calculator, see Appendix D, “Calculator Functions.”
Filter Area

You can use the filter in the calculator to display a subset of the available functions. For more information, see “Filtering Functions” on page 149.

Toolbar

For information, see “Toolbar” on page 163.

Schematic Expression Buttons

The buttons allow you to enter data to the buffer by selecting objects in the Schematic window.

The schematic expression buttons are available only in the ADE WaveScan calculator and are enabled only if the Select Mode checkbox is selected. For more information, see “Selecting Schematic Objects” on page 138.

Buffer Pull-down

In the RPN mode, buffer pull-down displays the current stack. In the algebraic mode, buffer pull-down displays a list of previously evaluated expressions.

Selecting Signals

This section describes the three ways you can select signals.

Selecting Signals from the Results Browser

1. Ensure that the Select Mode checkbox in the calculator is selected.

2. Choose Tools – Browser.
   The Results Browser window appears.

3. Double-click on a directory in the left panel of the Results Browser.
   The associated datasets appear in the right-hand list view.
   
   **Note:** The icon next to a directory in either list box indicates that the directory contains subdirectories. You can double-click on the directory to expand (display the subdirectories) or collapse (hide the subdirectories) it.

4. Double-click on the appropriate dataset.
The associated signals appear in the right-hand list view.

5. Select the desired signal.

The selected signal appears in the calculator buffer.

In the SKILL mode, the signal appears along with the dataset and directory name. For example:

```
v("out" ?resultsDir "/export/home/shikha/wavescan/ampsim.raw" ?result "tran-tran")`
```

In the MDL mode, the signal name appears. For example:

```
v(out)
```

**Selecting Traces from the Graph Window**

To select a trace,

1. Ensure that the *Select Mode* checkbox in the calculator is selected.

2. In the Graph Display window, select the trace for the signal you want to add to the calculator.

   An expression for the trace you selected appears in the calculator buffer. In ADE WaveScan, if you selected a trace that was the result of an evaluated expression, WaveScan displays the SKILL function name for that trace in the buffer.

To select a family of traces,

1. Ensure that the *Select Mode and Family* checkboxes in the calculator are selected.

2. In the Graph Display window, select a parametric trace.

   An expression for the parametric family appears in the calculator buffer.

**Selecting Schematic Objects**

You can select schematic objects only in the ADE WaveScan calculator. To select a schematic object,

1. Ensure that the *Select Mode* checkbox is selected.

2. Click the appropriate schematic expression button and select the relevant function.

   The Schematic window appears.

3. Click the appropriate object in the schematic and press the *Esc* key.
An expression corresponding to the selected object appears in the calculator buffer.

The following table describes the available functions for each analysis:

<table>
<thead>
<tr>
<th>Analysis Name</th>
<th>Button</th>
<th>Stands For</th>
</tr>
</thead>
<tbody>
<tr>
<td>tran</td>
<td>vt</td>
<td>Transient voltage</td>
</tr>
<tr>
<td></td>
<td>it</td>
<td>Transient current</td>
</tr>
<tr>
<td>ac</td>
<td>vf</td>
<td>Frequency voltage</td>
</tr>
<tr>
<td></td>
<td>if</td>
<td>Frequency current</td>
</tr>
<tr>
<td>dc</td>
<td>vdc</td>
<td>DC voltage</td>
</tr>
<tr>
<td></td>
<td>idc</td>
<td>DC terminal current</td>
</tr>
<tr>
<td>swept_dc</td>
<td>vs</td>
<td>Source sweep voltage</td>
</tr>
<tr>
<td></td>
<td>is</td>
<td>Source sweep current (I vs V graphs)</td>
</tr>
<tr>
<td>info</td>
<td>op</td>
<td>DC operating point</td>
</tr>
<tr>
<td></td>
<td>opt</td>
<td>Transient operating point</td>
</tr>
<tr>
<td></td>
<td>var</td>
<td>Design variable</td>
</tr>
<tr>
<td></td>
<td>mp</td>
<td>Model parameter</td>
</tr>
<tr>
<td>noise</td>
<td>vn</td>
<td>Noise voltage</td>
</tr>
<tr>
<td>rf</td>
<td>sp</td>
<td>Scattering parameters</td>
</tr>
<tr>
<td></td>
<td>zp</td>
<td>Impedance parameters</td>
</tr>
<tr>
<td></td>
<td>vswr</td>
<td>Voltage standing wave ratio</td>
</tr>
<tr>
<td></td>
<td>yp</td>
<td>Admittance parameters</td>
</tr>
<tr>
<td></td>
<td>hp</td>
<td>H-parameters</td>
</tr>
<tr>
<td></td>
<td>gd</td>
<td>Group delay</td>
</tr>
<tr>
<td></td>
<td>zm</td>
<td>Input impedance if all other ports are matched</td>
</tr>
<tr>
<td>data</td>
<td></td>
<td>Plots a previous analysis</td>
</tr>
</tbody>
</table>

**Building Expressions**

This section describes how to build expressions in the RPN and algebraic modes.
Building Expressions in the RPN Mode

RPN is the default mode for the calculator.

RPN is a way of expressing arithmetic expressions that avoids the use of parenthesis to define priorities for evaluation of operators. In algebraic notation, you might write:

\[(3 + 5) \times (7 - 2)\]

The parentheses tell us to add 3 to 5, then subtract 2 from 7, and multiply the two results. In RPN, the numbers and operators are listed one after the other and form a stack. The most recent number goes at the bottom of the stack. An operator takes the appropriate number of arguments from the bottom of the stack and replaces them by the result of the operation.

In this notation you would write the above expression as:

\[3 \ 5 \ + \ 7 \ 2 \ - \ *\]

Reading from left to right, this is interpreted as follows:

1. Push 3 onto the stack.
2. Push 5 onto the stack. The stack now contains (3, 5).
3. Apply the + operation: take the top two numbers off the stack, add them together, and put the result back on the stack. The stack now contains just the number 8.
4. Push 7 onto the stack.
5. Push 2 onto the stack. It now contains (8, 7, 2).
6. Apply the - operation: take the top two numbers off the stack, subtract the top one from the one below, and put the result back on the stack. The stack now contains (8, 5).
7. Apply the * operation: take the top two numbers off the stack, multiply them, and put the result back on the stack. The stack now contains just the number 40.

The following figure illustrates the appearance of the stack at the end of each step.

<table>
<thead>
<tr>
<th>Step</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td>8</td>
<td>7</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>5</td>
<td>8</td>
<td>7</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>40</td>
<td></td>
</tr>
</tbody>
</table>
Pushing Expressions onto the Stack

After you make an entry into the calculator buffer, you must tell the calculator that you have finished the current entry and are ready to make the next entry.

To push the current expression onto the stack,

➤ Click Enter.

This duplicates the contents of the buffer and pushes it onto the stack.

Clearing the Buffer and Stack

There are several ways to clear the calculator buffer and stack.

To remove a single character from the buffer,

➤ Press the Backspace or Delete key.

To clear the buffer without affecting the stack,

➤ Click Clear.

To clear the buffer and stack,

➤ Click Clst.

To build an expression in the RPN mode,

1. Ensure that the Select Mode checkbox in the calculator is selected.

2. Select a signal. For more information, see “Selecting Signals” on page 137.

   The signal appears in the calculator buffer.

3. In the calculator, click the function you want to use.

   ❑ If you clicked a single-parameter function, it appears in the buffer.

   ❑ If you clicked a multi-parameter function, the parameter panel for the function appears in the right hand side of the Calculator window. Enter the required information and click OK.

   The expression appears in the buffer.

4. Click the Eval button.
If the expression evaluates to a scalar, the result appears in the calculator buffer. If the expression evaluates to a signal, the Graph Window appears with a trace for the evaluated expression.

**Building Expressions in the Algebraic Mode**

To change the calculator mode to algebraic,

➤ Under the *Options* menu, turn off the *Set RPN* entry.

In the algebraic mode you build expressions from left to right. When you click an operator or function key, the operator or function is added to the buffer to the right of the cursor. WaveScan places the cursor to the right of the expression; you can move it by clicking elsewhere in the buffer.

To build an expression in the Algebraic mode,

1. Ensure that the *Select Mode* checkbox in the calculator is selected.
2. Select a signal. For more information, see “Selecting Signals” on page 137. The signal appears in the buffer in a half-highlighted mode. Then,
   - If you select a single-parameter function, the signal is wrapped in the function.
   - If you select a multi-parameter function, the parameter panel for the function appears in the right-hand side of the calculator with the selected signal in the first field. After you complete the parameters values, the signal in the buffer is replaced by the expression.
3. Click the *Eval* button.

If the expression evaluates to a scalar, the result appears in the calculator buffer. If the expression evaluates to a signal, the Graph Window appears with a trace for the evaluated expression.

**Example of Building an Expression**

To build an expression for measuring the delay between an input and output signal,

1. In the Calculator window, ensure that the *Select Mode* checkbox is selected.
2. In the Results Browser window, select the signal *in_p* signal from the *ampsim.raw* folder. For information on how to access this data, see *WaveScan Tutorial*.

   ```
   v("in_p" ?resultsDir "/export/home/shikha/wavescan/ampsim.raw" ?result "tran-tran")
   ```
appears in the buffer (this example assumes you are working with the SKILL calculator).

3. Select the delay function.

The parameter panel for delay appears in the right hand side of the Calculator window.

```
v("in_p" ?resultsDir "/export/home/shikha/wavescan/ampsim.raw" ?result "tran-tran")
```

appears in the Signal1 field.

4. Enter the following information in the function panel of the calculator.

   a. Click in the Signal 2 field.

   b. In the Results Browser window, select the out function.

```
v("out" ?resultsDir "/export/home/shikha/wavescan/ampsim.raw" ?result "tran-tran")
```

appears in the Signal2 field of the calculator.

   c. In the Edge Type1 field, choose rising and click >>>.
The rest of the fields to be specified for `delay` appear.

![Calculator screenshot]

**d.** In the `Edge Type 2` field, choose `rising`.

Ensure that the following values appear:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Value</th>
<th>Field Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Signal1</code></td>
<td><code>in_p</code></td>
<td><code>Edge Type</code></td>
<td><code>rising</code></td>
</tr>
<tr>
<td><code>Signal2</code></td>
<td><code>out</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>Threshold Value 1</code></td>
<td>2.5</td>
<td><code>Edge Number 2</code></td>
<td>1</td>
</tr>
<tr>
<td><code>Threshold Value 2</code></td>
<td></td>
<td><code>Edge Type 2</code></td>
<td><code>either</code></td>
</tr>
<tr>
<td><code>Edge Number 1</code></td>
<td>1</td>
<td><code>Edge Type</code></td>
<td><code>rising</code></td>
</tr>
</tbody>
</table>

5. Click `Apply`.

```plaintext
delay(v("in_p" ?resultsDir "/export/home/shikha/release/sierra/wavescanug/tutorial/new/spectremdl/wscan/ampsim.raw" ?result "tran-tran") 2.5 1 "rising" v("out" ?resultsDir "/export/home/shikha/wavescan/ampsim.raw" ?result "tran-tran") 2.5 1 "rising" )
```
appears in the buffer.

6. Click *Eval*.

WaveScan evaluates the expression and displays the result in the buffer.

**Using the Calculator Functions**

The function categories and functions available in the calculator depends on whether you are in the SKILL or MDL mode of the calculator.

**Function Categories in the SKILL Mode**

The following table lists the categories and functions available in each category in the SKILL mode.

<table>
<thead>
<tr>
<th>Category</th>
<th>Functions Available</th>
</tr>
</thead>
<tbody>
<tr>
<td>Math</td>
<td>1/x, 10<strong>x, abs, dB10, dB20, exp, int, ln, log10, sqrt, x</strong>2, y**x</td>
</tr>
<tr>
<td>Modifier</td>
<td>imag, mag, phase, real</td>
</tr>
<tr>
<td>Programmed Keys</td>
<td>f1, f2, f3, f4</td>
</tr>
<tr>
<td>Programmed RF Keys</td>
<td>rf1, rf2, rf3, rf4, rf5, rf6, rf7, rf8</td>
</tr>
<tr>
<td>RF Functions</td>
<td>Rn, b1f, ga, gac_freq, gac_gain, gmax, gmin, gmsg, gp, gpc_freq, gpc_gain, gt,</td>
</tr>
<tr>
<td></td>
<td>gumx, kf, lsb, nc_freq, nc_gain, nf, nfmin, rn, s11, s12, s21, s22, ssb</td>
</tr>
<tr>
<td>Special</td>
<td>average, bandwidth, clip, compression, compressionVRI, convolve, cross, dBm, delay,</td>
</tr>
<tr>
<td>Functions</td>
<td>deriv, dft, dftbb, eyeDiagram, flip, fourEval, frequency, gainBwProd, gainMargin,</td>
</tr>
<tr>
<td></td>
<td>getAsciiWave, groupDelay, harmonic, harmonicFreq, iiinter, integ, ipn, ipnVRI, lshift,</td>
</tr>
<tr>
<td></td>
<td>overshoot, phaseNoise, phaseMargin, psd, psdbb, riseTime, rms, rmsNoise, root,</td>
</tr>
<tr>
<td></td>
<td>sample, settlingTime, slewRate, spectralPower, stddev, tangent, thd, value,</td>
</tr>
<tr>
<td></td>
<td>xmax, xmin, xval, ymax, ymin</td>
</tr>
<tr>
<td>Trigonometric</td>
<td>acos, acosh, asin, asinh, atan, atanh, cos, cosh, sin, sinh, tan, tanh</td>
</tr>
<tr>
<td>User Defined</td>
<td>The functions you define. For more informations, see “Defining New Functions Using</td>
</tr>
<tr>
<td>Functions</td>
<td>SKILL in ADE” on page 146.</td>
</tr>
</tbody>
</table>

User Defined Functions: The functions you define. For more informations, see “Defining New Functions Using SKILL in ADE” on page 146.
For more information on the above functions, see Appendix D, “Calculator Functions.”

Defining New Functions Using SKILL in ADE

You can define a function and add it to the User Defined Functions category by following these steps:

1. Define the form that prompts for user-defined arguments to the function.
2. Define the syntax of the function in the callback procedure.
3. Register the function.

Defining a Form

The following example shows how to define an input form for a function that takes three arguments. The first argument is the buffer expression. The other two arguments are the boundaries of the range of the expression on which you want to operate.

```
procedure( CreateMyForm() )
    let( ( fieldList a b )
        a = ahiCreateStringField(?name 'from
                                   ?prompt "From"
                                   ?value ""
        )
        b = ahiCreateStringField(?name 'to
                                   ?prompt "To"
                                   ?value ""
        )
    )
    fieldList = list( list( a 5:0 120:25 40 )
                      list( b 160:0 110:25 30 )
                      )
    calCreateSpecialFunctionsForm( 'MyForm
                                     fieldList ))
```

In this example, the From and To fields are string fields created in a two-dimensional form specification for fieldList. The form is created by the call to `calCreateSpecialFunctionsForm`. This function creates and registers the form with the specified form symbol, MyForm.

Defining a Callback Procedure

You define a callback procedure that is called from the entry on the Calculator User Defined Functions category. Since this example uses a form to prompt for additional information required by the special function, the callback procedure is
procedure( MySpecialFunctionCB() 
    calCreateSpecialFunction( 
        ?formSym 'MyForm 
        ?formInitProc 'CreateMyForm 
        ?formTitle "Test" 
        ?formCallback "calSpecialFunctionInput( 'test 
                     '(from to) )"
    )
)

In this procedure, a call is made to `calCreateSpecialFunction`, which creates and displays the form and then builds the expression in the buffer with the specified form fields.

Using Stack Registers in the Procedure

You can use the special symbol `STACK` in the list of form fields to get expressions from the stack.

For example, if you want to insert a stack element between the From and To arguments in the special function expression, you can specify the callback line as follows:

`?formCallback "calSpecialFunctionInput('test '(from STACK to))"

If your special function does not require a form to prompt for additional arguments, you can define your callback as follows:

`procedure( MySpecialFunctionCB() 
    calSpecialFunctionInput( 'test nil )
)
`

Registering the Function

You register the function and callback with the `calRegisterSpecialFunction`:

`calRegisterSpecialFunction( 
    list( "test" 'MySpecialFunctionCB )
)
`

The next time you open the calculator, the functions you defined appear in the User Defined Functions category.

SKILL User Interface Functions for the Calculator

For information on SKILL Functions for the calculator, refer to chapter 22 of the *Virtuoso Analog Design Environment SKILL Language Reference*. 
Assigning Function Keys

You can assign buffer and stack manipulation procedures to the four function keys f1, f2, f3, and f4. To do this, use SKILL commands that you type in the CIW or add to your .cdsinit file.

For example, you can use the f1 key to create the expression for the magnitude and phase of an AC waveform in the buffer by defining the following RPN mode procedure:

```plaintext
procedure(f1( )
    calCalcInput('(enter phase xchxy mag append))
)
```

The `calCalcInput` function manipulates the buffer containing the expression

`VF("/net1")`

to create the expression

`mag(VF("/net1")) phase(VF("/net1"))`

Function Categories in the MDL Mode

The following table lists the categories and functions available in each category in the MDL mode.

<table>
<thead>
<tr>
<th>Category</th>
<th>Functions Available</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>All the functions</td>
</tr>
<tr>
<td>Frequency</td>
<td>bw, gainBwProd, gainMargin, groupDelay, phaseMargin</td>
</tr>
<tr>
<td>General</td>
<td>argmax, argmin, cfft, clip, convolve, cross, crosscorr, crosses, deltax, dutycycle, dutycycles, ft, flip, freq, freq_jitter, histo, ifft, iiint, integ, period_jitter, pp, rms, sample, settlingTime, sign, snr, trim, window, xval, yval</td>
</tr>
<tr>
<td>Math</td>
<td>abs, avg, ceil, cplx, derivexp, floor, int, ln, log10, max, min, mod, movingavg, pow, real, round, sqrt</td>
</tr>
<tr>
<td>Modifier</td>
<td>angle, conj, d2r, db, db10, dbm, im, mag, ph, r2d, re</td>
</tr>
<tr>
<td>PoleZero</td>
<td>pbode, pzfilter</td>
</tr>
<tr>
<td>Statistics</td>
<td>stathisto</td>
</tr>
<tr>
<td>Transient</td>
<td>falltime, overshoot, riseTime, slewrate</td>
</tr>
<tr>
<td>Trig</td>
<td>acos, acosh, asin, asinh, atan, atan2, cos, cosh, sin, sinh, tan, tanh</td>
</tr>
</tbody>
</table>
For more information on the above functions, see Appendix D, “Calculator Functions.”

**Filtering Functions**

To filter functions,

1. In the Calculator window, select one of the choices from the Filter field.

2. Click and type the filter pattern. This pattern works with the value you selected from the cyclic field.

   For example, assume you want to display all math functions whose names begin with c. First, select Math from the cyclic field next to the Filter field. Then, if you are using regular expression syntax, type c.* to display math functions starting with c. If you are using shell syntax, type c* to display math functions starting with c. For more information on regular and shell syntaxes, see “textFilterType” on page 205.

3. Click .

   The functions meeting the filtering criteria you specified are displayed.

**Entering Constants**

To enter a constant to the buffer,

- From the Const menu, select a constant.

  The constant appears in the buffer.

For more information on constants, see Appendix E, “Constants.”

**Printing Expressions**

You can print expressions only in the ADE WaveScan and standalone SKILL calculators. This section describes the various ways in which you can print the expression in the calculator buffer.

To print an expression,

1. In the calculator window, click Results Display ( ).
The Display Results dialog box appears.

2. In the Data field, ensure that Value is selected.

3. Click OK.

WaveScan displays the value of the expression in the Results Display Window.

To print the value of an expression at an X-axis point,

1. In the calculator window, click Results Display ( ).

   The Display Results dialog box appears.

2. In the Data region, select Point.

3. In the X Intercept field, specify the X-axis point.

4. Click OK.
WaveScan displays the value of the expression at the specified X-axis point.

**To print the value of the expression over an X-axis range,**

1. In the calculator window, click *Results Display* ( ).
   
   The Display Results dialog box appears.
2. In the *Data* region, select *Range*.
3. In the *Start/End* fields, specify the start and end of the range.
4. In the *Step/Scale* field, specify the interval for the range. Select *Log* if you want to print the value in a logarithmic scale.
5. Click OK.

WaveScan displays the value of the expression over the specified range.

**Working with Memories**

A memory associates a name with an expression. When you select a memory, the associated expression is added to the calculator buffer.

Memories are available in the SKILL mode of the calculator.

This section describes how you can create a memory for your expression, and copy, delete, and undelete it. You can also save the memories created in a session.
Creating Memories

To create a memory for your expression,

1. In the Calculator window, do one of the following:
   - Click the buffer pull-down (  ) and select the expression for which you want to create a memory.
   - Build an expression.

2. Choose Memories – Table – New Memories.

   The Memories Editor appears in the function panel with your expression in the Expression column.

3. In the Name column, type a name for your expression and click OK.

   WaveScan does the following:
   - Checks the memory name for errors. Duplicate names and illegal characters (such as space or =) are not allowed. WaveScan displays the errors in a message window.
   - Assigns the name to the specified expression, creating a memory that you can use to represent the expression. The memory is displayed under the Memories–Select menu.
Editing Memories

To edit a memory,

1. In the Calculator window, choose Memories – Table – Edit.
   
   The Memories Editor dialog box appears in the function panel. It displays the memories created in the current session.

2. Click in the field you want to edit.

3. Make the desired changes and click OK.

Copying Memories

To copy a memory,

1. In the Calculator window, choose Memories – Table – Edit.

   The Memories Editor dialog box appears in the function panel. It displays the memories created in the current session.

2. Select memory the you want to copy.

3. Do one of the following:

   - Right-click on the selected memory and choose Copy.
   - Choose Memories – Table – Copy.

   WaveScan copies the selected expression and displays it in a new row.

Deleting Memories

To delete a memory,

1. In the Calculator window, choose Memories – Table – Edit.

   The Memory Editor dialog box appears in the function panel. It displays the memories created in the current session.

2. Select the memory you want to delete.

3. Do one of the following:

   - Right-click on the selected memory and choose Delete.
   - Choose Memories – Table – Delete.
WaveScan marks the memory as deleted.

4. Click OK or Apply.

WaveScan deletes the memory. Once you click OK, you cannot undelete the memory.

Undeleting Memories

You can undelete a memory during the time it is marked deleted – once you apply the deletion, the memory is lost.

1. In the Memory Editor window, select the memory you want to undelete.

2. Do one of the following:
   - Choose Memories – Table – Undelete.
   - Right-click on the selected memory and choose Undelete.

3. Click OK.

WaveScan undeletes the selected memory.

Saving Memories

You can save the memories created in a session to a file.

To save memories,

1. In the Calculator window, choose Memories – Save.
The Save dialog box appears.

2. In the Look in field, select the directory where you want to save the memory file.

3. Do one of the following:
   - If you want to overwrite an existing memory file, select that memory file from the listbox below the Look in field.
   - In the File name field, type a name for the memory file you want to save.

4. Click Save.

WaveScan saves the memory list.

To autosave the memories in the current session to the defaultVarFileName file when you close the calculator or exit Wavescan, set the writeDefaultVarFileOnExit variable to true.

**Loading Memories**

You can load a memo file. The file you load replaces the current set of memories, and you lose any unsaved information.

To load a memories file,

1. In the Calculator window, choose Memories – Load.
The Open dialog box appears.

2. In the Look in field, select the directory from which you want to open a memories file.

3. Do one of the following:
   - Select the memories file you want to open from the listbox below the Look in field.
   - In the File name field, type the name of the file you want to open.

4. Click Open.

   WaveScan loads the specified memories file. To display the list of loaded memories, choose Memories – Select in the Calculator window. If the active dataset does not contain all the signals in a memory, the memory is not loaded.

   If the readDefaultVarFileOnStartup is set to true, WaveScan automatically loads the defaultVarFileName file.

**Working with Variables**

You can work with variables only in the MDL calculators.

A variable is an alias for an expression. You can use variables in labels and markers, as building blocks for other expressions, and for saving expressions.

You can create a variable within the context of a dataset; if there is no active dataset, the options under the Var menu are disabled.
This section describes how you can create a variable for your expression, and copy, delete, and undelete it. You can also save the variables created in a session.

Creating Variables

To create a variable for your expression,

1. In the Calculator window, do one of the following:
   - Click the buffer pull-down ( ) and select the expression for which you want to create a variable.
   - Build an expression.

2. Choose Var – Table – New Variable.

   The Variable Editor appears in the function panel with your expression in the Expression column.

3. In the Name column, type a name for your expression and click OK.

   WaveScan

   a. Checks the variable name for errors. Duplicate names and illegal characters (such as space or =) are not allowed. WaveScan displays the errors in a message window.
b. Assigns the name to the specified expression, creating a variable that you can use to represent the expression. The variable is displayed under the Var–Select menu. When you change the active data directory, variables that do not contain the signals in the active dataset are not displayed.

**Editing Variables**

To edit a variable,

1. In the Calculator window, choose Var – Table – Edit.

   The Variable Editor dialog box appears in the function panel. It displays the variables created in the current session.

2. Click in the field you want to edit.

3. Make the desired changes and click OK.

**Copying Variables**

To copy a variable,

1. In the Calculator window, choose Var – Table – Edit.

   The Variable Editor dialog box appears in the function panel. It displays the variables created in the current session.

2. Select the variable you want to copy.

3. Do one of the following:
   
   - Right-click on the selected variable and choose Copy.
   - Choose Var – Table – Copy.

   WaveScan copies the selected expression and displays it in a new row.

**Deleting Variables**

To delete a variable,

1. In the Calculator window, choose Var – Table – Edit.

   The Variable Editor dialog box appears in the function panel. It displays the variables created in the current session.
2. Select the variable you want to delete.

3. Do one of the following:
   - Right-click on the selected variable and choose *Delete*.
   - Choose *Var – Table – Delete*.
   
   WaveScan marks the variable as deleted.

4. Click *OK* or *Apply*.

   WaveScan deletes the variable. Once you click *OK*, you cannot undelete the variable.

**Undeleting Variables**

You can undelete a variable during the time it is marked deleted – once you apply the deletion, the variable is lost.

1. In the Variable Editor window, select the variable you want to undelete.

2. Do one of the following:
   - Choose *Var – Table – Undelete*.
   - Right-click on the selected variable and choose *Undelete*.

3. Click *OK*.

   WaveScan undeletes the selected variable.

**Saving Variables**

You can save the variables created in a session to a file.

To save variables,

1. In the Calculator window, choose *Var – Save*. 
The Save dialog box appears.

![Image: Save dialog box]

2. In the *Look in* field, select the directory where you want to save the variable file.

3. Do one of the following:
   - If you want to overwrite an existing variable file, select that variable file from the listbox below the *Look in* field.
   - In the *File name* field, type a name for the variable file you want to save.

4. Click *Save*.

   WaveScan saves the variable list.

To autosave the variables in the current session to the *defaultVarFileName* file when you close the calculator or exit Wavescan, set the *writeDefaultVarFileOnExit* variable to true.

### Loading Variables

You can load a variables file. The file you load replaces the current set of variables, and you lose any unsaved information.

To load a variables file,

1. In the Calculator window, choose *Var – Load*. 
The Open dialog box appears.

2. In the Look in field, select the directory from which you want to open a variables file.

3. Do one of the following:
   - Select the variables file you want to open from the listbox below the Look in field.
   - In the File name field, type the name of the file you want to open.

4. Click Open.

WaveScan loads the specified variable file. To display the list of loaded variables, choose Var – Select in the Calculator window. If the active dataset does not contain all the signals in a variable, the variable is not loaded.

If the readDefaultVarFileOnStartup is set to true, WaveScan automatically loads the defaultVarFileName file.

**Menu Bar and Toolbar**

This section describes the menu and toolbar options.
# Menu Bar

For a description of the menu commands, see the table below.

<table>
<thead>
<tr>
<th>Item</th>
<th>Usage and Description</th>
<th>For More Information, See</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Window</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Close</td>
<td>Closes the Calculator window. If the Calculator is the only WaveScan window open, Close exits WaveScan.</td>
<td></td>
</tr>
<tr>
<td>Exit</td>
<td>Available only in the standalone calculator. Closes all windows and exits WaveScan.</td>
<td>“Exiting WaveScan” on page 21.</td>
</tr>
<tr>
<td><strong>Tools</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Browser</td>
<td>Opens the Results Browser window.</td>
<td>Chapter 2, “Accessing Data.”</td>
</tr>
<tr>
<td><strong>Memories</strong></td>
<td></td>
<td>Working with Memories on page 151.</td>
</tr>
<tr>
<td><strong>Table</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edit</td>
<td>Displays the Memories Editor so that you can edit a memory.</td>
<td></td>
</tr>
<tr>
<td>New Memories</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Copy</td>
<td>Copies the selected memory.</td>
<td></td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes the selected memory.</td>
<td></td>
</tr>
<tr>
<td>Undelete</td>
<td>Undeletes a recently deleted memory.</td>
<td></td>
</tr>
<tr>
<td>Select</td>
<td>Displays memories that are valid for the active dataset.</td>
<td></td>
</tr>
<tr>
<td>Load</td>
<td>Opens the Open dialog box so that you can select the memory you want to load.</td>
<td></td>
</tr>
<tr>
<td>Save</td>
<td>Opens the Save dialog box so that you can save the memory.</td>
<td></td>
</tr>
<tr>
<td>Var</td>
<td>Available only in the standalone MDL calculator.</td>
<td>Working with Variables on page 156.</td>
</tr>
</tbody>
</table>
WaveScan User Guide
Working with the Calculator

<table>
<thead>
<tr>
<th>Item</th>
<th>Usage and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Table</strong></td>
<td></td>
</tr>
<tr>
<td>Edit</td>
<td>Displays the Variable Editor so that you can edit a variable.</td>
</tr>
<tr>
<td>New</td>
<td>Displays the Variable Editor so that you can create a variable for the selected expression.</td>
</tr>
<tr>
<td>Copy</td>
<td>Copies the selected variable.</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes the selected variable.</td>
</tr>
<tr>
<td>Undelete</td>
<td>Undeletes a recently deleted variable.</td>
</tr>
<tr>
<td>Select</td>
<td>Displays variables that are valid for the active dataset.</td>
</tr>
<tr>
<td>Load</td>
<td>Opens the Open dialog box so that you can select the variable you want to load.</td>
</tr>
<tr>
<td>Save</td>
<td>Opens the Save dialog box so that you can save the variable.</td>
</tr>
<tr>
<td>Const</td>
<td>Displays a list of constants.</td>
</tr>
</tbody>
</table>

| Options   |                                                                                       |
| Set RPN   | Sets the calculator mode to RPN. Turning this option off changes the calculator mode to algebraic. |

| Help      |                                                                                       |
| Help      | Displays this document, the *WaveScan User Guide*.                                    |

**Toolbar**

For a description of the toolbar, see the table below.

<table>
<thead>
<tr>
<th>Item</th>
<th>Usage and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Plot Expression]</td>
<td>Plots the result of the expression.</td>
</tr>
</tbody>
</table>
### Results Display

Available only in the ADE WaveScan and standalone SKILL calculators. Opens the Display Results dialog box so that you can specify how you want to print the calculator buffer expression.

### Status Bar

Displays warning and errors. An example is shown in the figure below.

```plaintext
> No data for 'argmin()'
```
Working with Tables

WaveScan helps you create and format tables to represent your simulation data in a text format.

This chapter discusses the following topics:

- “Opening the Report Table Window” on page 166
- “Viewing Types of Tables” on page 166
- “Formatting Tables” on page 170
- “Transposing Rows and Columns” on page 173
- “Moving Columns” on page 173
- “Resizing Columns” on page 173
- “Selecting Columns” on page 174
- “Hiding and Revealing Columns” on page 175
- “Sorting Columns” on page 176
- “Printing Tables” on page 176
- “Saving Tables” on page 179
- “Opening Saved Tables” on page 181
- “Menu Bar and Toolbar” on page 182
Opening the Report Table Window

To open the Report Table window,

1. In the Results Browser window, select the data you want to display in a table.

2. Choose Tools – Table.

   The Report Table window appears. For information about the parts of the window, click the cross-references.

   ![Report Table Window]

   For information on menu and toolbar options, see Menu Bar and Toolbar on page 182.

Viewing Types of Tables

The type of data you select determines the table created.

Scalar Data

You can create a table at the following levels:

- Dataset or subcircuit. Wavescan generates a separate table for each primitive (building blocks of the circuit). You can display the table for a particular primitive by selecting it from the cyclic field at the bottom center of the Report Table window. The instances in the dataset or subcircuit are displayed in rows, and the output parameters for each instance are displayed in columns.
WaveScan User Guide
Working with Tables

- Instance. WaveScan displays the instances in rows and the output parameters in columns.
- Parameter. WaveScan displays the instance and parameter.

If you create a table for parametric scalar data, it displays the sweep points for all primitives.

An Example of a Table for a Scalar Dataset

To create a table for a scalar dataset,

1. In the Results Browser window, select the `dcOpInfo-info` dataset from the `ampsim.raw` folder. For information on how to access this data, see WaveScan Tutorial.

2. Choose Tools – Table.

The following table appears.

The rows display the instances in the dataset and the columns display the output parameters for each instance. The primitives for this dataset are bjt, bsim3v3, capacitor, isource, resistor, v cvs, and vsource.

An Example of a Table for an Instance

To create a table for an operating point instance,

1. In the Results Browser window, double-click on the `dcOpInfo-info` dataset. For information on how to access this data, see WaveScan Tutorial.
2. In the right-hand list box, select \textit{ll}.

3. Choose \textit{Tools – Table}.

The following table appears.

<table>
<thead>
<tr>
<th>Name</th>
<th>betaac (A/A)</th>
<th>betadc (A/A)</th>
<th>cmu (F)</th>
<th>cmux (F)</th>
<th>cpi (F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>m.0</td>
<td>158.5</td>
<td>169.8</td>
<td>1.575E-13</td>
<td>0.0</td>
<td>4.546E-12</td>
</tr>
<tr>
<td>m.2</td>
<td>11.46</td>
<td>19.23</td>
<td>7.773E-13</td>
<td>0.0</td>
<td>3.599E-10</td>
</tr>
<tr>
<td>m.3</td>
<td>12.15</td>
<td>20.36</td>
<td>4.32E-13</td>
<td>0.0</td>
<td>3.738E-10</td>
</tr>
<tr>
<td>m.4</td>
<td>12.90</td>
<td>21.59</td>
<td>3.488E-13</td>
<td>0.0</td>
<td>3.883E-10</td>
</tr>
</tbody>
</table>

The rows display instances and the columns display the output parameters.

\textbf{Swept Data}

You can create a table at the following levels:

- \textit{Dataset}. The table contains signals all the way down the hierarchy (including all subcircuits). You can choose to display only voltage or current signals by selecting \textit{V} or \textit{I} from the cyclic field at the bottom center of the Report Table window.

- \textit{Subcircuit}. The table displays time in a row and the instances in columns.

- \textit{Signal}. WaveScan displays the time values (X-axis) in a column, and the voltage or current (Y-axis) value for each time point in a row.

If you create a table for parametric swept data, WaveScan generates a separate table for each combination of sweep values.

\textbf{An Example of a Table for a Swept Dataset}

To create a table for a swept dataset,

1. In the Results Browser window, double-click on \texttt{ampsim.raw}. (X-axis). For information on how to access this data, see \textit{WaveScan Tutorial}.

2. In the right-hand list box, select the \texttt{tran-tran} dataset.

3. Choose \textit{Tools – Table}.
The following table appears.

<table>
<thead>
<tr>
<th>time (s)</th>
<th>V0:p (A)</th>
<th>V1:p (A)</th>
<th>V2:p (A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>-1.416E-3</td>
<td>9.162E-4</td>
</tr>
<tr>
<td>5.0E-10</td>
<td>-1.128E-12</td>
<td>-1.416E-3</td>
<td>9.162E-4</td>
</tr>
<tr>
<td>1.000E-9</td>
<td>-1.276E-12</td>
<td>-1.416E-3</td>
<td>9.162E-4</td>
</tr>
<tr>
<td>1.039E-9</td>
<td>-5.684E-3</td>
<td>3.876E-3</td>
<td>8.490E-4</td>
</tr>
<tr>
<td>1.1E-9</td>
<td>-6.313E-3</td>
<td>3.959E-3</td>
<td>1.028E-3</td>
</tr>
<tr>
<td>1.150E-9</td>
<td>-6.678E-3</td>
<td>3.974E-3</td>
<td>1.351E-3</td>
</tr>
<tr>
<td>1.177E-9</td>
<td>-6.820E-3</td>
<td>4.060E-3</td>
<td>1.395E-3</td>
</tr>
</tbody>
</table>

The table displays all signals in the tran-tran dataset.

**An Example of a Table for a Signal**

To create a table for a signal,

1. In the Results Browser window, double-click on the tran-tran dataset.
2. Click on the out signal.
3. Choose Tools – Table.

The following table appears.

<table>
<thead>
<tr>
<th>time (s)</th>
<th>out (V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>4.441E-3</td>
</tr>
<tr>
<td>5.000E-10</td>
<td>4.441E-3</td>
</tr>
<tr>
<td>1.000E-9</td>
<td>4.441E-3</td>
</tr>
<tr>
<td>1.039E-9</td>
<td>6.872E-2</td>
</tr>
<tr>
<td>1.100E-9</td>
<td>1.601E-1</td>
</tr>
<tr>
<td>1.150E-9</td>
<td>2.240E-1</td>
</tr>
<tr>
<td>1.177E-9</td>
<td>2.565E-1</td>
</tr>
<tr>
<td>1.203E-9</td>
<td>2.857E-1</td>
</tr>
</tbody>
</table>

The table displays time values (X-axis) in a column, and the voltage (Y-axis) value for each time point in a row.
Formatting Tables

You can control the format of column data in your tables. If you format the cells of a column, the exponent is attached to the value in each cell. If you format the column header, the common factor from each cell is attached to the header.

Formatting Column Cells

To edit format attributes for the cells of a column,

1. Click on a column.
2. Choose View – Format.

The Format Attributes dialog box appears.

3. In the Active Format Location cyclic field, select Cell.
4. In the Scale Format field, select one of the following:

   Scientific looks like

<table>
<thead>
<tr>
<th>cmu (F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.575E-13</td>
</tr>
<tr>
<td>7.773E-13</td>
</tr>
<tr>
<td>4.320E-13</td>
</tr>
<tr>
<td>3.488E-13</td>
</tr>
</tbody>
</table>
5. Specify the number of Significant Digits to be displayed in each cell.

6. Click OK.

WaveScan applies the changes to the selected column in the Report Table window.

**Formatting a Column Header**

To edit format attributes for a column header,

1. Choose *View – Format*.

   The Format Attributes dialog box appears.

2. In the *Active Format Location* cyclic field, select *Header*. The Format Attributes dialog box looks like the figure below.
3. In the Scale Format field, select Engineering or Suffix, and specify the corresponding Scale Factor. The table below displays the scale factors for engineering as well as suffix formats.

<table>
<thead>
<tr>
<th>Engineering</th>
<th>Suffix</th>
</tr>
</thead>
<tbody>
<tr>
<td>10E12</td>
<td>T</td>
</tr>
<tr>
<td>10E9</td>
<td>G</td>
</tr>
<tr>
<td>10E6</td>
<td>M</td>
</tr>
<tr>
<td>10E3</td>
<td>K</td>
</tr>
<tr>
<td>10E-3</td>
<td>m</td>
</tr>
<tr>
<td>10E-6</td>
<td>u</td>
</tr>
<tr>
<td>10E-9</td>
<td>n</td>
</tr>
<tr>
<td>10E-12</td>
<td>p</td>
</tr>
<tr>
<td>10E-15</td>
<td>f</td>
</tr>
<tr>
<td>10E-18</td>
<td>a</td>
</tr>
</tbody>
</table>

4. Type the number of Significant Digits.

The following figure displays a column with the header formatted in the Engineering scale using a factor of 10E-3 with 4 significant digits.
The following figure displays the same column with the header formatted in the Suffix scale using a factor of $m$ with 4 significant digits.

<table>
<thead>
<tr>
<th>betaac (mA/A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>158500</td>
</tr>
<tr>
<td>11460</td>
</tr>
<tr>
<td>12150</td>
</tr>
<tr>
<td>12900</td>
</tr>
</tbody>
</table>

5. Click OK.

WaveScan applies the changes to the selected column in the Report Table window.

**Transposing Rows and Columns**

To transpose rows and columns,

➤ Choose View – Transpose.

The table columns and rows are transposed; column information is displayed in rows and vice versa. You cannot format a table when the rows and columns are transposed from their original locations.

**Moving Columns**

You can move any column, except the independent variable (time or frequency) column, to a new position.

To move a column,

1. Click a column header to select the column.
2. Drag the column to the new position and release the mouse.

   WaveScan moves the selected column.

**Resizing Columns**

You can resize all columns except the independent variable (time or frequency) column.
To change the column size,

➤ Click on the right edge of a column header, and drag it left or right to make the column narrower or wider.

Selecting Columns

You can select any column in the table by clicking on it. To select multiple columns, hold down the Control or Shift key while clicking on the columns you want to select.

To de-select a column, Ctrl-click on a selected column.

An Example of Selecting Multiple Columns

The following figure shows a table in the Report Table window. No columns are selected.

![Table example]

1. Select the single column, betaac (A/A).
2. Add the next three columns by Shift-clicking on cmux (F).
3. Add the column csub (F) by Ctrl-clicking on it.
The table now looks like the figure below.

![Table Image]

**Hiding and Revealing Columns**

You can hide or reveal multiple columns at a time.

**To hide a column,**

1. Select a column or columns.
2. Choose *View – Hide Column.*

   WaveScan hides the selected column or columns.

**To reveal hidden columns,**

1. Choose *View – Reveal Column.* This menu option is available only if one or more columns are hidden.
A list of hidden columns appears in the Reveal Columns dialog box.

2. Select the column or columns you want to reveal.
3. Click OK.

WaveScan reveals the selected column or columns. The positions of the revealed columns are the same as before they were hidden.

Sorting Columns

Initially, tables are sorted by the left-most column (time or frequency for non-operating point data and Name for operating-point data). When you sort a column for the first time, WaveScan sorts it in an ascending order. The next time you sort the same column, WaveScan flips it to descending order.

To sort a column,

1. Select a column.
2. Choose View – Sort Column.

WaveScan sorts the selected column.

Printing Tables

You can print a selected table or all the tables in the Report Table window.
To print tables,

1. In the Report Table window, do one of the following
   - Choose File – Print to print a single table.
   - Choose File – Print All to print all tables.

The Print dialog box appears.

2. In the Pages section, select one of the following:
   - All to print all pages.
   - From Page to specify a page range. Type the start and end page numbers.
   - Selection to print the current page.

3. In the Copies field, enter the number of copies you want to print.

4. In the Options section, specify whether you want to print the title sheet. The default title is Print Table. Enter another title if you want to rename the title sheet.

5. In the Print To section, do one of the following:
   - Select Printer to print to a printer and type the printer name.
   - Select File to print to a file.
6. Click the Properties tab.

7. In the Orientation field, specify whether you want to print in portrait or landscape mode.

8. In the Paper field, select the paper size and specify whether you want to print single-sided or duplex prints.

   If you select Duplex, by default the text is read by flipping the page about a vertical axis, i.e., the top of side one and the top of side two are along the same edge. To print so that the page flips about a horizontal axis, select Tumble.

9. In the Quality field, specify whether you want the quality of printout to be high, normal, or draft.

10. Do one of the following:

    - If you selected Printer in Step 3, click Print.
      WaveScan prints the table or tables. The footer in the printout displays the signal type or primitive. Specifying Commands in the Print tab and Color in the Properties tab does not affect the printed table.

    - If you selected File in Step 3, click Save.
      The Save dialog box appears. Type a file name for the table or tables and click Save.
      WaveScan prints the table to the specified file.
Saving Tables

You can save tables as XML or text files.

Saving Tables in XML Format

A saved table includes the following information:

- The location of the data (data directory and data set). The data in the table is not saved. If your simulation data changes, the table reflects the changes.
- Table attributes.

To save a table in XML format,

1. In the Report Table window, choose File – Save.
   The Save Table dialog box appears.

2. In the Files of type field, specify whether you want to display all files or files with the .tbl suffix. The suffix for table files is specified by the filesuffix variable in the .cdsenv file.

3. In the Look in field, select the directory where you want to save the table.

4. Do one of the following:
   - If you want to overwrite an existing table file, select that table file from the listbox below the Look in field.
In the *File name* field, type a name for the table file you want to save.

5. Click *Save*.

WaveScan saves the table.

### Saving Tables in Text Format

You can save a table as a text file and import it into spreadsheets such as Microsoft Excel.

**Note:** In the current version of WaveScan, you cannot import saved text back into a WaveScan table.

To save a table in text format,

1. In the Report Table window, choose *File – Save As CSV*.

   The Save As Text dialog box appears.

2. In the *Look in* field, select the directory where you want to save the table.

3. In the *File name* field, specify a name for the table file you want to save. The default is `report_table_current_type.csv` where *current_type* is the signal type displayed in the table. For example, if a table displays voltage signals, the default file name for it would be `report_table_V.csv`.

4. Click *Save*.
WaveScan saves the table as a text file. The text file contains the column headers separated by commas in the first row and then the values separated by commas in subsequent rows. For the following table,

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>in_m</td>
<td>4.441E-3</td>
</tr>
<tr>
<td>in_p</td>
<td>0.0</td>
</tr>
<tr>
<td>net10</td>
<td>-4.441E-3</td>
</tr>
<tr>
<td>net35</td>
<td>4.438</td>
</tr>
<tr>
<td>out</td>
<td>4.441E-3</td>
</tr>
<tr>
<td>vdd</td>
<td>5.000</td>
</tr>
<tr>
<td>vss</td>
<td>-5.000</td>
</tr>
</tbody>
</table>

The `report_table_I.csv` file looks as shown below.

"Name", "Value"
"in_m", 4.441E-3
"in_p", 0.0
"net10", -4.441E-3
"net35", 4.438
"out", 4.441E-3
"vdd!", 5.000
"vss!", -5.000

The text file displays the values displayed in the table. If you hide a column, for example, the hidden column is not displayed in the text file.

**Opening Saved Tables**

This section describes how to open saved tables. In the current version of WaveScan, you cannot open tables saved as text files in WaveScan.

To open a table,

1. Do one of the following:
   - In the Results Browser window, choose *File – Open Table*.
   - In the Report Table window, choose *File – Open*. 
The Open Table dialog box appears.

2. In the *Files of type* field, specify whether you want to display all files or files with the `.tbl` suffix. The suffix for table files is specified by the `filesuffix` variable in the `.cdsenv` file.

3. In the *Look in* field, select the directory from which you want to open a table.

4. Do one of the following:
   - Select the table file you want to open from the listbox below the *Look in* field.
   - In the *File name* field, type the name of the file you want to open.

5. Click *Open*.

WaveScan opens the specified table file.

### Menu Bar and Toolbar

This section describes the menu and toolbar options in the Report Table window.
Menu Bar

For a description of the menu commands, see the table below.

<table>
<thead>
<tr>
<th>Item</th>
<th>Usage and Description</th>
<th>For More Information, See</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>File</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Open</em></td>
<td>Opens the Open Table dialog box so that you can open a saved table.</td>
<td>“Opening Saved Tables” on page 181.</td>
</tr>
<tr>
<td><em>Save</em></td>
<td>Opens the Save Table dialog box so that you can save a table in XML format.</td>
<td>“Saving Tables in XML Format” on page 179.</td>
</tr>
<tr>
<td><em>Save As CSV</em></td>
<td>Opens the Save As Text dialog box so that you can save a table in text format.</td>
<td>“Saving Tables in Text Format” on page 180.</td>
</tr>
<tr>
<td><em>Print</em></td>
<td>Opens the Print dialog box so that you can print a table.</td>
<td>“Printing Tables” on page 176.</td>
</tr>
<tr>
<td><em>Print All</em></td>
<td>Opens the Print dialog box so that you can print all the tables in the Report Table window.</td>
<td>“Printing Tables” on page 176.</td>
</tr>
<tr>
<td><em>Save Session</em></td>
<td>Opens the Save Session dialog box so that you can save a session.</td>
<td>“Saving a Session” on page 22.</td>
</tr>
<tr>
<td><em>Close</em></td>
<td>Closes the Report Table window. If this is the only WaveScan window open, Close exits WaveScan.</td>
<td></td>
</tr>
<tr>
<td><em>Exit</em></td>
<td>Closes all windows and exits WaveScan.</td>
<td>“Exiting WaveScan” on page 21.</td>
</tr>
<tr>
<td><strong>View</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Hide Column</em></td>
<td>Hides the selected column.</td>
<td>“Hiding and Revealing Columns” on page 175.</td>
</tr>
<tr>
<td><em>Reveal Column</em></td>
<td>Displays a list of hidden columns so you can choose the column you want to reveal.</td>
<td>“Hiding and Revealing Columns” on page 175.</td>
</tr>
<tr>
<td><em>Sort Column</em></td>
<td>Sorts the selected column in ascending or descending order.</td>
<td>“Sorting Columns” on page 176.</td>
</tr>
</tbody>
</table>
WaveScan User Guide
Working with Tables

<table>
<thead>
<tr>
<th>Item</th>
<th>Usage and Description</th>
<th>For More Information, See</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format</td>
<td>Opens the Format Attributes dialog box so you can edit column attributes.</td>
<td>“Formatting Tables” on page 170.</td>
</tr>
<tr>
<td>Tools</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Browser</td>
<td>Opens the Results Browser window.</td>
<td>Chapter 2, “Accessing Data.”</td>
</tr>
<tr>
<td>Help</td>
<td>Displays this document, the WaveScan User Guide.</td>
<td></td>
</tr>
</tbody>
</table>

### Toolbar

The Report Table window toolbar contains only one item, which is described below.

<table>
<thead>
<tr>
<th>Item</th>
<th>Usage and Description</th>
<th>For More Information, See</th>
</tr>
</thead>
<tbody>
<tr>
<td>Print</td>
<td>Opens the Print window so that you can print a table.</td>
<td>“Printing Tables” on page 176</td>
</tr>
</tbody>
</table>

### Arrow Keys

You can use the arrow keys to move among

- Voltages and currents in tables of swept data.
- Primitives in tables of scalar data.

- Next type displays the table for the next primitive.
- Previous type displays the table for the previous primitive.
- Last type displays the table for the last primitive.
- First type displays the table for the first primitive.

### Status Bar

Displays the following:

- Warnings and error messages.
Prompts for further action.

Pop-up Menu

This menu appears when you right-click in the Report Table window. For a description of the menu options, see the table below.

<table>
<thead>
<tr>
<th>Item</th>
<th>Usage and Description</th>
<th>For More Information, See</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hide Column</strong></td>
<td>Hides the selected column.</td>
<td>“Hiding and Revealing Columns” on page 175.</td>
</tr>
<tr>
<td><strong>Reveal Column</strong></td>
<td>Displays a list of hidden columns so you can choose the column you want to reveal.</td>
<td>“Hiding and Revealing Columns” on page 175.</td>
</tr>
<tr>
<td><strong>Sort Column</strong></td>
<td>Sorts the column in ascending or descending order.</td>
<td>“Sorting Columns” on page 176</td>
</tr>
<tr>
<td><strong>Transpose</strong></td>
<td>Transposes rows and columns.</td>
<td>“Transposing Rows and Columns” on page 173.</td>
</tr>
<tr>
<td><strong>Format</strong></td>
<td>Opens the Format Attributes dialog box so you can edit table attributes.</td>
<td>“Formatting Tables” on page 170</td>
</tr>
<tr>
<td><strong>ToClipboard</strong></td>
<td>Stores the value of the selected cell in the system buffer. You can then paste the value using the Paste key or the middle mouse button.</td>
<td></td>
</tr>
</tbody>
</table>
WaveScan Variables

From release 5.1.41 onwards, the variables and values that specify the basic behavior of the component tools of WaveScan are part of the .cdsenv file. For information on the order in which WaveScan reads the .cdsenv file, see Creating the .cdsenv File in Chapter 10 of the Cadence Design FrameworkII User Guide.

If you already have a .wsenv file on your computer, you must rename it as .cdsenv, or copy the contents of the .wsenv file to the .cdsenv file.

This appendix describes the WaveScan variables in the .cdsenv file. In each entry, the first column is the tool, the second column is the variable, the third column is the data type, and the fourth column contains the value to be used.
The default `.cdsenv` file contains the following entries.

<table>
<thead>
<tr>
<th>Application Variables</th>
<th>wavescan.application</th>
<th>statefile</th>
<th>string</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>&quot;wavescan.xml&quot;</td>
<td></td>
</tr>
<tr>
<td>wavescan.application</td>
<td>readstate</td>
<td>string &quot;true&quot;</td>
<td></td>
</tr>
<tr>
<td>wavescan.application</td>
<td>writestate</td>
<td>string &quot;false&quot;</td>
<td></td>
</tr>
<tr>
<td>wavescan.application</td>
<td>graphtemplatefile</td>
<td>string &quot;none&quot;</td>
<td></td>
</tr>
<tr>
<td>wavescan.application</td>
<td>exitdialog</td>
<td>string &quot;true&quot;</td>
<td></td>
</tr>
<tr>
<td>wavescan.application</td>
<td>envdirectory</td>
<td>string &quot;./&quot;</td>
<td></td>
</tr>
<tr>
<td>wavescan.application</td>
<td>readDefaultVarFileOnStartup</td>
<td>string &quot;false&quot;</td>
<td></td>
</tr>
<tr>
<td>wavescan.application</td>
<td>writeDefaultVarFileOnExit</td>
<td>string &quot;false&quot;</td>
<td></td>
</tr>
<tr>
<td>wavescan.application</td>
<td>defaultVarFileName</td>
<td>string &quot;.wsvariables&quot;</td>
<td></td>
</tr>
<tr>
<td>wavescan.application</td>
<td>printer</td>
<td>string &quot;&quot;</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Results Browser Variables</th>
<th>wavescan.browser</th>
<th>historyLength</th>
<th>string &quot;10&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>plotStyle</td>
<td>string &quot;replace&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>dataDirHome</td>
<td>string &quot;./&quot;</td>
<td></td>
</tr>
</tbody>
</table>

| Filter Variable | wavescan.filter | textFilterType | string "shell" |

<table>
<thead>
<tr>
<th>Graph Frame Variables</th>
<th>wavescan.graphFrame</th>
<th>width</th>
<th>string &quot;570&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>wavescan.graphFrame</td>
<td>height</td>
<td>string &quot;590&quot;</td>
</tr>
<tr>
<td></td>
<td>wavescan.graphFrame</td>
<td>autoTraceSelect</td>
<td>string &quot;true&quot;</td>
</tr>
<tr>
<td></td>
<td>wavescan.graphFrame</td>
<td>rightMouseZoom</td>
<td>string &quot;True&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rectangular Graph Variables</th>
<th>wavescan.rectGraph</th>
<th>background</th>
<th>string &quot;white&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>wavescan.rectGraph</td>
<td>foreground</td>
<td>string &quot;black&quot;</td>
</tr>
<tr>
<td></td>
<td>wavescan.rectGraph</td>
<td>visibleDigitalRows</td>
<td>string &quot;3&quot;</td>
</tr>
<tr>
<td></td>
<td>wavescan.rectGraph</td>
<td>visibleStripChartRows</td>
<td>string &quot;5&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Circular Graph Variables</th>
<th>wavescan.circGraph</th>
<th>background</th>
<th>string</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>wavescan.circGraph</td>
<td>foreground</td>
<td>string &quot;black&quot;</td>
</tr>
</tbody>
</table>
### Graph Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>wavescan.graph visibleLegendRows</td>
<td>string &quot;3&quot;</td>
</tr>
<tr>
<td>wavescan.graph fileSuffix</td>
<td>string &quot;grf&quot;</td>
</tr>
<tr>
<td>wavescan.graph subTitle</td>
<td>string &quot;&quot;</td>
</tr>
<tr>
<td>wavescan.graph displayDate</td>
<td>string &quot;true&quot;</td>
</tr>
<tr>
<td>wavescan.graph defaultSubtitle</td>
<td>string &quot;true&quot;</td>
</tr>
<tr>
<td>wavescan.graph cursorOn</td>
<td>string &quot;false&quot;</td>
</tr>
<tr>
<td>wavescan.graph majGridsOn</td>
<td>string &quot;true&quot;</td>
</tr>
<tr>
<td>wavescan.graph minGridsOn</td>
<td>string &quot;true&quot;</td>
</tr>
<tr>
<td>wavescan.graph majorForeground</td>
<td>string &quot;gray&quot;</td>
</tr>
<tr>
<td>wavescan.graph minorForeground</td>
<td>string &quot;lightGray&quot;</td>
</tr>
<tr>
<td>wavescan.graph snapOn</td>
<td>string &quot;snapOff&quot;</td>
</tr>
<tr>
<td>wavescan.graph snapOff</td>
<td>string</td>
</tr>
<tr>
<td>wavescan.graph fontName</td>
<td>string &quot;Dialog&quot;</td>
</tr>
<tr>
<td>wavescan.graph fontStyle</td>
<td>string &quot;plain&quot;</td>
</tr>
<tr>
<td>wavescan.graph fontSize</td>
<td>string &quot;medium&quot;</td>
</tr>
<tr>
<td>wavescan.graph stripChartOn</td>
<td>string &quot;false&quot;</td>
</tr>
</tbody>
</table>

### Axis Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>wavescan.depAxis logScale</td>
<td>string &quot;false&quot;</td>
</tr>
<tr>
<td>wavescan.depAxis forceOrigin</td>
<td>string &quot;false&quot;</td>
</tr>
<tr>
<td>wavescan.indepAxis logScale</td>
<td>string &quot;false&quot;</td>
</tr>
<tr>
<td>wavescan.indepAxis forceOrigin</td>
<td>string &quot;false&quot;</td>
</tr>
<tr>
<td>wavescan.axis foreground</td>
<td>string &quot;black&quot;</td>
</tr>
<tr>
<td>wavescan.axis significantDigits</td>
<td>string &quot;3&quot;</td>
</tr>
</tbody>
</table>

### Circular Scale Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>wavescan.circularScale polarForeground</td>
<td>string &quot;gray&quot;</td>
</tr>
<tr>
<td>wavescan.circularScale impedForeground</td>
<td>string &quot;gray&quot;</td>
</tr>
<tr>
<td>wavescan.circularScale admittForeground</td>
<td>string &quot;gray&quot;</td>
</tr>
<tr>
<td>wavescan.circularScale significantDigits</td>
<td>string &quot;2&quot;</td>
</tr>
<tr>
<td>wavescan.circularScale &quot;realimag&quot;</td>
<td>string</td>
</tr>
</tbody>
</table>
### WaveScan User Guide

#### WaveScan Variables

<table>
<thead>
<tr>
<th>Trace Variables</th>
<th>Variable</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>wavescan.trace.type</td>
<td>string</td>
<td>&quot;line&quot;</td>
</tr>
<tr>
<td></td>
<td>wavescan.trace.lineThickness</td>
<td>string</td>
<td>&quot;fine&quot;</td>
</tr>
<tr>
<td></td>
<td>wavescan.trace.lineStyle</td>
<td>string</td>
<td>&quot;solid&quot;</td>
</tr>
<tr>
<td></td>
<td>wavescan.trace.symbolsOn</td>
<td>string</td>
<td>&quot;false&quot;</td>
</tr>
<tr>
<td></td>
<td>wavescan.trace.symbolStyle</td>
<td>string</td>
<td>&quot;plus&quot;</td>
</tr>
<tr>
<td></td>
<td>wavescan.trace.symbolCount</td>
<td>string</td>
<td>&quot;20&quot;</td>
</tr>
<tr>
<td></td>
<td>wavescan.trace.dToAConversion</td>
<td>string</td>
<td>&quot;5.0&quot;</td>
</tr>
<tr>
<td></td>
<td>wavescan.trace.dToAUnit</td>
<td>string</td>
<td>&quot;V&quot;</td>
</tr>
<tr>
<td></td>
<td>wavescan.trace.modifier</td>
<td>string</td>
<td>&quot;mag&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Marker Variables</th>
<th>Variable</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>wavescan.marker.foreground</td>
<td>string</td>
<td>&quot;black&quot;</td>
</tr>
<tr>
<td></td>
<td>wavescan.marker.arrowOn</td>
<td>string</td>
<td>&quot;true&quot;</td>
</tr>
<tr>
<td></td>
<td>wavescan.marker.fontName</td>
<td>string</td>
<td>&quot;Dialog&quot;</td>
</tr>
<tr>
<td></td>
<td>wavescan.marker.fontStyle</td>
<td>string</td>
<td>&quot;plain&quot;</td>
</tr>
<tr>
<td></td>
<td>wavescan.marker.fontSize</td>
<td>string</td>
<td>&quot;medium&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Graph Label Variables</th>
<th>Variable</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>wavescan.graphLabel.foreground</td>
<td>string</td>
<td>&quot;black&quot;</td>
</tr>
<tr>
<td></td>
<td>wavescan.graphLabel.direction</td>
<td>string</td>
<td>&quot;east&quot;</td>
</tr>
<tr>
<td></td>
<td>wavescan.graphLabel.significantDigits</td>
<td>string</td>
<td>&quot;3&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Calculator Variables</th>
<th>Variable</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>wavescan.calculator.rpnMode</td>
<td>string</td>
<td>&quot;true&quot;</td>
</tr>
<tr>
<td></td>
<td>wavescan.calculator.reportVarErrors</td>
<td>string</td>
<td>&quot;false&quot;</td>
</tr>
<tr>
<td></td>
<td>wavescan.calculator.stackSize</td>
<td>string</td>
<td>&quot;8&quot;</td>
</tr>
<tr>
<td></td>
<td>wavescan.calculator.xLocation</td>
<td>string</td>
<td>&quot;600&quot;</td>
</tr>
<tr>
<td></td>
<td>wavescan.calculator.yLocation</td>
<td>string</td>
<td>&quot;50&quot;</td>
</tr>
<tr>
<td></td>
<td>wavescan.calculator.width</td>
<td>string</td>
<td>&quot;640&quot;</td>
</tr>
<tr>
<td></td>
<td>wavescan.calculator.height</td>
<td>string</td>
<td>&quot;330&quot;</td>
</tr>
<tr>
<td></td>
<td>wavescan.calculator.significantDigits</td>
<td>string</td>
<td>&quot;4&quot;</td>
</tr>
<tr>
<td></td>
<td>wavescan.calculator.notation</td>
<td>string</td>
<td>&quot;suffix&quot;</td>
</tr>
<tr>
<td>Table Variables</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-------</td>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>wavescan.tableFrame width</td>
<td>string</td>
<td>&quot;570&quot;</td>
<td></td>
</tr>
<tr>
<td>wavescan.tableFrame height</td>
<td>string</td>
<td>&quot;590&quot;</td>
<td></td>
</tr>
<tr>
<td>wavescan.reportTable fileSuffix</td>
<td>string</td>
<td>&quot;tbl&quot;</td>
<td></td>
</tr>
<tr>
<td>wavescan.reportTable stringColWidth</td>
<td>string</td>
<td>&quot;10&quot;</td>
<td></td>
</tr>
<tr>
<td>wavescan.reportTable cellPadding</td>
<td>string</td>
<td>&quot;5&quot;</td>
<td></td>
</tr>
<tr>
<td>wavescan.reportTable maxColumnWidth</td>
<td>string</td>
<td>&quot;200&quot;</td>
<td></td>
</tr>
<tr>
<td>wavescan.reportTable notation</td>
<td>string</td>
<td>&quot;scientific&quot;</td>
<td></td>
</tr>
<tr>
<td>wavescan.reportTable location</td>
<td>string</td>
<td>&quot;cell&quot;</td>
<td></td>
</tr>
<tr>
<td>wavescan.reportTable significantDigits</td>
<td>string</td>
<td>&quot;4&quot;</td>
<td></td>
</tr>
</tbody>
</table>
Application Variables

**statefile**

Specifies the default file name for saving the current session. The default file name appears in the Save Session dialog box when you exit WaveScan.

**Syntax**

```plaintext
wavescan.application statefile string "file"
```

**Values**

- **file**
  - Name of file.
  - Default: `wavescan.xml`
readstate

Specifies whether WaveScan reads the saved statefile at startup.

Syntax

```
wavescan.application readstate string "true" | "false"
```

Values

- **true**  
  WaveScan reads the statefile at startup. This is the default value.

- **false**  
  WaveScan does not read the statefile at startup.
**writestate**

Specifies whether WaveScan writes a statefile when it exits. This sets the default value for the *Save* checkbox in the Save Session dialog box. If the `exitdialog` variable is set to false, WaveScan writes a statefile automatically.

**Syntax**

```
wavescan.application writestate string "true" | "false"
```

**Values**

- **true**
  - WaveScan writes a statefile when it exits.

- **false**
  - WaveScan does not write a statefile when it exits. This is the default value.
graphtemplatefile

Specifies the name of the graph template file to be loaded at startup.

**Syntax**

```
wavescan.application graphtemplatefile string "none"
```

**Values**

- `file`  
  Name of graph attributes template file to be loaded at startup. Default value is none.
exitdialog

Specifies whether the Save Session dialog box appears when a session ends.

Syntax

```
wavescan.application exitdialog string "true" | "false"
```

Values

- `true` Save Session dialog box appears when a session ends. This is the default value.
- `false` Save Session dialog box does not appear when a session ends. WaveScan saves your session if the writestate variable is `true`.

envdirectory

Specifies the directory in which WaveScan looks for the .cdsenv file.

Syntax

wavescan.application envdirectory string "directory"

Values

| directory | Directory in which WaveScan looks for the .cdsenv file. Default: ./ |
readDefaultVarFileOnStartup

Specifies whether the calculator loads the default variables file at startup.

**Syntax**

```
wavescan.calculator readDefaultVarFileOnStartup string "true" | "false"
```

**Values**

- **true**: Calculator loads the default variables file.
- **false**: Calculator does not load the default variables file. This is the default value.
writeDefaultVarFileOnExit

Specifies whether the calculator saves the variables in the current session to the default variables file when it exits.

Syntax

wavescan.calculator writeDefaultVarFileOnExit string "true" | "false"

Values

<table>
<thead>
<tr>
<th>true</th>
<th>Calculator saves the variables to the default variables file.</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>Calculator does not save the variables to the default variables file. This is the default value.</td>
</tr>
</tbody>
</table>
defaultVarFileName

Specifies the default file name for the variables file.

Syntax

```
wavescan.calculator defaultVarFileName string "file"
```

Values

<table>
<thead>
<tr>
<th>file</th>
<th>Name of variables file. Default: .wsvariables</th>
</tr>
</thead>
</table>
**printer**

Specifies the default printer.

**Syntax**

```
wavescan.application printer string "default_printer"
```

**Values**

- `default_printer` Name of the default printer.
Results Browser Variables

**historyLength**

Specifies the maximum number of dataset paths saved in the *Location* pull-down in the Results Browser window.

**Syntax**

```plaintext
wavescan.browser historyLength string "maxDirectories"
```

**Values**

- **maxDirectories**: Maximum number of data directories that fit in the *Location* field.
  - Default: 10
  - Valid values: 0–20
plotstyle

Specifies the default plotting style for a new graph. This sets the default value for the plot style pull-down in the top right corner of the Results Browser window.

Syntax

```
wavescan.browser plotStyle string "Append" | "Replace" | "New SubWin" | "New Win"
```

Values

- **append**: Appends the new graph to the current graph.
- **replace**: Replaces the current graph with the new graph. This is the default value.
- **newsub**: Plots the graph in a new subwindow.
- **newwin**: Plots the graph in a new window.
dataDirHome

Specifies the default directory for the Choose Data Directory dialog box.

Syntax

`wavescan.browser dataDirHome string "directory"

Values

`directory` Directory which the Choose Data Directory dialog box defaults to.
Default: ./
Filter Variable

textFilterType

Specifies the filter syntax for the Results Browser and Calculator windows.

Syntax

```
wavescan.filter textFilterType string "shell" | "regular"
```

Values

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>shell</td>
<td>Filter accepts shell syntax filter values. For example, l* displays all files beginning with l. This is the default value.</td>
</tr>
<tr>
<td>regular</td>
<td>Filter accepts regular expression syntax filter values. For example, l.* displays all files beginning with l, and l* displays all files.</td>
</tr>
</tbody>
</table>
Graph Frame Variables

**width**

Controls the width of the Graph Window.

**Syntax**

```
wavescan.graphFrame width string "width_pixels"
```

**Values**

- `width_pixels` Width of the graph window.
  - Default: 570
  - Valid values: A positive integer.
**height**

Controls the height of the Graph Window.

**Syntax**

```
wavescan.graphFrame height string "height_pixels"
```

**Values**

- **height_pixels**
  - Height of the graphwindow.
  - Default: 590
  - Valid values: A positive integer.
autoTraceSelect

Specifies whether WaveScan selects the trace closest to the system cursor.

Syntax

```
wavescan.graphFrame autoTraceSelect string "true" | "false"
```

Values

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>WaveScan automatically selects the trace closest to the system cursor. This is the default value.</td>
</tr>
<tr>
<td>false</td>
<td>WaveScan does not select a trace automatically.</td>
</tr>
</tbody>
</table>
rightMouseZoom

Specifies whether you can use the right mouse button to zoom your graph.

Syntax

\[\text{wavescan.graphFrame rightMouseZoom string } "\text{true}" \mid "\text{false}"\]

Values

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>Use the right mouse button to zoom your graph. This is the default value.</td>
</tr>
<tr>
<td>false</td>
<td>Use the middle mouse button to zoom your graph. This enables the right mouse pop-up menu in the Graph Window.</td>
</tr>
</tbody>
</table>
Rectangular Graph Variables

**background**

Controls the default background color in rectangular graphs. You can change this in the Graph Attributes dialog box.

**Syntax**

```
wavescan.rectGraph background string "background_color"
```

**Values**

```
background_color
```

Background color in rectangular graphs.  
Default: white  
Valid values: Colors available on the palette below.

The colors shown left to right, top to bottom are as follows.

<table>
<thead>
<tr>
<th>Color</th>
<th>Color</th>
<th>Color</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>white</td>
<td>lightGray</td>
<td>gray</td>
<td>darkGray</td>
</tr>
<tr>
<td>black</td>
<td>red</td>
<td>pink</td>
<td>mistyRose</td>
</tr>
<tr>
<td>orange</td>
<td>gold</td>
<td>lightTan</td>
<td>lightBrown</td>
</tr>
<tr>
<td>yellow</td>
<td>lightGoldenrod</td>
<td>lemonChiffon</td>
<td>blanchedAlmond</td>
</tr>
<tr>
<td>green</td>
<td>yellowGreen</td>
<td>olivedrab</td>
<td>honeydew</td>
</tr>
<tr>
<td>cyan</td>
<td>aquamarine</td>
<td>darkSeaGreen</td>
<td>cadetBlue</td>
</tr>
<tr>
<td>blue</td>
<td>greyBlue</td>
<td>lightSteelBlue</td>
<td>skyBlue</td>
</tr>
<tr>
<td>----------</td>
<td>------------</td>
<td>----------------</td>
<td>-----------</td>
</tr>
<tr>
<td>magenta</td>
<td>blueViolet</td>
<td>purple</td>
<td>violet</td>
</tr>
</tbody>
</table>
**foreground**

Controls the default foreground color (titles, grids, axis, labels, and text) in rectangular graphs. You can change this in the Graph Attributes dialog box.

**Syntax**

```plaintext
wavescan.rectGraph foreground string "foreground_color"
```

**Values**

- `foreground_color`: Foreground color in rectangular graphs.
- Default: black
- Valid values: Colors available on the palette below.

The colors shown left to right, top to bottom are as follows.

- white  lightGray  gray  darkGray
- black  red  pink  mistyRose
- orange  gold  lightTan  lightBrown
- yellow  lightGoldenrod  lemonChiffon  blanchedAlmond
- green  yellowGreen  olivedrab  honeydew
- cyan  aquamarine  darkSeaGreen  cadetBlue
- blue  greyBlue  lightSteelBlue  skyBlue
- magenta  blueViolet  purple  violet
visibleDigitalRows

Controls the maximum number of digital traces displayed in the graph window. A scrollable list is displayed after this number is exceeded.

Syntax

`wavescan.rectGraph visibleDigitalRows string "maxRows"`

Values

`maxRows`  
Maximum number of digital traces displayed in the graph window.  
Default: 3  
Valid values: A positive integer.
visibleStripChartRows

Controls the maximum number of strip charts displayed in the graph window. A scrollable list is displayed after this number is exceeded.

Syntax

wavescan.rectGraph visibleStripChartRows string "maxCharts"

Values

maxCharts Maximum number of strip charts displayed in the graph window. Default: 5
Valid values: A positive integer.
Circular Graph Variables

**background**

Controls the default background color in circular graphs. You can change this in the Graph Attributes dialog box.

**Syntax**

```
wavescan.circGraph background string "background_color"
```

**Values**

- **background_color**: Background color in circular graphs.
- **Default**: honeydew
- **Valid values**: Colors available on the palette below.

The colors shown left to right, top to bottom are as follows.

- white
- lightGray
- gray
- darkGray
- black
- red
- pink
- mistyRose
- orange
- gold
- lightTan
- lightBrown
- yellow
- lightGoldenrod
- lemonChiffon
- blanchedAlmond
- green
- yellowGreen
- olivedrab
- honeydew
- cyan
- aquamarine
- darkSeaGreen
- cadetBlue
blue    greyBlue    lightSteelBlue    skyBlue
magenta blueViolet    purple    violet
foreground

Controls the default foreground color (titles, grids, axis, labels, and text) in circular graphs. You can change this in the Graph Attributes dialog box.

Syntax

```
wavescan.circGraph foreground string "foreground_color"
```

Values

```
foreground_color
```
Foreground color in circular graphs.

Default: black

Valid values: Colors available on the palette below.

The colors shown left to right, top to bottom are as follows.

- white
- lightGray
- gray
- darkGray
- black
- red
- pink
- mistyRose
- orange
- gold
- lightTan
- lightBrown
- yellow
- lightGoldenrod
- lemonChiffon
- blanchedAlmond
- green
- yellowGreen
- olivedrab
- honeydew
- cyan
- aquamarine
- darkSeaGreen
- cadetBlue
- blue
- greyBlue
- lightSteelBlue
- skyBlue
- magenta
- blueViolet
- purple
- violet
Graph Variables

visibleLegendRows

Controls the maximum number of legend rows visible on the screen. A scrollable list is displayed after this number is exceeded.

Syntax

```
wavescan.graph visibleLegendRows string "row_number"
```

Values

```
row_number
```

Number of legend rows visible on the screen.
Default: 3
Valid values: A positive integer.
fileSuffix

Specifies the default suffix for saving graph and graph template files. This sets the value for the *Files of type* field in the Save Graph dialog box.

**Syntax**

```
wavescan.graph fileSuffix string "graph_suffix"
```

**Values**

- `graph_suffix`  
  Suffix for graph files.  
  Default: `grf`
**subTitle**

Specifies the default subtitle of the graph.

**Syntax**

```plaintext
draft.graph subTitle string "myGraph"
```

**Values**

```
myGraph  Subtitle of the graph.
```
displayDate

Specifies whether the current date is displayed on the graph.

Syntax

\texttt{wavescan.graph \textit{displayDate} string "$true$ \mid "$false$"

Values

\begin{tabular}{ll}
\texttt{true} & Displays the current date on the graph. This is the default value. \\
\texttt{false} & Does not display the date on the graph.
\end{tabular}
defaultSubtitle

Specifies whether the graph displays the default subtitle.

Syntax

\texttt{wavescan.graph defaultSubtitle string "true" | "false"}

Values

<table>
<thead>
<tr>
<th>true</th>
<th>Graph displays the default subtitle. Currently, this is blank. This is the default value.</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>Graph displays the subtitle you specified in the \texttt{subTitle} variable.</td>
</tr>
</tbody>
</table>
cursorOn

Specifies whether the trace cursor is on or off by default. You can change this setting through the Graph menu in the Graph Window (Graph–Cursor On).

Syntax

\texttt{wavescan.graph cursorOn string "true" \mid "false"}

Values

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>Cursor is displayed as you move the mouse over the trace.</td>
</tr>
<tr>
<td>false</td>
<td>Cursor is not displayed as you move the mouse over the trace. This is the default value.</td>
</tr>
</tbody>
</table>
**majGridsOn**

Specifies whether major grids are displayed by default in the graph window. You can change this setting in the Graph Attributes dialog box.

**Syntax**

```plaintext
wavescan.graph majGridsOn string "true" | "false"
```

**Values**

- **true**  
  Major grids are displayed in the graph window. This is the default value.

- **false**  
  Major grids are not displayed in the graph window.
**minGridsOn**

Specifies whether minor grids are displayed by default in the graph window. You can change this setting in the Graph Attributes dialog box.

**Syntax**

```
wavescan.graph minGridsOn string "true" | "false"
```

**Values**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>Minor grids are displayed in the graph window. This is the default value.</td>
</tr>
<tr>
<td>false</td>
<td>Minor grids are not displayed in the graph window.</td>
</tr>
</tbody>
</table>
**majorForeground**

Controls the default color of major grids. You can change this setting in the Graph Attributes dialog box.

**Syntax**

```
wavescan.graph majorForeground string "major_color"
```

**Values**

*major_color*  
Color of major grids.  
Default: *gray*  
Valid values: Colors available on the palette below.

The colors shown left to right, top to bottom are as follows.

- white  
- lightGray  
- gray  
- darkGray  
- black  
- red  
- pink  
- mistyRose  
- orange  
- gold  
- lightTan  
- lightBrown  
- yellow  
- lightGoldenrod  
- lemonChiffon  
- blanchedAlmond  
- green  
- yellowGreen  
- olivedrab  
- honeydew  
- cyan  
- aquamarine  
- darkSeaGreen  
- cadetBlue  
- blue  
- greyBlue  
- lightSteelBlue  
- skyBlue  
- magenta  
- blueViolet  
- purple  
- violet
**minorForeground**

Controls the default color of minor grids. You can change this setting in the Graph Attributes dialog box.

**Syntax**

```
wavescan.graph minorForeground string "minor_color"
```

**Values**

- **minor_color**: Color of minor grids.
- Default: lightGray
- Valid values: Colors available on the palette below.

The colors shown left to right, top to bottom are as follows.

```plaintext
white lightGray gray darkGray
black red pink mistyRose
orange gold lightTan lightBrown
yellow lightGoldenrod lemonChiffon blanchedAlmond
green yellowGreen olivedrab honeydew
cyan aquamarine darkSeaGreen cadetBlue
blue greyBlue lightSteelBlue skyBlue
magenta blueViolet purple violet
```
snapOn

Specifies whether the cursor and marker snap to data points by default. You can change the setting for the

cursor through the Graph – Snap-to-Data menu in the Graph Window.

marker in the Marker Attributes dialog box.

Syntax

```
wavescan.graph snapOn string "snapOff" | "snapToData" | "snapToPeak"
```

Values

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>snapOn</td>
<td>Cursor and marker snap to data points.</td>
</tr>
<tr>
<td>snapOff</td>
<td>Cursor and marker do not snap to data points. This is the default value.</td>
</tr>
<tr>
<td>snapToPeak</td>
<td>Cursor and marker snap to trace peaks.</td>
</tr>
</tbody>
</table>
fontName

Specifies the default font name for text in graphs. You can change this in the Graph Attributes dialog box.

Syntax

```
wavescan.graph fontName string "Dialog" | "SansSerif" | "Serif" | "Monospaced"
    | "DialogInput"
```

Values

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dialog</td>
<td>Font name of text is Dialog. This is the default value.</td>
</tr>
<tr>
<td>SansSerif</td>
<td>Font name of text is SansSerif.</td>
</tr>
<tr>
<td>Serif</td>
<td>Font name of text is Serif.</td>
</tr>
<tr>
<td>Monospaced</td>
<td>Font name of text is Monospaced.</td>
</tr>
<tr>
<td>DialogInput</td>
<td>Font name of text is DialogInput.</td>
</tr>
</tbody>
</table>
**fontStyle**

Specifies the default font style of text in graphs. You can change this in the Graph Attributes dialog box.

**Syntax**

```
wavescan.graph fontSize string "plain" | "italic" | "bold"
```

**Values**

- **plain**  
  Font style of text is plain. This is the default value.
- **italic**  
  Font style of text is italic.
- **bold**  
  Font style of text is bold.
**fontSize**

Specifies the default font size of text in graphs. You can change this in the Graph Attributes dialog box.

**Syntax**

```
wavescan.graph fontSize string "small" | "medium" | "large"
```

**Values**

- **small**
  Font size of text is small.

- **medium**
  Font size of text is medium. This is the default value.

- **large**
  Font size of text is large.
stripChartOn

Specifies whether the strip chart mode is on by default. You can change this through the Strip Chart Mode icon in the Graph Window.

Syntax

wavescan.graph stripChartOn string "true" | "false"

Values

true  Strip charts are displayed.
false Strip charts are not displayed. All traces are displayed in one graph window. This is the default value.
Axis Variables

**depAxis logScale**

Specifies whether logarithmic scale is on by default for the dependent axis (Y-axis). You can change this setting in the Axis Attributes dialog box.

**Syntax**

```plaintext
wavescan.depAxis logScale string "true" | "false"
```

**Values**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>Logarithmic scale is on.</td>
</tr>
<tr>
<td>false</td>
<td>Linear scale is on. This is the default value.</td>
</tr>
</tbody>
</table>
**depAxis forceOrigin**

Specifies whether the graph forces the origin to be included on the axis by default for the dependent axis (Y-axis). You can change this setting in the Axis Attributes dialog box.

**Syntax**

```plaintext
wavescan.depAxis forceOrigin string "true" | "false"
```

**Values**

- **true**: Graph originates from the beginning of the dependent axis.
- **false**: Graph does not originate from the beginning of the dependent axis. This is the default value.
**indepAxis logScale**

Specifies whether logarithmic scale is on by default for the independent axis (X-axis). You can change this setting in the Axis Attributes dialog box.

**Syntax**

```plaintext
wavescan.indepAxis logScale string "true" | "false"
```

**Values**

- `true` : Logarithmic scale is on.
- `false` : Linear scale is on. This is the default value.
**indepAxis forceOrigin**

Specifies whether the graph forces the origin to be included on the axis by default for the independent axis (X-axis). You can change this setting in the Axis Attributes dialog box.

**Syntax**

```wavescan.indepAxis forceOrigin string "true" | "false"```

**Values**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>Graph originates from the beginning of the independent axis.</td>
</tr>
<tr>
<td>false</td>
<td>Graph does not originate from the beginning of the independent axis. This is the default value.</td>
</tr>
</tbody>
</table>
foreground

Controls the default axis color in graphs. You can change this in the Axis Attributes dialog box.

Syntax

```plaintext
wavescan.axis foreground string "axis_color"
```

Values

<table>
<thead>
<tr>
<th><code>axis_color</code></th>
<th>Color of axis in graphs.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Default: black</td>
</tr>
<tr>
<td></td>
<td>Valid values: Colors available on the palette below.</td>
</tr>
</tbody>
</table>

The colors shown left to right, top to bottom are as follows.

- white
- lightGray
- gray
- darkGray
- black
- red
- pink
- mistyRose
- orange
- gold
- lightTan
- lightBrown
- yellow
- lightGoldenrod
- lemonChiffon
- blanchedAlmond
- green
- yellowGreen
- olivedrab
- honeydew
- cyan
- aquamarine
- darkSeaGreen
- cadetBlue
- blue
- greyBlue
- lightSteelBlue
- skyBlue
- magenta
- blueViolet
- purple
- violet
**significantDigits**

Controls the default number of significant digits displayed on the axis labels. You can change this in the Axis Attributes dialog box.

**Syntax**

```
wavescan.axis significantDigits string "sigDigits"
```

**Values**

- `sigDigits`  
  Number of significant digits displayed on the axis labels.
  Default: 3
  Valid values: 0–24
Circular Scale Variables

polarForeground

Controls the default color of the graph title and grids in polar graphs. You can change this in the Graph Attributes dialog box.

Syntax

```
wavescan.circularScale polarForeground string "polar_color"
```

Values

```
polar_color
```

Foreground color in polar graphs.
Default: gray
Valid values: Colors available on the palette below.

The colors shown left to right, top to bottom are as follows.

<table>
<thead>
<tr>
<th>Color</th>
<th>Color</th>
<th>Color</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>white</td>
<td>lightGray</td>
<td>gray</td>
<td>darkGray</td>
</tr>
<tr>
<td>black</td>
<td>red</td>
<td>pink</td>
<td>mistyRose</td>
</tr>
<tr>
<td>orange</td>
<td>gold</td>
<td>lightTan</td>
<td>lightBrown</td>
</tr>
<tr>
<td>yellow</td>
<td>lightGoldenrod</td>
<td>lemonChiffon</td>
<td>blanchedAlmond</td>
</tr>
<tr>
<td>green</td>
<td>yellowGreen</td>
<td>olivedrab</td>
<td>honeydew</td>
</tr>
<tr>
<td>cyan</td>
<td>aquamarine</td>
<td>darkSeaGreen</td>
<td>cadetBlue</td>
</tr>
</tbody>
</table>
WaveScan User Guide
WaveScan Variables

blue  greyBlue  lightSteelBlue  skyBlue
magenta  blueViolet  purple  violet
impedForeground

Controls the default color of the graph title and grids in impedance graphs. You can also specify this in the Graph Attributes dialog box.

Syntax

```
wavescan.circularScale impedForeground string "impedance_color"
```

Values

```
impedance_color
```

Foreground color in impedance graphs.

Default: gray

Valid values: Colors available on the palette below.

The colors shown left to right, top to bottom are as follows.

<table>
<thead>
<tr>
<th>white</th>
<th>lightGray</th>
<th>gray</th>
<th>darkGray</th>
</tr>
</thead>
<tbody>
<tr>
<td>black</td>
<td>red</td>
<td>pink</td>
<td>mistyRose</td>
</tr>
<tr>
<td>orange</td>
<td>gold</td>
<td>lightTan</td>
<td>lightBrown</td>
</tr>
<tr>
<td>yellow</td>
<td>lightGoldenrod</td>
<td>lemonChiffon</td>
<td>blanchedAlmond</td>
</tr>
<tr>
<td>green</td>
<td>yellowGreen</td>
<td>olivedrab</td>
<td>honeydew</td>
</tr>
<tr>
<td>cyan</td>
<td>aquamarine</td>
<td>darkSeaGreen</td>
<td>cadetBlue</td>
</tr>
<tr>
<td>blue</td>
<td>greyBlue</td>
<td>lightSteelBlue</td>
<td>skyBlue</td>
</tr>
<tr>
<td>magenta</td>
<td>blueViolet</td>
<td>purple</td>
<td>violet</td>
</tr>
</tbody>
</table>
**admittForeground**

Controls the default color of the graph title and grids in admittance graphs. You can also specify this in the Graph Attributes dialog box.

**Syntax**

```
wavescan.circularScale admittForeground string "admittance_color"
```

**Values**

- **admittance_color**
  
  Foreground color in admittance graphs.
  
  Default: gray
  
  Valid values: Colors available on the palette below.

The colors shown left to right, top to bottom are as follows.

- white
- lightGray
- gray
- darkGray
- black
- red
- pink
- mistyRose
- orange
- gold
- lightTan
- lightBrown
- yellow
- lightGoldenrod
- lemonChiffon
- blanchedAlmond
- green
- yellowGreen
- olivedrab
- honeydew
- cyan
- aquamarine
- darkSeaGreen
- cadetBlue
- blue
- greyBlue
- lightSteelBlue
- skyBlue
- magenta
- blueViolet
- purple
- violet
significantDigits

Controls the default number of significant digits displayed in circular graphs. You can change this setting in the Scale Attributes dialog box.

Syntax

```
wavescan.circularScale significantDigits string "sigDigits"
```

Values

```
sigDigits Number of significant digits displayed in circular graphs.
Default: 2
Valid values: 0–24
```
**cursorScale**

Specifies the default scaling for cursor coordinates. You can change this setting in the Scale Attributes dialog box.

**Syntax**

```
wavescan.circularScale cursorScale string "realimag" | "magphase" | "impedance"
 | "admittance" | "reflection"
```

**Values**

- **realimag**
  - Scaling is Real/Imaginary. This is the default value.
- **magphase**
  - Scaling is Magnitude/Phase.
- **impedance**
  - Scaling is Impedance.
- **admittance**
  - Scaling is Admittance.
- **reflection**
  - Scaling is Reflection.
Trace Variables

type

Specifies the default trace style. You can change this setting in the Trace Attributes dialog box.

Syntax

```plaintext
wavescan.trace type string "line" | "points" | "bars" | "spectral" | "histogram" | "eye"
```

Values

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>line</td>
<td>Trace is a line. This is the default value.</td>
</tr>
<tr>
<td>points</td>
<td>Trace is a series of points.</td>
</tr>
<tr>
<td>bars</td>
<td>Trace is a series of bars.</td>
</tr>
<tr>
<td>spectral</td>
<td>Trace is a spectral.</td>
</tr>
<tr>
<td>histogram</td>
<td>Trace is a histogram.</td>
</tr>
<tr>
<td>eye</td>
<td>Trace is an eye.</td>
</tr>
</tbody>
</table>
lineThickness

Controls the default trace line thickness. You can change this setting in the Trace Attributes dialog box.

Syntax

```
wavescan.trace lineThickness string "fine" | "medium" | "bold"
```

Values

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fine</td>
<td>Trace line is fine. This is the default value.</td>
</tr>
<tr>
<td>medium</td>
<td>Trace line is medium.</td>
</tr>
<tr>
<td>bold</td>
<td>Trace line is bold.</td>
</tr>
</tbody>
</table>
**lineStyle**

Specifies the default trace style. You can change this setting in the Trace Attributes dialog box.

**Syntax**

```plaintext
wavescan.trace lineStyle string "solid" | "dashed" | "dotted"
```

**Values**

- **solid**
  Trace is a solid line. This is the default value.

- **dashed**
  Trace is a dashed line.

- **dotted**
  Trace is a dotted line.
symbolsOn

Specifies whether symbols are displayed on data points on the trace by default. You can change this setting through the Trace menu in the Graph Window (Trace–Symbols On), or the Trace Attributes dialog box.

Syntax

```
wavescan.trace symbolsOn string "true" | "false"
```

Values

- **true**: Data points are displayed.
- **false**: Data points are not displayed. This is the default value.
symbolStyle

Specifies the default symbol for data points. You can change this setting in the Trace Attributes dialog box.

Syntax

```
wavescan.trace symbolStyle string "plus" | "point" | "dot" | "square" | "box"
   | "upArrow" | "downArrow"
```

Values

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>plus</td>
<td>Data points are indicated by a plus sign. This is the default value.</td>
</tr>
<tr>
<td>point</td>
<td>Data points are indicated by a point.</td>
</tr>
<tr>
<td>dot</td>
<td>Data points are indicated by a dot.</td>
</tr>
<tr>
<td>square</td>
<td>Data points are indicated by a square.</td>
</tr>
<tr>
<td>box</td>
<td>Data points are indicated by a box.</td>
</tr>
<tr>
<td>upArrow</td>
<td>Data points are indicated by an arrow pointing upwards.</td>
</tr>
<tr>
<td>downArrow</td>
<td>Data points are indicated by a an arrow pointing downwards.</td>
</tr>
</tbody>
</table>
symbolCount

Specifies the default number of data points displayed on a trace. You can also specify this in the Trace Attributes dialog box.

Syntax

```
wavescan.trace symbolCount string "All Points" | "20" | "10" | "5"
```

Values

| 20 | 20 data points are displayed on the trace. This is the default value. |
| 10 | 10 data points are displayed on the trace. |
| 5  | 5 data points are displayed on the trace. |
dToAConversion

Specifies the default value for the scale maximum when placing a digital trace in an analog graph – the value equivalent to logical 1.

Syntax

```
wavescan.trace dToAConversion "quantity"
```

Values

```
quantity
```
Quantity equivalent to the Y-axis value of a digital trace.
Default: 5
Valid value: A positive integer.
**dToAUnit**

Specifies the default scale unit when placing a digital trace in an analog graph.

**Syntax**

```
wavescan.trace fontSize string "unit"
```

**Values**

<table>
<thead>
<tr>
<th>unit</th>
<th>Unit of conversion.</th>
<th>Default: volts</th>
</tr>
</thead>
</table>
**modifier**

Specifies the X axis scale used to plot AC data on a rectangular graph. You can also specify this in the Results Browser window.

**Syntax**

```
wavescan.trace modifier string "Mag" | "Phase" | "WPhase" | "Real" | "Imag" | "dB10" | "dB20"
```

**Values**

- **Mag**
  - Plots magnitude versus frequency. This is the default value.
- **Phase**
  - Plots phase versus frequency.
- **WPhase**
  - Plots wrapped phase versus frequency.
- **Real**
  - Plots real versus frequency.
- **Imag**
  - Plots imaginary versus frequency.
- **dB10**
  - Plots dB10 versus frequency.
- **dB20**
  - Plots dB20 versus frequency.
Marker Variables

foreground

Controls the marker color. You can change this setting in the Marker Attributes dialog box.

Syntax

```
wavescan.marker foreground string "marker_color"
```

Values

```
marker_color
Marker color.
Default: black
Valid values: Colors available on the palette below.
```

The colors shown left to right, top to bottom are as follows.

<table>
<thead>
<tr>
<th>color</th>
<th>color</th>
<th>color</th>
<th>color</th>
</tr>
</thead>
<tbody>
<tr>
<td>white</td>
<td>lightGray</td>
<td>gray</td>
<td>darkGray</td>
</tr>
<tr>
<td>black</td>
<td>red</td>
<td>pink</td>
<td>mistyRose</td>
</tr>
<tr>
<td>orange</td>
<td>gold</td>
<td>lightTan</td>
<td>lightBrown</td>
</tr>
<tr>
<td>yellow</td>
<td>lightGoldenrod</td>
<td>lemonChiffon</td>
<td>blanchedAlmond</td>
</tr>
<tr>
<td>green</td>
<td>yellowGreen</td>
<td>olivedrab</td>
<td>honeydew</td>
</tr>
<tr>
<td>cyan</td>
<td>aquamarine</td>
<td>darkSeaGreen</td>
<td>cadetBlue</td>
</tr>
<tr>
<td>blue</td>
<td>greyBlue</td>
<td>lightSteelBlue</td>
<td>skyBlue</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>----------------</td>
<td>---------</td>
</tr>
<tr>
<td>magenta</td>
<td>blueViolet</td>
<td>purple</td>
<td>violet</td>
</tr>
</tbody>
</table>
arrowOn

Specifies whether an arrow is displayed between a marker and its label. You can change this setting in the Marker Attributes dialog box.

Syntax

`wavescan.marker arrowOn string "true" | "false"`

Values

<table>
<thead>
<tr>
<th>true</th>
<th>Arrow is displayed between the data point and marker text. This is the default value.</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>Arrow is not displayed between the data point and marker text.</td>
</tr>
</tbody>
</table>
fontName

Specifies the marker font. This overrides the value you specified in the graph fontName variable.

You can change this setting in the Marker Attributes dialog box.

Syntax

```
wavescan.marker fontName string "Dialog" | "SansSerif" | "Serif" | "Monospaced" |
                          "DialogInput"
```

Values

- **Dialog**
  - Font name of markers is Dialog. This is the default value.
- **SansSerif**
  - Font name of markers is SansSerif.
- **Serif**
  - Font name of markers is Serif.
- **Monospaced**
  - Font name of markers is Monospaced.
- **DialogInput**
  - Font name of markers is DialogInput.
**fontStyle**

Specifies the font style for markers. This overrides the value you specified in the graph `fontStyle` variable.

You can change this setting in the Marker Attributes dialog box.

**Syntax**

```
wavescan.marker fontStyle string "plain" | "italic" | "bold"
```

**Values**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>plain</td>
<td>Font style of markers is plain. This is the default value.</td>
</tr>
<tr>
<td>italic</td>
<td>Font style of markers is italic.</td>
</tr>
<tr>
<td>bold</td>
<td>Font style of markers is bold.</td>
</tr>
</tbody>
</table>
**fontSize**

Controls the font size of markers. This overrides the value you specified in the graph `fontSize` variable.

You can change this setting in the Marker Attributes dialog box.

**Syntax**

```
wavescan.marker fontSize string "small" | "medium" | "large"
```

**Values**

- **small**
  - Font size of markers is small.

- **medium**
  - Font size of markers is medium. This is the default value.

- **large**
  - Font size of markers is large.
Graph Label Variables

foreground

Controls the color of graph labels. You can change this setting in the Label Attributes dialog box.

Syntax

\texttt{wavescan.graphLabel \textit{foreground} string "label\_color"}

Values

\begin{verbatim}
\begin{tabular}{ll}
\textit{label\_color} & Color of graph labels. \\
\textbf{Default}: \texttt{black} & Valid values: Colors available on the palette below.
\end{tabular}
\end{verbatim}

The colors shown left to right, top to bottom are as follows.

\begin{verbatim}
white    lightGray    gray     darkGray  
black    red         pink     mistyRose  
orange   gold        lightTan lightBrown  
yellow   lightGoldenrod lemonChiffon blanchedAlmond  
green    yellowGreen olivedrab honeydew  
cyan     aquamarine darkSeaGreen cadetBlue  
\end{verbatim}
<table>
<thead>
<tr>
<th>blue</th>
<th>greyBlue</th>
<th>lightSteelBlue</th>
<th>skyBlue</th>
</tr>
</thead>
<tbody>
<tr>
<td>magenta</td>
<td>blueViolet</td>
<td>purple</td>
<td>violet</td>
</tr>
</tbody>
</table>
direction

Specifies the default direction for graph labels. You can change this setting in the Label Attributes dialog box.

Syntax

```
wavescan.graphLabel direction string "east" | "west" | "north" | "south"
```

Values

- **east**: Graph label faces east. This is the default value.
- **west**: Graph label faces west.
- **north**: Graph label faces north.
- **south**: Graph label faces south.

Description

**East**: Text is left to right. Label looks like the figure below.

```
ABC
```

**West**: Text is right to left. Label looks like the figure below.

```
ABC
```

**North**: Label looks like the figure below.

```
ABC
```

**South**: Label looks like the figure below.

```
ABC
```
significantDigits

Controls the default number of significant digits displayed in graph labels. You can change this setting in the Label Attributes dialog box.

Syntax

```
wavescan.graphLabel significantDigits string "sigDigits"
```

Values

`sigDigits` Number of significant digits displayed in graph labels.
Default: 3
Valid values: 0–24
Calculator Variables

rpnMode

Specifies whether the Calculator is in the rpn mode by default. You can change this setting through the Options menu in the Calculator window (Options–Set RPN).

Syntax

```plaintext
wavescan.calculator rpnMode string "true" | "false"
```

Values

- **true**: Calculator is in rpn mode. This is the default value.
- **false**: Calculator is in algebraic mode.
reportVarErrors

Specifies whether WaveScan displays validation errors by default while you are creating variables.

Syntax

wavescan.calculator reportVarErrors string "true" | "false"

Values

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>Calculator displays validation errors.</td>
</tr>
<tr>
<td>false</td>
<td>Calculator does not display validation errors.</td>
</tr>
<tr>
<td></td>
<td>This is the default value.</td>
</tr>
</tbody>
</table>
stackSize

Controls the maximum number of expressions displayed in the Calculator stack. A scrollable list is displayed after this number is exceeded.

Syntax

```
wavescan.calculator stackSize string "stack_number"
```

Values

<table>
<thead>
<tr>
<th>stack_number</th>
<th>Number of expressions displayed in the Calculator.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default: 8</td>
<td>Valid values: 0–20</td>
</tr>
</tbody>
</table>
**xLocation**

Controls the position where the Calculator window appears.

**Syntax**

```
wavescan.calculator xLocation string "x_position"
```

**Values**

- `x_position`  
  Horizontal distance of the Calculator window from the left of the screen.  
  Default: 600  
  Valid values: A positive integer.
yLocation

Controls the position of the Calculator window.

Syntax

```
wavescan.calculator yLocation string "y_position"
```

Values

```
y_position
```

Vertical distance of the Calculator window from the top of the screen.
Default: 50
Valid values: A positive integer.
width

Controls the width of the calculator window.

Syntax

\texttt{wavescan\.calculator width string "width\_pixels"}

Values

\begin{tabular}{|l|l|}
\hline
\textit{width\_pixels} & Width of the Calculator window. \\
\hline
Default: 640 & \\
Valid values: A positive integer greater than 640. \\
\hline
\end{tabular}
height

Controls the height of the Calculator window.

Syntax

```
wavescan.calculator height string "height_pixels"
```

Values

- `height_pixels` Height of the Calculator window.
  - Default: 330
  - Valid values: A positive integer greater than 330.
significantDigits

Controls the number of significant digits displayed in the Calculator window.

Syntax

`wavescan.calculator significantDigits string "sigDigits"`

Values

<table>
<thead>
<tr>
<th>sigDigits</th>
<th>Number of significant digits displayed in the Calculator window.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default: 4</td>
<td>Valid values: 0–24</td>
</tr>
</tbody>
</table>
**notation**

Specifies the default notation for expression results in the Calculator.

**Syntax**

```
wavescan.calculator notation string "suffix" | "engineering" | "scientific"
```

**Values**

- **suffix**
  
  Session uses the suffix notation. This is the default value.

- **engineering**
  
  Session uses the engineering notation.

- **scientific**
  
  Session uses the scientific notation.
Table Variables

width

Specifies the initial width of the Report Table window.

Syntax

```
wavescan.tableFrame width string "width_pixels"
```

Values

- `width_pixels`: Width of the Report Table window.
  - Default: 570
  - Valid values: A positive integer.
**height**

Specifies the initial height of the Report Table window.

**Syntax**

```
wavescan.tableFrame height string "height_pixels"
```

**Values**

- **height_pixels**: Height of the table frame.
  - Default: 590
  - Valid values: A positive integer.
fileSuffix

Specifies the default suffix for saving table files. This sets the value for the *Files of type* field in the Save Table dialog box.

Syntax

```
wavescan.reportTable fileSuffix string "table_suffix"
```

Values

```
table_suffix
```

Suffix for table files.

Default: tbl
stringColumnWidth

Specifies the minimum column width of a table cell.

Syntax

```
wavescan.reportTable stringColWidth string "min_size"
```

Values

```
min_size
Minimum cell size.
Default: 10
```
**cellPadding**

Cell Padding is used to determine the table cell size.

WaveScan adds `cellPadding` to the table title: if the sum is more than the `stringColumnWidth`, WaveScan uses the sum for the cell size; if the sum is less than the `stringColumnWidth`, WaveScan uses the `stringColumnWidth` for the cell size.

**Syntax**

```
wavescan.reportTable cellPadding string "cell_distance"
```

**Values**

`cell_distance`  
The cell padding which is added to the table title to determine the table cell size.  
Default: 5  
Valid values: 0–15
maxColumnWidth

Controls the maximum width of a column.

Syntax

```
wavescan.reportTable maxColumnWidth string "maxWidth"
```

Values

```
maxWidth
```

Maximum width of a column.

Default: 200

Valid values: 0–500
### notation

Specifies the table notation.

#### Syntax

```
wavescan.reportTable notation string "scientific" | "engineering" | "suffix"
```

#### Values

- **scientific**: Table displays data in the scientific notation. This is the default value.
- **engineering**: Table displays data in the engineering notation.
- **suffix**: Table displays data in the suffix notation.
location

Specifies the default location for scale formatting in tables. This sets the value for the Active Format Location field in the Format Attributes dialog box.

Syntax

`wavescan.reportTable location string "cell" | "header"

Values

cell Displays numbers independently in each cell. This is the default value.

header Factors out the base quantity and displays it as the header.
significantDigits

Controls the number of significant digits displayed in a table cell.

**Syntax**

```
wavescan.reportTable significantDigits string "sigDigits"
```

**Values**

- `sigDigits`: Maximum number of significant digits displayed in a table cell.
  - Default: 4
  - Valid values: 0–24
Shell Environment Variables

This appendix describes the UNIX shell environment variables for WaveScan.
CDS_WAVESCAN_JRE_DIR

Sets the JRE directory path. If you do not set this variable, WaveScan runs with JRE 1.4, which is the Java version that ships with Cadence releases.

You can set the JRE root directory in one of the following ways:

- If you want the environment variable to be valid for just one WaveScan session, type the following in a terminal window:
  ```
  env CDS_WAVESCAN_JRE_DIR=jre14_root_dir wavescan
  ```

- If you want the environment variable to be valid for all WaveScan sessions launched from a shell, type the following in a terminal window:
  ```
  setenv CDS_WAVESCAN_JRE_DIR jre14_root_dir
  ```
  Press enter and then type `wavescan` in the same window.

- If you want the environment variable to be valid for all WaveScan sessions, type the following in your `.cshrc` file:
  ```
  setenv CDS_WAVESCAN_JRE_DIR jre14_root_dir
  ```
  Then type `wavescan` in a terminal window.
CDS_WAVESCANN_MAXHEAP

Sets the maximum heap amount for WaveScan in the Java Virtual Machine (JVM). The default heap size is 1024M. You must specify this variable in megabytes, and you must type the M. For example, 3072M.

WaveScan can expand in size till it reaches the maximum heap size specified by this variable. You should set this environment variable to take maximum advantage of your resources. If the sum of your virtual memory resources, physical RAM, and swap area is significantly greater than 1024M, you can reset the maximum heap size accordingly. Also, if the sum of your system resources is smaller than 1024M, you must reset the heapsize or you might get a memory resource error at startup.

If you run out of memory any time during a WaveScan session, the following message is displayed in the terminal window:

Exception occurred during event dispatching:
java.lang.OutOfMemoryError
# Accelerator Keys

## Graph Window

The following table describes the accelerator keys for the Graph Window.

<table>
<thead>
<tr>
<th>Key</th>
<th>Menu Option</th>
<th>Usage and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>c</td>
<td>Trace – Trace Cursor</td>
<td>Turns trace cursor on or off.</td>
</tr>
<tr>
<td>v</td>
<td>Trace – Vert Cursor</td>
<td>Turns vertical cursor on or off.</td>
</tr>
<tr>
<td>h</td>
<td>Trace – Horiz Cursor</td>
<td>Turns horizontal cursor on or off.</td>
</tr>
<tr>
<td>d</td>
<td>Trace – Delta Cursor</td>
<td>Turns delta cursor on or off.</td>
</tr>
<tr>
<td>m</td>
<td>Marker – Place – Trace Marker</td>
<td>Places trace marker.</td>
</tr>
<tr>
<td>a</td>
<td>Marker – Add Delta</td>
<td>Places delta marker.</td>
</tr>
<tr>
<td>V</td>
<td>Marker – Place – Vert Marker</td>
<td>Places vertical marker.</td>
</tr>
<tr>
<td>H</td>
<td>Marker – Place – Horiz Marker</td>
<td>Places horizontal marker.</td>
</tr>
<tr>
<td>z</td>
<td>Zoom – Zoom</td>
<td>Displays zoom rectangle.</td>
</tr>
<tr>
<td>x</td>
<td>Zoom – X-Zoom</td>
<td>Displays the cursor indicating that you can zoom along the X-axis.</td>
</tr>
<tr>
<td>y</td>
<td>Zoom – Y-Zoom</td>
<td>Displays the cursor indicating that you can zoom along the Y-axis.</td>
</tr>
<tr>
<td>i</td>
<td>Zoom – ZoomIn</td>
<td>Zooms in.</td>
</tr>
<tr>
<td>o</td>
<td>Zoom – ZoomOut</td>
<td>Zooms out.</td>
</tr>
<tr>
<td>f</td>
<td>Zoom – Fit</td>
<td>Expands the graph such that the entire trace is displayed.</td>
</tr>
</tbody>
</table>
Accelerator Keys

The following table describes the accelerator keys for the Calculator window.

<table>
<thead>
<tr>
<th>Key</th>
<th>Menu Option</th>
<th>Usage and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>u</td>
<td>Zoom – Unzoom</td>
<td>Incrementally reverses a series of zooms.</td>
</tr>
<tr>
<td>l</td>
<td></td>
<td>Toggles the selected axis between linear and log scale.</td>
</tr>
<tr>
<td>Alt</td>
<td></td>
<td>Select a trace and hold down the Alt key to display the trace information.</td>
</tr>
</tbody>
</table>

Calculator

The following table describes the accelerator keys for the Calculator window.

<table>
<thead>
<tr>
<th>Key</th>
<th>Menu Option</th>
<th>Usage and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ctrl-N</td>
<td>Var – Table – New Variable</td>
<td>Displays the Variable Editor so that you can create a variable for the expression.</td>
</tr>
<tr>
<td>Ctrl-C</td>
<td>Var – Table – Copy</td>
<td>Copies the selected variable.</td>
</tr>
<tr>
<td>Ctrl-D</td>
<td>Var – Table – Delete</td>
<td>Deletes the selected variable.</td>
</tr>
</tbody>
</table>
Calculator Functions

This chapter describes the functions in the function panel for both the SKILL and MDL modes.

Table D-1 Functions in the SKILL Mode

<table>
<thead>
<tr>
<th>Function</th>
<th>Table</th>
<th>Table</th>
<th>Table</th>
<th>Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/x</td>
<td>cos</td>
<td>ga</td>
<td>harmonicFreq</td>
<td></td>
</tr>
<tr>
<td>10**x</td>
<td>cosh</td>
<td>gac_freq</td>
<td>iinteg</td>
<td></td>
</tr>
<tr>
<td>abs</td>
<td>cross</td>
<td>gac_gain</td>
<td>imag</td>
<td></td>
</tr>
<tr>
<td>acos</td>
<td>dB10</td>
<td>gainBwProd</td>
<td>int</td>
<td></td>
</tr>
<tr>
<td>acosh</td>
<td>dB20</td>
<td>gainMargin</td>
<td>integ</td>
<td></td>
</tr>
<tr>
<td>asin</td>
<td>dBm</td>
<td>getAsciiWave</td>
<td>ipn</td>
<td></td>
</tr>
<tr>
<td>asinh</td>
<td>delay</td>
<td>gmax</td>
<td>ipnVRI</td>
<td></td>
</tr>
<tr>
<td>atan</td>
<td>deriv</td>
<td>gmin</td>
<td>kf</td>
<td></td>
</tr>
<tr>
<td>ananh</td>
<td>dft</td>
<td>gmsg</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>average</td>
<td>dftbb</td>
<td>gp</td>
<td>log10</td>
<td></td>
</tr>
<tr>
<td>b1f</td>
<td>exp</td>
<td>gpc_freq</td>
<td>lsb</td>
<td></td>
</tr>
<tr>
<td>bandwidth</td>
<td>eyeDiagram</td>
<td>gpc_gain</td>
<td>lshift</td>
<td></td>
</tr>
<tr>
<td>clip</td>
<td>flip</td>
<td>groupDelay</td>
<td>mag</td>
<td></td>
</tr>
<tr>
<td>compression</td>
<td>foo</td>
<td>gt</td>
<td>nc_freq</td>
<td></td>
</tr>
<tr>
<td>compression VRI</td>
<td>fourEval</td>
<td>gumx</td>
<td>nc_gain</td>
<td></td>
</tr>
<tr>
<td>convolve</td>
<td>frequency</td>
<td>harmonic</td>
<td>nf</td>
<td></td>
</tr>
<tr>
<td>nfmin</td>
<td>s11</td>
<td>tan</td>
<td>gt</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
<td>------</td>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>overshoot</td>
<td>s12</td>
<td>tangent</td>
<td>gumx</td>
<td></td>
</tr>
<tr>
<td>phaseMargin</td>
<td>s21</td>
<td>tanh</td>
<td>harmonic</td>
<td></td>
</tr>
<tr>
<td>phaseNoise</td>
<td>s22</td>
<td>thd</td>
<td>harmonicFreq</td>
<td></td>
</tr>
<tr>
<td>psd</td>
<td>sample</td>
<td>value</td>
<td>iinteg</td>
<td></td>
</tr>
<tr>
<td>psdbb</td>
<td>settlingTime</td>
<td>x**2</td>
<td>imag</td>
<td></td>
</tr>
<tr>
<td>real</td>
<td>sin</td>
<td>xmax</td>
<td>int</td>
<td></td>
</tr>
<tr>
<td>riseTime</td>
<td>sinh</td>
<td>xmin</td>
<td>integ</td>
<td></td>
</tr>
<tr>
<td>rms</td>
<td>slewRate</td>
<td>xval</td>
<td>ipn</td>
<td></td>
</tr>
<tr>
<td>rmsNoise</td>
<td>spectralPower</td>
<td>y**x</td>
<td>ipnVRI</td>
<td></td>
</tr>
<tr>
<td>rn</td>
<td>sqrt</td>
<td>ymax</td>
<td>kf</td>
<td></td>
</tr>
<tr>
<td>Rn</td>
<td>ssb</td>
<td>ymin</td>
<td></td>
<td></td>
</tr>
<tr>
<td>root</td>
<td>stddev</td>
<td>groupDelay</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table D-2 Functions in the MDL Mode

<table>
<thead>
<tr>
<th>Function</th>
<th>Function</th>
<th>Function</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>abs</td>
<td>crosses</td>
<td>imag</td>
<td>rms</td>
</tr>
<tr>
<td>acos</td>
<td>d2r</td>
<td>int</td>
<td>round</td>
</tr>
<tr>
<td>acosh</td>
<td>db</td>
<td>integ</td>
<td>sample</td>
</tr>
<tr>
<td>angle</td>
<td>dB10</td>
<td>ln</td>
<td>settlingTime</td>
</tr>
<tr>
<td>argmax</td>
<td>dBm</td>
<td>log10</td>
<td>sign</td>
</tr>
<tr>
<td>argmin</td>
<td>deltax</td>
<td>mag</td>
<td>sin</td>
</tr>
<tr>
<td>asin</td>
<td>deriv</td>
<td>max</td>
<td>sinh</td>
</tr>
<tr>
<td>asinh</td>
<td>dutycycle</td>
<td>min</td>
<td>slewRate</td>
</tr>
<tr>
<td>atan</td>
<td>dutycycles</td>
<td>mod</td>
<td>snr</td>
</tr>
<tr>
<td>atanh</td>
<td>exp</td>
<td>movingavg</td>
<td>sqrt</td>
</tr>
<tr>
<td>avg</td>
<td>falltime</td>
<td>overshoot</td>
<td>stathisto</td>
</tr>
<tr>
<td>bw</td>
<td>fft</td>
<td>period_jitter</td>
<td>tan</td>
</tr>
<tr>
<td>ceil</td>
<td>flip</td>
<td>ph</td>
<td>tanh</td>
</tr>
<tr>
<td>cfft</td>
<td>floor</td>
<td>phaseMargin</td>
<td>window</td>
</tr>
<tr>
<td>clip</td>
<td>frequency</td>
<td>pow</td>
<td>xval</td>
</tr>
<tr>
<td>conj</td>
<td>freq_jitter</td>
<td>pp</td>
<td>yval</td>
</tr>
<tr>
<td>convolve</td>
<td>gainBwProd</td>
<td>psd</td>
<td></td>
</tr>
<tr>
<td>cos</td>
<td>gainMargin</td>
<td>pzbode</td>
<td></td>
</tr>
<tr>
<td>cosh</td>
<td>groupdelay</td>
<td>pzfilter</td>
<td></td>
</tr>
<tr>
<td>cplx</td>
<td>histo</td>
<td>r2d</td>
<td></td>
</tr>
<tr>
<td>cross</td>
<td>ifft</td>
<td>real</td>
<td></td>
</tr>
<tr>
<td>crosscorr</td>
<td>iinteg</td>
<td>riseTime</td>
<td></td>
</tr>
</tbody>
</table>
1/x

Returns the inverse value. This function is available only in the SKILL mode.
10**x

Returns the $10^x$ value. This function is available only in the SKILL mode.
abs

Returns the absolute value of a signal.
acos

Returns the arc cosine of a signal.
acosh

Returns the hyperbolic arc cosine of a signal.
angle

Returns the angle of a complex number in degrees. This function is available only in the MDL mode.
**argmax**

Returns the X value corresponding to the maximum Y value of a signal. This function is available only in the MDL mode.

**Example**

![Graph showing argmax function with Maximum Y value and V(out) labels.]
argmin

Returns the X value corresponding to the minimum Y value of a signal. This function is available only in the MDL mode.

Example
**asin**

Returns the arc sine of a signal.
asinh

Returns the hyperbolic arc sine of a signal.
atan

Returns the arc tangent of a signal.
atanh

Returns the hyperbolic arc tangent of a signal.
average

Returns the average value of a signal.
b1f

Returns the stability factor b1f. This function is available only in the SKILL mode.
**bw (bandwidth)**

Calculates the bandwidth of a waveform.

- **Signal** is the name of the signal.
- **Db** is the decibels down from the peak i.e. how far below the peak value you want to see data.
- **Response** is the response type.
  
  - When 'low, computes the low-pass bandwidth by determining the smallest frequency at which the magnitude of the input waveform drops Db decibels below the DC gain. DC gain is obtained by zero-order extrapolation from the lowest or highest computed frequency, if necessary. An error occurs if the magnitude of the input waveform does not drop Db decibels below the DC gain.

  ![Diagram](image)

  - When 'high, computes the high-pass bandwidth by determining the largest frequency at which the magnitude of the input waveform drops db decibels below the gain at the highest frequency in the response waveform. An error occurs if the...
magnitude of the input waveform does not drop n decibels below the gain at high frequency.

- When 'band, computes the band-pass bandwidth by:
  - a. Determining the lowest frequency \( f_{max} \) at which the magnitude of the input waveform is maximized
  - b. Determining the highest frequency less than \( f_{max} \) at which the input waveform magnitude drops \( Db \) decibels below the maximum
  - c. Determining the lowest frequency greater than \( f_{max} \) at which the input waveform magnitude drops \( Db \) decibels below the maximum
  - d. Subtracting the value returned by step b from the value returned by step c. The value returned by step b or step c must exist.
ceil

Rounds a real number up to the closest integer value. This function is available only in the MDL mode.
cfft

Performs a Fast Fourier Transform on a complex time domain waveform and returns its frequency spectrum. The `cfft` function takes two time signals that in combination form a complex input signal. Available only in MDL mode.

- `sig_re` is the real part of the signal.
- `sig_im` is the imaginary part of the signal.
- `from` is the starting X value.
- `to` is the ending Y value.
- `numPoints` is the number of data points to be used for calculating the cfft. If this number is not a power of 2, it is automatically raised to the next higher power of 2.
- `window` is the algorithm used for calculating the cfft. In this release, only one algorithm is supported.
- `smoothing` is not supported in this release.
clip

Returns the portion of a signal between two points along the X-axis. You can use the clip function to restrict the range of action of other special functions of the calculator such as integ, rms, and frequency.

- **Signal** is the name of the signal.
- **From** is the starting point on the X-axis.
- **To** is the ending point on the Y-axis.

**Example**

The following input signal

\[
\text{signal} = V(\text{sinewave}), \quad \text{From} = 0, \quad \text{To} = 2.5
\]
is transformed into the following output signal.
**compression**

Returns the $N$th compression point value of a waveform at the specified extrapolation point. This function is available only in the SKILL mode.

- *Signal* is the name of the signal.
- *Harm Num* is the harmonic number.
- *Ext. Point (X)* is the extrapolation point of the waveform. The extrapolation point is the X-axis value.
- *Compression dB* specifies the compression coefficient (N).
compressionVRI

Performs an $n$th compression point measurement on a power waveform. This function is available only in the SKILL mode.

- $Signal$ is the name of the signal.
- $Harm\ Num$ is the harmonic index of the waveform.
- $Ext.\ Point\ (X)$ is the extrapolation point for the waveform. The default value is the minimum $x$ value of the input voltage waveform. The extrapolation point is the coordinate value in dBm that indicates the point on the output power waveform where the constant-slope power line begins. This point should be in the linear region of operation.
- $Load\ Resistance$ is the resistance. The default value is 50
- $Compression\ dB$ specifies the delta (in dB) between the power waveform and the ideal gain line that marks the compression point. The default value is 1.
**conj**

Returns the conjugate of a complex number. This function is available only in the MDL mode.
convolve

Returns a waveform consisting of the time domain convolution of two signals. sig1 is the name of the first signal.

- sig2 is the name of the second signal.
- n_interp_steps is the number of steps for interpolating waveforms.

Equation

Convolution is defined by the following equation:

\[ \int_{\text{from}}^{\text{to}} f_1(s) f_2(t-s) ds \]
COS

Returns the cosine of a signal.
cosh

Returns the hyperbolic cosine of a signal.
cplx

Returns a complex number created from two real arguments. This function is available only in the MDL mode

- $R$ is the value representing the real part.
- $I$ is the value representing the imaginary part.
wave

Returns the X value where a signal crosses the threshold Y value.

In the SKILL mode,
- **Signal** is the name of the signal.
- **Threshold Value** is the threshold to be crossed.
- **Edge Number** is the occurrence of the crossing. The first crossing is **Edge Number**=1, the second crossing is **Edge Number**=2, and so on. The value of **Edge Number** can be negative numbers: **Edge Number**=−1 for the previous occurrence, **Edge Number**=−2 for the occurrence before the previous occurrence, and so on.
- **Edge Type** is the direction of the crossing event. **rising** directs the function to look for crossings where the Y value is increasing, **falling** for crossings where the Y value is decreasing, and **either** for crossings in either direction.

In the MDL mode,
- **sig** is the name of the signal.
- **dir** is the direction of the crossing event. **rise** directs the function to look for crossings where the Y value is increasing, **fall** for crossings where the Y value is decreasing, and **cross** for crossings in either direction.
- **n** is the occurrence of the crossing. The first crossing is **n**=1, the second crossing is **n**=2, and so on. The value of **n** can be negative numbers: **n**=−1 for the previous occurrence, **n**=−2 for the occurrence before the previous occurrence, and so on.
- **thresh** is the threshold to be crossed.
- **start** is the time at which the function is enabled.
- **xtol** is the absolute tolerance in the X direction.
- **ytol** is the absolute tolerance in the Y direction.
- **accuracy** specifies that the function uses interpolation in the SKILL mode.

In the MDL mode, **accuracy** specifies whether the function should use interpolation, or use iteration controlled by the absolute tolerances to calculate the value. **interp** directs the function to use interpolation, and **exact** directs the function to consider the xtol and yval values.
Example

The following diagram illustrates how the result is determined for the values
\( signal = V(\text{out1}) \), \( Threshold\ Value = 1 \), \( Edge\ Number = 1 \), and \( Edge\ Type = \text{falling} \)
**crosscorr**

Returns the cross correlation of the specified signals. This function is available only in the MDL mode.

When the input signals are double waveforms,

\[ \text{crosscor} \left( \text{sig1, sig2} \right) = \texttt{convolve} \left( \text{sig1, flip(sig2)} \right) \]

When one of the input signals is a complex waveform (\( \text{sig2} \) in the following case),

\[ \text{crosscor} \left( \text{sig1, sig2} \right) = \texttt{convolve} \left( \text{sig1, flip(conj(sig2))} \right) \]

- \( \text{sig1} \) is the name of the first signal.
- \( \text{sig2} \) is the name of the second signal.
- \( n\_\text{interp\_steps} \) is the number of steps for interpolating waveforms.
crosses

Returns the X values where a signal crosses the threshold Y value. This function is available only in the MDL mode.

- \textit{sig} is the name of the signal.
- \textit{dir} is the direction of the crossing event. ‘\textit{rise}’ directs the function to look for crossings where the Y value is increasing, ‘\textit{fall}’ for crossings where the Y value is decreasing, and ‘\textit{cross}’ for crossings in either direction.
- \textit{n} is the occurrence of the crossing. If \(n=1\), the function returns the first crossing and all subsequent crossings. If \(n=3\), the function returns the third crossing and all subsequent crossings. The value of \(n\) can be negative numbers: if \(n=-2\), only the last two crossings are returned.
- \textit{thresh} is the threshold to be crossed.
- \textit{start} is the time at which the function is enabled.
- \textit{xtol} is the absolute tolerance in the X direction.
- \textit{ytol} is the absolute tolerance in the Y direction.
- \textit{accuracy} specifies that the function uses interpolation in the SKILL mode.

In the MDL mode, \textit{accuracy} specifies whether the function should use interpolation, or use iteration controlled by the absolute tolerances to calculate the value. ‘\textit{interp}’ directs the function to use interpolation, and ‘\textit{exact}’ directs the function to consider the xtol and yval values.
Example

The following input signal with the values \(\text{sig}=V_{\text{out}}, \text{dir}='\text{rise}, \text{and} \text{thresh}=1.0\)

is transformed into the following output waveform.
d2r (degrees-to-radians)

Converts a waveform from degrees to radians. This function is available only in the MDL mode.
db10

Converts a signal, in watts, to db where $db = 20 \times \log(x)$. This function is available only in the MDL mode.
db20

Returns the dB magnitude for a voltage or current. This function is available only in the SKILL mode.
dBm

Converts a signal, in watts, to dbm where \( \text{dbm} = 10 \log(x) + 30 \).
delay

Computes the delay between two points using the cross function. This function is available only in the SKILL mode.

- signal1 is the name of the first signal.
- signal2 is the name of the second signal.
- Threshold Value1 is the first threshold to be crossed.
- Edge Number1 is the number that specifies which crossing is to be the trigger event. For example, if Edge Number1=2, the trigger event is the second edge of the first waveform with the specified type that crosses Threshold Value 1.
- Edge Type1 is the direction of the first crossing event. rising directs the function to look for crossings where the Y value is increasing, falling for crossings where the Y value is decreasing, and either for crossings in either direction.
- Threshold Value2 is the second threshold to be crossed.
- Edge Number2 is the number that specifies which crossing is to be the trigger event. For example, if Edge Number2=2, the trigger event is the second edge of the first waveform with the specified type that crosses Threshold Value 2.
- Edge Type2 is the direction of the second crossing event. rising directs the function to look for crossings where the Y value is increasing, falling for crossings where the Y value is decreasing, and either for crossings in either direction.

\[
\text{delay} = \text{xcross2} - \text{xcross1}
\]
deltax

Returns the difference in the abscissas of two cross events. This function is available only in the MDL mode.

- \textit{sig1} is the signal whose cross event begins the measurement interval.
- \textit{sig2} is the signal whose cross event ends the measurement interval.
- \textit{dir1} is the direction of the cross at the beginning of the measurement interval. \textit{rise} directs the function to look for crossings where the Y value is increasing, \textit{fall} for crossings where the Y value is decreasing, and \textit{cross} for crossings in either direction.
- \textit{n1} is the occurrence of the crossing for the beginning of the measurement interval. The first crossing is \textit{n}=1, the second crossing is \textit{n}=2, and so on.
- \textit{thresh1} is the Y value whose crossing begins the measurement interval.
- \textit{start1} is the time at which the function is enabled.
- \textit{dir2} is the direction of the cross at the end of the measurement interval. \textit{rise} directs the function to look for crossings where the Y value is increasing, \textit{fall} for crossings where the Y value is decreasing, and \textit{cross} for crossings in either direction.
- \textit{n2} is the occurrence of the crossing for the end of the measurement interval. The first crossing is \textit{n}=1, the second crossing is \textit{n}=2, and so on.
- \textit{thresh2} is the Y value whose crossing ends the measurement interval.
- \textit{start2} is the offset from \textit{start1} where the function begins looking for the cross that ends the delay measurement.
- \textit{xtol} is the absolute tolerance in the X direction.
- \textit{ytol} is the absolute tolerance in the Y direction.
  Default: 1
- \textit{accuracy} specifies that the function uses interpolation in the SKILL mode.

In the MDL mode, \textit{accuracy} specifies whether the function should use interpolation, or use iteration controlled by the absolute tolerances to calculate the value. \textit{interp} directs the function to use interpolation, and \textit{exact} directs the function to consider the \textit{xtol} and \textit{yval} values.
Example

The following diagram illustrates how the result is determined with the values $\text{sig}_1 = V(\text{in})$, $\text{sig}_2 = V(\text{out})$, $\text{dir}_1 = \text{fall}$, $\text{thresh}_1 = 0.5$, $\text{dir}_2 = \text{fall}$, $\text{thresh}_2 = 0.5$, $\text{start}_2 = 10\text{n}$.

This delay masks out the first falling edge of the signal $V(\text{out})$. 

$$\text{thresh}_1 = \text{thresh}_2 = 0.5V$$

$\text{start}_2 = 10\text{n}$
deriv

Returns the derivative of a signal.
dft (Discrete Fourier Transform)

This function is available only in the SKILL mode.

The tool which converts a temporal (time domain) description of a signal (real or complex) into one in terms of its frequency components is called the Fourier Transform. dft (Discrete Fourier Transform) is the discrete formulation of the Fourier Transform, which takes such regularly spaced data values (samples in time domain), and returns the value of the Fourier Transform for a set of values in frequency domain which are equally spaced. Most of the time, however, we work on real-valued signals only.

Consider a complex series (signal) \( w(k) \) with \( N \) samples of the form

\[
\begin{align*}
& w(0), \ w(1), \ w(2), \ldots, \ w(k), \ldots, \ w(N-1)
\end{align*}
\]

Further, assume that the series outside the range \( 0, N-1 \) is extended \( N \)-periodic, that is, \( w(k) = w(k+N) \) for all \( k \). The dft of this series will be denoted \( W(n) \), will also have \( N \) samples and will be defined as:

\[
W(n) = \frac{1}{N} \sum_{k=0}^{N-1} w(k) \left( e^{-2\pi i k n / N} \right)
\]

where \( n=0, \ldots, N-1 \)

- The first sample \( W(0) \) of the transformed series is the DC component, more commonly known as the average of the input series.
- The dft of a real series results in a symmetric series about the Nyquist frequency (described below).
- The highest positive (or negative) frequency sample is called the Nyquist frequency. This is the highest frequency component that should exist in the input series for the DFT to receive 'unpredictable' results. More specifically, if there are no frequencies above Nyquist frequency, the original signal can be exactly reconstructed from the samples. The Nyquist Theorem (or Shannon's Sampling Theorem) exactly specifies this – that for a band limited signal, you must sample at a frequency over twice the maximum frequency of the signal to reconstruct it from the samples.

While the dft transform above can be applied to any complex valued series, in practice for large series it can take considerable time to compute, the time taken being proportional to the square of the number of points (samples) in the series. A much faster algorithm has been developed by Cooley and Tukey called the FFT (Fast Fourier Transform). The only requirement of the most popular implementation of this algorithm (Radix-2 Cooley-Tukey) is that the number of points in the series be a power of 2 i.e. \( N=2^n \).
Given $N$ input points, the fft returns $N$ frequency components, of which the first $(N/2 + 1)$ are valid. (The other components are mirror images and are considered invalid since the frequencies they represent do not satisfy the Nyquist Theorem above.) They start with the DC component, and are spaced apart by a frequency of $(1 / (n \cdot \text{deltaT}))$. The magnitude of the complex number returned is the frequency's relative strength.

The dft function computes the discrete Fourier Transform of the buffer by fft algorithm where $\text{deltaT} = (t2 - t1) / N$. The waveform is sampled at the following $N$ timepoints:

$t1, t1 + \text{deltaT}, t1 + 2 \cdot \text{deltaT}, \ldots, t1 + (N - 1) \cdot \text{deltaT}$

The output of $\text{dft}()$ is a frequency waveform, $W(f)$, which has $(N/2 + 1)$ complex values: the dc term, the fundamental, and $(N/2 - 1)$ harmonics.

**Note:** The last time point, $(t1 + (N - 1) \cdot \text{deltaT})$, is $(t2 - \text{deltaT})$ rather than $t2$. The dft function assumes that $w(t1)$ equals $w(t2)$.

- **Signal** is the name of the signal.
- **From** is the starting point of the range over which you want to compute the transform.
- **To** is the ending point of the range over which you want to compute the transform. Be sure to cover at least one complete period of your slowest frequency.
- **Number of Samples** is the number of samples you want to take in expanding the Fourier transform. This number should be a power of 2. If it is not, the system increases the value to the next higher power of 2. Sample at a rate that is at least twice your highest frequency component (the Nyquist rate). Pick a sampling rate high enough that closely spaced frequency components can be resolved.
- **Window Type** is the window you want to use.
- **Smoothing Factor** is the smoothing factor applicable to the Kaiser window only. The Smoothing Factor field accepts values from 0 to 15. The value 0 implies no smoothing and is equivalent to a rectangular window.
- **Coherent Gain** is a scaling parameter. A non-zero value scales the power spectral density by $1/(f_{\text{cohGain}})$. Valid values: $0 \leq f_{\text{cohGain}} \leq 1$. You can use 1 if you do not want the scaling parameter to be used. Default value: 1.

**Coherent gain factor**

When you run the transient analysis, keep the maximum time step small enough to represent the highest frequency component accurately. The maximum time step should be smaller than
the sampling period that you use for the dft of the time domain waveform. The samples in the
dft will either hit a data point (calculated exactly by the simulator) or an interpolated point
between two data points.

Choosing a maximum timestep during transient simulation that is smaller than the dft
sampling period ensures that sampling maintains a resolution at least equal to that of the
transient time-domain waveform.

The start and stop times should not coincide with the boundaries of the time-domain
waveform. The boundary solutions might be imprecise and generate incorrect results if used
in other calculations.

One of the uses of fast Fourier Transform windowing is to reduce discontinuities at window
edges caused by having a nonintegral number of periods of a signal in a window. This
removes the abrupt edges, making them fall off smoothly to zero, and can improve the validity
of the fft components obtained. You can also use fft windowing to 'dig out' the details of signal
components that are very close Gin frequency or that consist of both large and small
amplitudes.
dftbb

This function is available only in the SKILL mode.

dftbb computes the Discrete Fourier Transform (fast Fourier transform) of a complex signal
\[ z(t) = x(t) + j y(t) \]

\[ Z(n) = \text{Re}Z(n) + j \text{Im}Z(n) = \sum_{k=0}^{N-1} z(k) \exp(-j \theta n k) \]

where \( \theta = \frac{2 \pi}{N} \); \( n = 0, 1, ..., N-1 \).

Both waveforms are sampled at the following \( N \) timepoints:
\[ t1, t1 + \delta T, t1 + 2 \delta T, ..., t1 + (N - 1) \delta T \]

The output of dftbb(\text{waveform1, waveform2}) are \( N \) complex values.

The dftbb function is required because the dft function gives out the amplitudes \( \sqrt{\text{Re}^2 + \text{Im}^2} \) of dfts of real signals only – not Re and Im. Therefore, you cannot replace one dft of the complex signal \( z(t) = i(t) + j q(t) \) with two dfts of two real signals \( i(t) \) and \( q(t) \):

\[ I(n) = \text{Re}I(n) + j \text{Im}I(n) = \sum_{k=0}^{N-1} i(k) \exp(-j \theta n k) \]

\[ Q(n) = \text{Re}Q(n) + j \text{Im}Q(n) = \sum_{k=0}^{N-1} q(k) \exp(-j \theta n k) \]

and then compute:
\[ \text{Re}Z(n) = \text{Re}I(n) - \text{Im}Q(n) \]
\[ \text{Im}Z(n) = \text{Im}I(n) + \text{Re}Q(n) \] for \( n = 0, 1, ..., N-1 \).

The above definition is for single-sided output waveforms. This holds true for double-sided output waveforms except that the previous output waveform is translated so that \( n \) varies from \(-N/2\) to \((N/2) - 1\).

- Signal1 is the first waveform
- Signal2 is the second waveform.
- From is the starting point of the range over which you want to compute the transform.
To is the ending point of the range over which you want to compute the transform. Be sure to cover at least one complete period of your slowest frequency.

Number of Samples is the number of samples you want to take in expanding the Fourier transform. This number should be a power of 2. If it is not, the system increases the value to the next higher power of 2. Sample at a rate that is at least twice your highest frequency component (the Nyquist rate). Pick a sampling rate high enough that closely spaced frequency components can be resolved.

Window Type is the window you want to use.

Smoothing Factor is the smoothing factor applicable to the Kaiser window only. The Smoothing Factor field accepts values from 0 to 15. The value 0 implies no smoothing and is equivalent to a rectangular window.

Coherent Gain is a scaling parameter. A non-zero value scales the power spectral density by 1/(f_cohGain). Valid values: 0 <= f_cohGain <= 1. You can use 1 if you do not want the scaling parameter to be used. Default value: 1.

Coherent gain factor is

Spectrum Type is a string that can be either singleSided or doubleSided. When Spectrum Type is single-sided, the resultant waveform is only on one side of the Y-axis starting from 0 to N-1. When it is double-sided, the resultant waveform is symmetric to the Y-axis from -N/2 to N/2.
dutycycle

Calculates the ratio of the time for which the signal remains high to the period of the signal. You should use this function on periodic signals only. This function is available only in the MDL mode.

- $\textit{sig}$ is the name of the signal.
- $\textit{theta}$ is the percentage that defines the logic high of the signal. A threshold value is calculated as follows:
  $y\text{Thresh}=\textit{theta}/100\times(Y_{\text{max}}+Y_{\text{min}})$
  The portion of the signal above $y\text{Thresh}$ is taken as high.
dutycycles

Returns the dutycycle of a nearly-periodic signal as a function of time. This function is available only in the MDL mode.

- $\text{sig}$ is the name of the signal.
- $\theta$ is the percentage that defines the logic high of the signal. A threshold value is calculated as follows:
  \[ y\text{Thresh} = \theta/100 \times (Y_{\text{max}} - Y_{\text{min}}) \]
  The portion of the signal above $y\text{Thresh}$ is taken as high.

Example

The following input signal with the values $\text{sig}=\text{V(out)}$ and $\theta=40$
transforms into the following output signal:
exp

Returns the e^x value of a signal.
eyeDiagram

Gives an eye-diagram plot in which the waveform signal is divided into fixed time periods, which are then superimposed on each other. The result is a plot that has many overlapping lines enclosing an empty space known as the eye. The quality of the receiver circuit is characterized by the dimension of the eye. An open eye means that the detector will be able to distinguish between 1's and 0's in its input, while a closed eye means that a detector placed on $V_{out}$ is likely to give errors for certain input bit sequences.

This function is available only in the SKILL mode.

- **Signal** is the name of the signal.
- **Start Time** is the X-axis value from where the eye-diagram plot is to begin.
- **Stop Time** is the X-axis value where the eye-diagram plot is to end.
- **Period** is the time period for the eye diagram.
falltime

Returns the fall time for a signal measured between percent high and percent low of the
difference between the initial and final values. The measurement is always done with ordinate
values. This function is available only in the MDL mode.

■  

arg is the name of the signal.

■  

initval is the X value (if inittype is ‘x) or Y value (if inittype is ‘y) that starts the falltime
interval.

■  

finalval is the X value (if inittype is ‘x) or Y value (if inittype is ‘y) that ends the falltime
interval.

■  

inittype specifies whether the initial value is an X value (‘x) or a Y value (‘y).

■  

finaltype specifies whether the final value is an X value (‘x) or a Y value (‘y).

■  

theta1 is the threshold high expressed as a percentage of the difference between the
initial and final values.

■  

theta2 is the threshold low expressed as a percentage of the difference between the
initial and final values.

■  

xtol is the absolute tolerance in the X direction.

■  

ytol is the absolute tolerance in the Y direction.

■  

accuracy specifies that the function uses interpolation in the SKILL mode.

In the MDL mode, accuracy specifies whether the function should use interpolation, or
use iteration controlled by the absolute tolerances to calculate the value. ‘interp
directs the function to use interpolation, and ‘exact directs the function to consider the
xtol and yval values.
Example

The following diagram illustrates how the result is determined with the values \( \text{arg} = V(\text{out}), \text{initval} = 10\, \mu\text{V}, \text{inittype} = \times, \text{finalval} = 19\, \mu\text{V}, \text{finaltype} = \times, \theta_1 = 10, \theta_2 = 90 \)
**fft**

Performs a Fast Fourier Transform on the signal and returns its frequency spectrum. This function is available only in the MDL mode.

- *sig* is the name of the signal.
- *from* is the starting X value.
- *to* is the ending X value.
- *numPoints* is the number of data points to be used for calculating the fft. If this number is not a power of 2, it is automatically raised to the next higher power of 2.
- *window* is the algorithm used for calculating the fft. For more information, see `window`.

**Example**

The following input signal with the values *sig*=`V(out)`, *from*=1ns, *to*=200ns, *numpoints*=512, and *window*=`'bartlett`
transforms into the following output signal. The left subwindow shows the magnitude part of the spectrum and the right subwindow shows the phase part.
**flip**

Returns a reversed version of a signal (rotates the signal along the Y-axis).

**Example**

The following input signal

![Input Signal Graph](image)

is transformed into the following output signal.

![Output Signal Graph](image)
floor

Rounds a real number down to the closest integer value. This function is available only in the MDL mode.
This function is available only in the SKILL mode.
**fourEval**

Evaluates the Fourier series represented by the buffer expression. This function is an inverse Fourier transformation and thus the inverse of the *dft* function. It transforms the buffer expression from the frequency domain to the time domain. This function is available only in the SKILL mode.

- Signal is the name of the signal.
- From is time at which you want to begin evaluating the series.
- To is time till which you want to evaluate the series.
- By is the increment for evaluating the series.
**freq**

Returns a waveform representing the frequency of a signal versus time.

In the MDL mode,

- *sig* is the signal.
- *thresh* is the threshold Y-axis value to be crossed.
- *dir* is the direction of the crossing event.

**Example**

The following input signal
is converted to the following output signal
freq_jitter

Returns a waveform representing the deviation from the average frequency. This function is available only in the MDL mode.

- $sig$ is the name of the signal.
- $thresh$ is the threshold Y-axis value to be crossed.
- $dir$ is the direction of the crossing event.
- $binsize$ is the integer used to calculate the average frequency of the signal.
  - If $binsize=0$, all frequencies are used to calculate the average.
  - If $binsize=N$, the last $N$ frequencies are used to calculate the average.

Example

The following input signal with the values $sig=V(\text{out})$, $thresh=0.5$, $dir=\text{ rise}$, $binsize=4$.
is transformed into the following output signal
ga

Returns the available gain. This function is available only in the SKILL mode.
gac_freq

This function is available only in the SKILL mode.

- Gain (dB)
- Start
- Stop
- Step
**gac_gain**

This function is available only in the SKILL mode.

- Frequency (Hz)
- Start
- Stop
- Step
**gainBwProd**

Returns the product of DC gain and upper cutoff frequency for a low-pass type filter or amplifier.
gainmargin

Computes the gain margin of the loop gain of an amplifier.

The gain margin is calculated as the magnitude (in dB) of the gain at f0. The frequency f0 is the smallest frequency in which the phase of the gain provided is -180 degrees. For stability, the gain margin must be positive.
getAsciiWave

Returns a piecewise linear function from a column of x and y values in a file. This function is available only in the SKILL mode.
gmax

Returns the maximum available gain for a two port. This function is available only in the SKILL mode.
gmin

Returns the optimum noise reflection coefficient for NFmin. This function is available only in the SKILL mode.
gmsg

Returns the maximum stable power gain for a two port. This function is available only in the SKILL mode.
gp

Returns the power gain. This function is available only in the SKILL mode.
gpc_freq

This function is available only in the SKILL mode.

- Gain (dB)
- Start
- Stop
- Step
**gpc_gain**

This function is available only in the SKILL mode.

- Frequency (Hz)
- Start
- Stop
- Step
groupdelay

Computes the group delay of the expression in the buffer. Group delay is defined as the derivative of the phase with respect to frequency. Group delay is expressed in seconds. It is calculated using the vp function as shown below:

$$GroupDelay = \frac{d\phi}{d\omega} = \frac{d}{df} \left[ \frac{\text{phase}(\text{netX})}{360} \right]$$

Example

The following input signal
is converted to the following output signal
gt

Returns the transducer gain. This function is available only in the SKILL mode.
gumx

Returns the maximum unilateral power gain for a two port. This function is available only in the SKILL mode.
harmonic

Returns the harmonic waveform of the specified harmonic. This function is available only in the SKILL mode.

- **Signal** is the name of the signal.

- **Harmonic Number** is the index number that designates the harmonic information to be returned. For the pss, pac, and pxf analyses, the index is an integer. For the pdisto analysis, the index is a list of integers that correspond to the frequency names listed in the funds analysis parameter in the netlist. You can specify more than harmonic number at a time.
**harmonicFreq**

Returns the harmonic waveform of the specified harmonic. This function is available only in the SKILL mode.

- *Signal* is the name of the signal.
- *Harmonic Number* is the index number that designates the harmonic information to be returned. For the pss, pac, and pxf analyses, the index is an integer. For the pdisto analysis, the index is a list of integers that correspond to the frequency names listed in the funds analysis parameter in the netlist. You can specify more than harmonic number at a time.
**histo**

Creates a histogram from a signal. This function is available only in the MDL mode.

- **$sig$** is the waveform.
- **$nbins$** is the number of bins to be created.
- **$min$** is the value that specifies the smaller end point of the range of values included in the histogram.
- **$max$** is the value that specifies the larger end point of the range of values included in the histogram.

**Example**

The input values $sig=V(out)$, $nbins=10$, $min=-1.0$, $max=4.0$

creates a display with 10 bins that might look like this when the leftmost bin is empty.
**ifft**

Performs an inverse Fast Fourier Transform on a frequency spectrum and returns the time domain representation of the spectrum. This function is available only in the MDL mode.

The frequency spectrum.

**Example**

The input signal on the left side with the values $\text{sig}=V(\text{out}), \text{from}=1\text{ns}, \text{to}=200\text{ns}, \text{npoints}=512$ results in the graph on the right side.

The signal $\text{out}$

Fast fourier transform of the signal $\text{out}$
Now if you perform an `ifft` with the values $\text{sig} = V(\text{out})$, $\text{from} = 1\,\text{ns}$, $\text{to} = 200\,\text{ns}$, $\text{npoints} = 512$, the result is the same as the original signal ($\text{out}$) – from 1ns to 200ns.
iinteg

Returns the incremental area under the waveform.

Example

The following input signal
transforms into the following output signal

Each X value on the output trace is equal to the area under the input trace from start till that particular X-value.
**im**

Returns the imaginary part of a complex number. This function is available only in the MDL mode.
imag

Returns the imaginary component. This function is available only in the SKILL mode.
int

Returns the integer portion of a real value.
**integ**

Returns the area bounded under the curve.

**Example**

The following diagram illustrates how the result with the values `integ(trim(sig=V(sine), from=10n, to=50n))` is determined. The result is equal to the shaded area in the graph.
ipn

Plots the $n$th order intercept between two harmonics of a waveform that you define. This function is available only in the SKILL mode.

- **Signal** is the name of the signal.
- **Spur Order** determines what order of interference is calculated for the spurious and reference waves. The default value 3 corresponds to the IP3 function. If you use a value other than 3, that order of interference is calculated between those two waves.
- **Spur Harmonic** is the harmonic number for the spurious waveform.
- **Extrapolation Point** is the extrapolation point for the IPN function. This is the X-axis value.
- **Reference Harmonic** is the harmonic number for the reference waveform.
**ipnVRI**

Performs an intermodulation $n$th-order intercept point measurement

You can use this function to simplify the declaration of an IPN measurement. This function extracts the spurious and reference harmonics from the input waveform(s), and uses $\text{dBm}(\text{spectralPower}(i \text{ or } v/r,v))$ to calculate the respective powers. The function then passes these power curves or numbers and the remaining arguments to the IPN function to complete the measurement.

From each of the spurious and reference power waveforms (or points), the IPN function extrapolates a line of constant slope (dB/dB) according to the specified order and input power level. These lines represent constant small-signal power gain (ideal gain). The IPN function calculates the intersection of these two lines and returns the value of either the $x$ coordinate (input referred) or $y$ coordinate (output referred).

- $signal$ is the name of the signal.
- $Spur Harmonic$ is the harmonic index for the spurious waveform.
- $Reference Harmonic$ is the harmonic index for the reference waveform.
- $Spur Order$ determines what order of interference is calculated for the spurious and reference waves. The default value 3 corresponds to the IP3 function. If you use a value other than 3, that order of interference is calculated between those two waves.
- $Extrapolation Point$ is the extrapolation point for the IPN function. This is the $X$-axis value. The default is the minimum $X$-axis value of the input voltage waveform.
- $Load Resistance$ is the resistance into the output port.
  To get the $X$-coordinate of the intercept, select $Input Referred IPN$. To get the $Y$-coordinate of the intercept, specify $Output Referred IPN$.

- $Circuit Input Power$ specifies whether the input power is a variable sweep or a single point.
**kf**

Returns the stability factor K. This function is available only in the SKILL mode.
In

Returns the natural logarithm of a signal.
**log10**

Returns the base 10 logarithm of a signal.
Isb (Load Stability Circles)

Returns the load stability circles. This function is available only in the SKILL mode.

- **Start (Hz)** is the start of the frequency range.
- **Stop (Hz)** is the end of the frequency range.
- **Step** is the increment for the frequency range.
Ishift (Left Shift)

Shifts the data in the Graph Window to the left by the specified amount. A negative value shifts the data to the right. This function is available only in the SKILL mode.

- *Signal* is the name of the signal.
- *Delta X* is the amount by which you want to shift the data.
**mag**

Returns the magnitude of a signal.
**max**

Returns the absolute value of a signal, or the maximum value of two real values. This function is available only in the MDL mode.
**min**

Returns the minimum value of a signal. This function is available only in the MDL mode.
mod

Returns the floating point remainder of the dividend divided by the divisor. The divisor cannot be zero. This function is available only in the MDL mode.
movingavg

Calculates the moving average for the specified signal. This function is available only in the MDL mode.
nc_freq (Noise Circles - Sweep Frequency)

This function is available only in the SKILL mode.

- Level (dB)
- Start
- Stop
- Step
nc_gain (Noise Circles - Sweep Level)

This function is available only in the SKILL mode.

- Frequency (Hz)
- Start
- Stop
- Step
**nf**

Retrieves F from the PSF file. This function is available only in the SKILL mode.

\[ NF = \text{db}10(F) \]

where NF is the noise figure and F is the noise factor.
**nfmin**

Retrieves Fmin from the PSF file. This function is available only in the SKILL mode.

\[ NF\text{min}=\text{db10}(F\text{min}) \]

where NFmin is the minimum noise figure and Fmin is the minimum noise factor.
overshoot

Returns the overshoot/undershoot of a signal as a percentage of the difference between initial and final values.

In the SKILL mode,

- **Signal** is the name of the signal.
- **Initial Value Type** specifies whether the initial value is voltage (\(y\)) or time (\(x \text{ at } y\)).
- **Initial Value** is the initial value. To calculate the undershoot of a signal, the Initial Value should be higher than Final Value.
- **Final Value Type** specifies whether the final value is voltage (\(y\)) or time (\(x \text{ at } y\)).
- **Final Value** is the final value.

In the MDL mode,

- **sig** is the name of the signal.
- **initval** is the initial value. To calculate the undershoot of a signal, the initval should be higher than finalval.
- **finalval** is the final value.
- **inittype** specifies whether the initial value is a time (\(’x\)) or voltage value (\(’y\)).
- **finaltype** specifies whether the final value is a time (\(’x\)) or voltage value (\(’y\)).
Example

The following diagram illustrates how the result is obtained with the values \( \text{signal}=V_{\text{out}} \), \( \text{Initial Value Type}=y \), \( \text{Final Value Type}=y \), \( \text{Initial Value}=1 \), and \( \text{Final Value}=3 \).

\[ \text{OvershootOut} = \frac{\text{Maximum Value} - \text{Final Value}}{\text{Final Value} - \text{Initial Value}} \]

OvershootOut is given by the following formula:
*period_jitter*

Returns a waveform representing the deviation from the average period. This function is available only in the MDL mode.

- `sig` is the name of the signal.
- `thresh` is the threshold Y-axis value defining the period/frequency of the signal.
- `dir` is the direction of the crossing event.
- `binsize` is the integer used to calculate the average frequency of the signal.
  - If `binsize`=0, all periods are used to calculate the average.
  - If `binsize`=N, the last N periods are used to calculate the average.
phase

Returns the phase of a signal in radians.
phaseMargin

Computes the phase margin of the loop gain of an amplifier. The phase margin is calculated as the difference between the phase of the gain in degrees at f0 and at -180 degrees. The frequency f0 is the smallest frequency where the gain is 1. For stability, the phase margin must be positive.

small-sig is the loop gain of interest over a sufficiently large frequency range.
phaseNoise

Plots the phase noise waveform for noise analysis results. This function is available only in the SKILL mode.
**pow**

Returns the value of base raised to the power of exponent \( (base^{exponent}) \). This function is available only in the MDL mode.

- \( base \) is the base argument.
- \( exponent \) is the exponent argument.
**pp (peak-to-peak)**

Returns the difference between the highest and lowest values of a signal. This function is available only in the MDL mode.

**Example 1**

The following diagram illustrates how the pp value is determined.
psd (Power Spectral Density)

Describes how the power (or variance) of a time series (signal) is distributed with frequency. Mathematically, it is defined as the Fourier Transform of the auto correlation sequence of the time series (signal). The waveform is first interpolated to generate evenly spaced data points in time. The spacing of the data points is the inverse of the \textit{dft} sampling frequency. The \textit{psd} is computed by first breaking up the time interval into overlapping segments. Each segment is multiplied, time point by time point, by the specified windowing function. The \textit{dft} is performed on each windowed segment of the baseband waveform. At each frequency, the \textit{dfts} from all segments are averaged together and the squared modulus of these averages gives the \textit{psd}.

This function is available only in the SKILL mode.

- \textit{Signal} is the name of the signal.
- \textit{From} is the starting time for the spectral analysis interval.
- \textit{To} is the ending time for the spectral analysis interval.
- \textit{Number of Samples} is the number of time domain points to be used.
- \textit{Window Type} is the window you want to use. If you select \textit{Kaiser}, type in a value for the Kaiser smoothing factor. The smoothing factor must be in the range $0 \leq \text{factor} \leq 15$, where 0 is the same as using a rectangular window.
- \textit{Smoothing Factor} applies only to the \textit{Kaiser} window type.
- \textit{Window Size} is the number of frequency domain points to use in the Fourier analysis. A larger window size results in an expectation operation over fewer samples, which leads to larger variations in the power spectral density. A small window size can smear out sharp steps in the power spectral density that might really be present.

- \textit{Detrending Mode}  
The \textit{psd} function works by applying a moving windowed FFT to time-series data. If there is a deterministic trend to the underlying data, you may want to remove the trend before performing the spectral analysis. For example, consider analyzing phase noise in a VCO model. Without the noise the phase increases more or less linearly with time, so it is appropriate to set the detrending mode to linear. To subtract an average value, set the detrending mode to mean. Where the spectrum of raw data is desired, set the detrending mode to none.

- \textit{Coherent Gain} is a scaling parameter. A non-zero value scales the power spectral density by $1/(\text{Coherent Gain})$. Valid values: $0 < \text{Coherent Gain} < 1$. You can use 1 if you do not want the scaling parameter to be used. Default value: 1.
Coherent Gain Factor

If you choose magnitude, dB20, or dB10, then enter a scaling factor, a non-zero factor scales the power spectral density by 1/(factor). Valid values for the factor are 0 < factor < 1. You can also use a value of 1 if you do not want the Coherent Gain Factor to be used.
**psdbb (Power Spectral Density Baseband)**

Returns an estimate for the power spectral density of a waveform1+j * waveform2. This function is available only in the SKILL mode.

- **Signal1** is the first waveform.
- **Signal2** is the second waveform.
- **From** is the starting time for the spectral analysis interval
- **To** is the ending time for the spectral analysis interval
- **Number of Samples** is the number of time domain points to use. The maximum frequency in the Fourier analysis is proportional to the Number of Samples parameter and inversely proportional to the difference between the starting time and the ending time
- **Window Type** is the type of window you want to use. If you select the *Kaiser* window type, then type in a value for the Kaiser smoothing factor. The smoothing factor must be in the range 0 <= factor <= 15, where 0 is the same as using a rectangular window.
- **Smoothing Factor** applies only to the *Kaiser* window type.
- **Window Size** is the number of frequency domain points to use in the Fourier analysis. A larger window size results in an expectation operation over fewer samples, which leads to larger variations in the power spectral density. A small window size can smear out sharp steps in the power spectral density that might really be present

- **Detrending Mode**
  The psdbb function works by applying a moving windowed FFT to time-series data. If there is a deterministic trend to the underlying data, you may want to remove the trend before performing the spectral analysis. For example, consider analyzing phase noise in a VCO model. Without the noise the phase increases more or less linearly with time, so it is appropriate to set the detrending mode to linear. To subtract an average value, set the detrending mode to mean. Where the spectrum of raw data is desired, set the detrending mode to none.

- **Coherent Gain** is a scaling parameter. A non-zero value scales the power spectral density by 1/(Coherent Gain). Valid values: 0 < Coherent Gain < 1. You can use 1 if you do not want the scaling parameter to be used. Default value: 1.

- **Coherent Gain Factor**
  If you choose *magnitude, dB20, or dB10*, then enter a scaling factor, a non-zero factor scales the power spectral density by 1/(factor). Valid values for the factor are 0 < factor < 1. You can also use a value of 1 if you do not want the *Coherent Gain Factor* to be used.
pzbode

Calculates and plots the transfer function for a circuit from pole zero simulation data.

- \(poles\) is the poles.
- \(zeroes\) is the zeroes.
- \(c\) is the transfer gain constant.
- \(minfreq\) is the minimum frequency for the bode plot.
- \(maxfreq\) is the maximum frequency for the bode plot.
- \(npoints\) is the frequency interval for the bode plot, in points per decade.

Example

The following diagram illustrates how the result with the values \(poles=POLES\{I\{R_1\}\}\), \(zeroes=ZEROES\{I\{R_1\}\}, c=I\{R_1\}\{K\}, minfreq=1e-3, maxfreq=1e3, and npoints=1000\) is determined.

Polar Plot

![Polar Plot](image1)

Corresponding bode plot

![Corresponding bode plot](image2)
**pzfilter**

Filters the poles and zeroes according to the specified criteria. The `pzfilter` function works only on pole zero simulation data.

- **poles** is the poles.
- **zeroes** is the zeroes.
- **maxfreq** is the frequency up to which the poles and zeroes are plotted.
- **reldist** is the relative distance between the pole and zero. Pole-zero pairs with a relative distance lower than the specified value are not plotted.
- **absdist** is the absolute distance between the pole and zero. Pole-zero pairs with an absolute distance lower than the specified value are not plotted.
- **minq** is the minimum Q-factor. Pole-zero pairs with a Q-factor less than the specified value are not cancelled. The equations that define the Q-factor of a complex pole or zero are described in the section below.

**Note:** If you do not specify **maxfreq**, **reldist**, **absdist**, or **minq**, `pzfilter` filters out the poles and zeroes with a frequency higher than 10 GHz (default value of **maxfreq**).

### Equations Defining the Q-Factor of a Complex Pole or Zero

<table>
<thead>
<tr>
<th>Condition</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{Re}(X) &lt; 0.0$</td>
<td>$Q = 0.5 \times \sqrt{[\text{Im}(X)/\text{Re}(X)]^2 + 1}$</td>
</tr>
<tr>
<td>$\text{Re}(X) = 0$</td>
<td><strong>UNDEFINED</strong></td>
</tr>
<tr>
<td>$\text{Re}(X) &gt; 0.0$</td>
<td>$Q = -0.5 \times \sqrt{[\text{Im}(X)/\text{Re}(X)]^2 + 1}$</td>
</tr>
</tbody>
</table>

### Filtration Rules

- Real poles can be cancelled only by real zeroes. A real pole $P$ is cancelled by a real zero $Z$ if the following equation is satisfied:

  \[ |P - Z| < \text{absdist} + \frac{|P + Z|}{2} \times \text{reldist} \]

- Complex poles and zeroes always occur in conjugated pairs. A pair of conjugated poles can only be canceled by a pair of conjugated zeroes. A pole pair $P_1 = a + jb, P_2 = a - jb$ is cancelled by a zero pair $Z_1 = c + jd, Z_2 = c - jd$, if the following equation is satisfied:
Poles in the right-half plane are never cancelled because they show the instability of the circuit.

Example

The values $\text{poles}=\text{POLES}<\text{I}<\text{R}_2>$, $\text{zeroes}=\text{ZEROES}<\text{I}<\text{R}_2>$, $\text{absdist}=0.05$, and $\text{ming}=10000$ filters pole-zero pairs with a relative distance of less than 0.05 Hz from the plot on the left side. In the filtered plot shown on the right side, two pole-zero pairs have been filtered out.
r2d (radians-to-degrees)

Converts a scalar or waveform expressed in radians to degrees. This function is available only in the MDL mode.
re

Returns the real portion of a complex number. This function is available only in the MDL mode.
real

In SKILL, returns the real component of a signal.

In MDL, Creates a real number from an integer number.
risetime

Returns the rise time for a signal measured between percent low and percent high of the difference between the initial and final value.

In the SKILL mode,

- **Signal** is the of the signal.
- **Initial Value Type** specifies whether the initial value is an X-axis (x at y) or Y-axis value (y).
- **Initial Value** is the value that starts the rise time interval.
- **Final Value Type** specifies whether the final value is an X-axis (x at y) or Y-axis value (y).
- **Final Value** is the value that ends the rise time interval.
- **Percent Low** is the percent low.
- **Percent High** is the percent high.

In the MDL mode,

- **sig** is the name of the signal.
- **initval** is the X-axis (if inittype is ’x) or Y-axis value (if inittype is ’y) that starts the rise time interval. The measurement is always done in ordinate values.
- **finalval** is the X-axis (if inittype is ’x) or Y-axis (if inittype is ’y) that ends the rise time interval. The measurement is always done in ordinate values.
- **inittype** specifies whether the initial value is an X-axis (’x) or Y-axis value (’y).
- **finaltype** specifies whether the final value is an X-axis (’x) or Y-axis value (’y)
- **theta1** is the percent low.
- **theta2** is the percent high.
- **xtol** is the absolute tolerance in the X direction.
- **ytol** is the absolute tolerance in the Y direction.
- **accuracy** specifies that the function uses interpolation in the SKILL mode.
  
In the MDL mode, **accuracy** specifies whether the function should use interpolation, or use iteration controlled by the absolute tolerances to calculate the value. ’interp
directs the function to use interpolation, and \( '\text{exact} \) directs the function to consider the xtol and yval values.

**Example 1**

The following diagram illustrates how the result with the values \( \text{signal}=V(\text{out}) \), \( \text{Initial Value}=19u \), \( \text{Final Value}=30u \), \( \text{Initial Value Type}=x \), \( \text{Final Value Type}=x \), \( \text{Percent Low}=10 \), and \( \text{Percent High}=90 \) is determined.

![Diagram](image-url)
Example 2

The following diagram illustrates how the result with the values \textit{signal} = V_{(out)}, \textit{Initial Value} = 0V, \textit{Final Value} = 5V, \textit{Initial Value Type} = y, \textit{Final Value Type} = y, \textit{Percent Low} = 10, and \textit{Percent High} = 90 is determined.
rms (root-mean-square)

Returns the root mean square of a signal.
rmsNoise

Computes the integrated root-mean-square of the total output noise over the bandwidth specified in hertz in the From and To fields. This function is available only in the SKILL mode.

- From is the starting time for the measurement.
- To is the ending time for the measurement.
rn

Returns the normalized equivalent noise resistance. This function is available only in the SKILL mode.
**Rn**

Returns the equivalent noise resistance. This function is available only in the SKILL mode.
root

Computes the value of x at which f(x) equals the specified threshold. This function is available only in the SKILL mode.

- *Signal* is the name of the signal.
- *Threshold* is the waveform value at which to compute the root value.
- *Nth Root* is the root you want to see.
round

Rounds a number to the closest integer value. This function is available only in the MDL mode.
s11

Returns 2-port S-parameters. This function is available only in the SKILL mode.
s12

Returns 2-port S-parameters. This function is available only in the SKILL mode.
s21

Returns 2-port S-parameters. This function is available only in the SKILL mode.
s22

Returns 2-port S-parameters. This function is available only in the SKILL mode.
sample

sample

Returns a waveform representing a sample of the signal based on step size or points per
decade.

In the SKILL mode,

■  **Signal** is the name of the signal.
■  **From** is the X-axis value at which the sampling begins.
■  **To** is the X-axis value at which the sampling stops
■  **Type** specifies whether the sample should be linear or logarithmic.
■  **By** specifies the step size for the sample (if **type** is **linear**) or the points per decade (if
  **type** is **logarithmic**).

In the MDL mode,

■  **sig** is the signal.
■  **from** is the X-axis value at which the sampling begins.
■  **to** is the X-axis value at which the sampling stops.
■  **type** specifies whether the sample should be linear or logarithmic.
■  **by** specifies the step size for the sample (if **type** is `'linear`) or the points per decade (if
  **type** is `'logarithmic`)
Example

The following input signal with the values $signal = v(2)$, $From = 7.5\mu s$, $To = 18\mu s$, and $By = 5\mu s$, $type = 'linear'$

transforms into the following output signal
settlingTime

Calculates the time required by a signal to settle at a final value within a specified limit.

In the SKILL mode,

- **Signal** is the name of the signal.
- **Initial Value Type** specifies whether the initial value is an X-axis \( x \) or Y-axis value \( y \).
- **Initial Value** is the starting value for the measurement.
- **Final Value Type** specifies whether the final value is an X-axis \( x \) or Y-axis value \( y \).
- **Final Value** is the final value for the measurement.
- **Percent of Step** is the percentage of \( (Final \ value - Initial \ Value) \) within which the signal has to settle.

In the MDL mode,

- **sig** is the name of the signal.
- **initval** is the starting value for the measurement.
- **finalval** is final value for the measurement.
- **inittype** specifies whether the initial value is an X-axis \( x \) or Y-axis value \( y \).
- **finaltype** specifies whether the final value is an X-axis \( x \) or Y-axis value \( y \).
- **theta** is the percentage of \( (finalval-initval) \) within which the signal has to settle.

**Example**

The following diagram illustrates how the result with the values \( signal=V(out), Initial \ Value \) Type\(=y, Initial \ Value=0, Final \ Value \) Type\(=x \ at \ y, Final \ Value=1.0, \) and **Percent of Step**=5 is determined.
sign

Returns a value that corresponds to the sign of a number. This function is available only in the MDL mode.
sin

Returns the sine of a signal.
**sinh**

Returns the hyperbolic sine of a signal.
slewrate

Computes the average rate at which the buffer expression changes from percent low to percent high of the difference between the initial value and the final value.

In the SKILL mode,

- **Signal** is the name of the signal.
- **Initial Value Type** specifies whether the initial value is an X-axis (\(x \text{ at } y\)) or Y-axis value (\(y\)).
- **Initial Value** is the starting value for the measurement.
- **Final Value Type** specifies whether the final value is an X-axis (\(x \text{ at } y\)) or Y-axis value (\(y\)).
- **Final Value** is the final value for the measurement.
- **Percent Low** is the percent low.
- **Percent High** is the percent high.

In the MDL mode,

- **sig** is the name of the signal.
- **initval** is the starting value for the measurement.
- **finalval** is final value for the measurement.
- **inittype** specifies whether the initial value is an X-axis (’\(x\)) or Y-axis value (’\(y\)).
- **finaltype** specifies whether the final value is an X-axis (’\(x\)) or Y-axis value (’\(y\)).
- **theta1** is the percent low.
- **theta2** is the percent high.
- **xtol** is the absolute tolerance in the X direction.
- **ytol** is the absolute tolerance in the Y direction.
- **accuracy** specifies that the function uses interpolation in the SKILL mode.

In the MDL mode, **accuracy** specifies whether the function should use interpolation, or use iteration controlled by the absolute tolerances to calculate the value. ’interp directs the function to use interpolation, and ’exact directs the function to consider the xtol and yval values.
**snr**

Calculates the signal to noise ratio from a complex frequency based signal. This function is available only in the MDL mode.

- *sig* is the name of the signal.
- *sig_from* is the left window border of the signal. The *sig_from* value must be greater than or equal to *noise_from*.
- *sig_to* is the right window border of the signal. The *sig_to* value must be less than or equal to *noise_to*.
- *noise_from* is the left window border of the noise.
- *noise_to* is the right window border of the noise.
Example

You have the following frequency plot.

To determine the signal-to-noise ratio, you use the values

```
export real snr(fft(V(out),0,1e-3,1024),9e3,11e3,1,500e3)
```

which, in this case, returns

```
29.268026738835342dB
```
**spectralPower**

Plots the spectral power for the specified current waveform and voltage waveform. This function is available only in the SKILL mode.

- *Current waveform* is the current waveform for which you want to calculate the spectral power.
- *Voltage waveform* is the voltage waveform for which you want to calculate the spectral power.
sqrt

Returns the square root of a signal.
ssb

Returns source stability circles. This function is available only in the SKILL mode.

- **Start (Hz)** is the start of the frequency range.
- **Stop (Hz)** is the end of the frequency range.
- **Step** is the increment.
stathisto

Creates a histogram from a signal. This function is available only in the MDL mode.

- **sig** is the waveform.
- **nbins** is the number of bins to be created.
- **min** is the value that specifies the smaller end point of the range of values included in the histogram.
- **max** is the value that specifies the larger end point of the range of values included in the histogram.
- **innerswpval** is the inner-most sweep parameter in the dataset. You use this parameter to slice through parametric waveforms to extract the data for the histogram. Default: The first available value of time in the dataset.

**Example**

Assume that you have the results of running a Monte Carlo analysis on top of a transient analysis, so that the inner-most swept variable is time. Now, for the particular value of time specified by the **innerswpval** argument specification, the stathisto function creates a histogram by analyzing all the Monte Carlo iterations and extracting from each one the value of the signal at the specified time.

For example, to create a histogram for the time 100ns, you might use the following statement.

```
stathisto(I(V10\:p), innerswpval=100e-9)
```

To create a histogram for the time 650ps, you might use the following statement.

```
stathisto(I(V10\:p), innerswpval=.65e-9)
```
stddev

Computes the standard deviation of a waveform (or a family of waveforms) over its entire range. Standard deviation (stddev) is defined as the square-root of the variance where variance is the integral of the square of the difference of the expression \( f(x) \) from average \( (f(x)) \), divided by the range of \( x \). This function is available only in the SKILL mode.

For example, if \( y = f(x) \)

\[
\text{stddev}(y) = \frac{\int_{\text{from}}^{\text{to}} (y - \text{average}(y))^2}{\text{to} - \text{from}}
\]

If you want a different range, use the clip function to clip the waveform to the range you want.
**tan**

Returns the tangent of a signal.
tangent (Tangent Line)

Plots a line that passes through x and y coordinates and the slope that you specify. This function is available only in the SKILL mode.

- *Signal* is the name of the signal.
- *X Point* is the X-axis value you specify.
- *Y Point* is the Y-axis value you specify.
- *Slope* is the specified slope.
tanh

Returns the hyperbolic tangent of a signal.
thd (Total Harmonic Distortion)

Computes the percentage of total harmonic content of a signal with respect to the fundamental frequency. The computation uses the \texttt{dft} function. Assume that the \texttt{dft} function returns complex coefficients $A_0, A_1, \ldots, A_f, \ldots$. Note that fundamental frequency $f$ is the frequency contributing to the largest power in the signal. $A_0$ is the complex coefficient for the DC component and $A_i$ is the complex coefficient for the $i$th harmonic where $i \neq 0, f$. Then, total harmonic distortion is computed as:

$$\sqrt{\sum_{i=1, i \neq 0, f} |A_i|^2} \times 100\%$$

This function is available only in the SKILL mode.

- \textit{Signal} is the name of the signal.
- \textit{From} is the starting frequency.
- \textit{To} is the ending frequency.
- \textit{Number} of Samples is the number of time domain points to be used.
- \textit{Fundamental (Hz)} is the fundamental frequency of the signal.

The accuracy of the total harmonic distortion measurement depends on simulator options and the analysis parameters. For an accurate measurement set the following simulation options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Suggested Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RELTOL</td>
<td>1e-5</td>
</tr>
<tr>
<td>ABSTOL</td>
<td>1e-13</td>
</tr>
<tr>
<td>VNTOL</td>
<td>3e-8</td>
</tr>
<tr>
<td>TRTOL</td>
<td>1</td>
</tr>
<tr>
<td>METHOD</td>
<td>gear</td>
</tr>
<tr>
<td>MAXORD</td>
<td>3</td>
</tr>
</tbody>
</table>

Set the simulation timestep to be 1/100th of a cycle, and simulate for ten cycles. End the simulation slightly beyond the tenth cycle. When you use the calculator, measure during the
tenth cycle by specifying the beginning of the cycle as the *From* time and the end as the *To* time.
**trim**

Returns the portion of a signal between two points along the X-axis. This function is available only in the MDL mode.

- **sig** is the name of the signal.
- **from** is the starting X-axis value.
- **to** is the ending X-axis value.

**Example 1**

In WaveScan,

```
trim ( sig=V(sinewave), from=17n, to=29n )
```

transforms the following input signal
into the following output signal

![Graph showing a peak between 17n and 29n]
value

Computes the value of the waveform at the specified point. This function is available only in the SKILL mode.

- *Signal* is the name of the signal.
- *Interpolate At* is the point at which you want the value to be computed.
window

Applies the specified window to a signal. This function is available only in the MDL mode.

- **arg** is the name of the signal.
- **window** is the window to be applied.

Equations and Examples

This section describes the equations used by each type of window and then shows an example. In the equations:

\[ N = \text{total number of waveform points} \]
\[ n = \text{current waveform point} \]

<table>
<thead>
<tr>
<th>Window</th>
<th>Equation and Example</th>
<th>Where</th>
</tr>
</thead>
<tbody>
<tr>
<td>'rectangular</td>
<td>( w(n) = 1 )</td>
<td>( 0 \leq n \leq N )</td>
</tr>
<tr>
<td>'bartlett</td>
<td>( w(n) = 1 - \text{abs}\left(2 \times \frac{n}{N} - 1\right) )</td>
<td>( 0 \leq n \leq N )</td>
</tr>
<tr>
<td></td>
<td>( w(n) = 0 )</td>
<td>otherwise</td>
</tr>
</tbody>
</table>
### WaveScan User Guide
#### Calculator Functions

<table>
<thead>
<tr>
<th>Window</th>
<th>Equation and Example</th>
<th>Where</th>
</tr>
</thead>
<tbody>
<tr>
<td>'bartletthann'</td>
<td>[ w(n) = 0.62 \times 0.48 \times \text{abs}\left(\frac{n}{N} - 0.5\right) + 0.38 \times \cos\left(2 \times \pi \times \left(\frac{n}{N} - 0.5\right)\right) ]</td>
<td>(0 \leq n \leq N) otherwise</td>
</tr>
<tr>
<td></td>
<td>[ w(n) = 0 ]</td>
<td></td>
</tr>
</tbody>
</table>

![Graph of 'bartletthann' window](image)

| 'blackman'      | \[ w(n) = 0.42 \times 0.50 \times \cos\left(2 \times \pi \times \frac{n}{N}\right) + 0.08 \times \cos\left(4 \times \pi \times \frac{n}{N}\right) \] | \(0 \leq n \leq N\) otherwise |
|                 | \[ w(n) = 0 \]                                                                        |            |

![Graph of 'blackman' window](image)
### WaveScan User Guide

**Calculator Functions**

<table>
<thead>
<tr>
<th>Window</th>
<th>Equation and Example</th>
<th>Where</th>
</tr>
</thead>
<tbody>
<tr>
<td>'blackmanharris</td>
<td>[ w(n) = 0.35875 - 0.48829 \times \cos(2\times pi \times \frac{n}{N}) + 0.14128 \times \cos(4\times pi \times \frac{n}{N}) - 0.01168 \times \cos(6\times pi \times \frac{n}{N}) ]</td>
<td>( 0 \leq n \leq N ) otherwise</td>
</tr>
<tr>
<td></td>
<td>( w(n) = 0 )</td>
<td></td>
</tr>
<tr>
<td>'cosine2</td>
<td>[ w(n) = 0.5 - 0.5 \times \cos(2\times pi \times \frac{n}{N}) ]</td>
<td>( 0 \leq n \leq N ) otherwise</td>
</tr>
<tr>
<td></td>
<td>( w(n) = 0 )</td>
<td></td>
</tr>
</tbody>
</table>
### Window Equation and Example

<table>
<thead>
<tr>
<th>Window</th>
<th>Equation and Example</th>
<th>Where</th>
</tr>
</thead>
<tbody>
<tr>
<td>'cosine4</td>
<td>[ w(n) = \left( 0.5 - 0.5 \times \cos \left( 2 \times \frac{\pi}{N} \times \frac{n}{N} \right) \right)^2 ]</td>
<td>[ 0 \leq n \leq N ]</td>
</tr>
<tr>
<td></td>
<td>[ w(n) = 0 ]</td>
<td>otherwise</td>
</tr>
<tr>
<td>'extcosbell</td>
<td>[ w(n) = 0.5 - 0.5 \times \cos \left( 10 \times \frac{\pi}{N} \times \frac{n}{N} \right) ]</td>
<td>[ \text{abs}(n/N - 0.5) &gt; 0.4 ]</td>
</tr>
<tr>
<td></td>
<td>[ w(n) = 1 ]</td>
<td>otherwise</td>
</tr>
</tbody>
</table>
## WaveScan User Guide

### Calculator Functions

<table>
<thead>
<tr>
<th>Window</th>
<th>Equation and Example</th>
<th>Where</th>
</tr>
</thead>
</table>
| 'flattop    | \[
|             | \begin{align*}
|             | w(n) &= 1 - 1.93 \times \cos\left(2 \times \pi \cdot \frac{n}{N}\right) + 1.29 \times \cos\left(4 \times \pi \cdot \frac{n}{N}\right) - 0.388 \\
|             | & \times \cos\left(6 \times \pi \cdot \frac{n}{N}\right) + 0.322 \times \cos\left(8 \times \pi \cdot \frac{n}{N}\right) \\
|             | w(n) &= 0 \\
|             | \end{align*}
|             | otherwise \ 0 \leq n \leq N                                                         |                            |

| 'halfcyclesine | \[
|                | \begin{align*}
|                | w(n) &= \sin\left(\pi \cdot \frac{n}{N}\right) \\
|                | w(n) &= 0 \quad 0 \leq n \leq N \quad \text{otherwise} \\

![Graph of 'flattop window](image1)

![Graph of 'halfcyclesine window](image2)
<table>
<thead>
<tr>
<th>Window</th>
<th>Equation and Example</th>
<th>Where</th>
</tr>
</thead>
<tbody>
<tr>
<td>'half3cyclesine</td>
<td>$w(n) = \left( \sin(\pi \times \frac{n}{N}) \right)^3$</td>
<td>$0 \leq n \leq N$</td>
</tr>
<tr>
<td>and</td>
<td></td>
<td></td>
</tr>
<tr>
<td>'halfcyclesine3</td>
<td>$w(n) = 0$</td>
<td>otherwise</td>
</tr>
<tr>
<td>'half6cyclesine</td>
<td>$w(n) = \left( \sin(\pi \times \frac{n}{N}) \right)^6$</td>
<td>$0 \leq n \leq N$</td>
</tr>
<tr>
<td>and</td>
<td></td>
<td></td>
</tr>
<tr>
<td>'halfcyclesine6</td>
<td>$w(n) = 0$</td>
<td>otherwise</td>
</tr>
<tr>
<td>Window</td>
<td>Equation and Example</td>
<td>Where</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------</td>
<td>-------</td>
</tr>
<tr>
<td>'hamming</td>
<td>$w(n) = 0.54 - 0.46 \times \cos\left(2 \times \pi \times \frac{n}{N}\right)$</td>
<td>$0 \leq n \leq N$</td>
</tr>
<tr>
<td></td>
<td>$w(n) = 0$</td>
<td>otherwise</td>
</tr>
</tbody>
</table>

![Graph of 'hamming window](image1)

<table>
<thead>
<tr>
<th>Window</th>
<th>Equation and Example</th>
<th>Where</th>
</tr>
</thead>
<tbody>
<tr>
<td>'hanning</td>
<td>$w(n) = 0.5 - 0.5 \times \cos\left(2 \times \pi \times \frac{n}{N}\right)$</td>
<td>$0 \leq n \leq N$</td>
</tr>
<tr>
<td></td>
<td>$w(n) = 0$</td>
<td>otherwise</td>
</tr>
</tbody>
</table>

![Graph of 'hanning window](image2)
<table>
<thead>
<tr>
<th>Window</th>
<th>Equation and Example</th>
<th>Where</th>
</tr>
</thead>
<tbody>
<tr>
<td>'nuttall</td>
<td>[ w(n) = 0.3635819 - 0.4891775 \times \cos\left(2 \times \frac{\pi \times n}{N}\right) + 0.1365995 \times \cos\left(4 \times \frac{\pi \times n}{N}\right) - 0.0106411 \times \cos\left(6 \times \frac{\pi \times n}{N}\right) ] [ w(n) = 0 ] [ \text{otherwise} ]</td>
<td>[ 0 \leq n \leq N ]</td>
</tr>
<tr>
<td>'parzen</td>
<td>[ w(n) = 1 - 6 \times \text{abs}\left(2 \times \frac{n}{N} - 1\right) + 6 \times \text{abs}\left(2 \times \frac{n}{N} - 1\right) ] [ \text{abs}(2^{\ast}n/N-1) \leq 0.5 ] [ w(n) = 2 \times \text{abs}\left(2 \times \frac{n}{N} - 1\right) ] [ \text{otherwise} ]</td>
<td></td>
</tr>
<tr>
<td>'triangular</td>
<td>Same as 'bartlett.</td>
<td></td>
</tr>
</tbody>
</table>
x**2

Returns the $x^2$ value. This function is available only in the SKILL mode.
**xmax**

Computes the value of the independent variable x at which the expression attains its maximum value, that is, the value of x that maximizes y=f(x).

The maximum might occur at more than one point on the X-axis, so you must choose (in the Nth Maximizer field) which maximum value you want to see. The calculator returns the value of the Nth Maximizer counting from the left, that is, toward increasing X-axis values. If you enter a negative integer, the direction of search is reversed toward decreasing X-axis values (counting from the right).

This function is available only in the SKILL mode.

- *signal* is the name of the signal.
- *Nth Maximizer* is the maximum value of the expression.
**xmin**

Computes the value of the independent variable x at which the expression has its minimum value, that is, the value of x that minimizes y=f(x).

The minimum might occur at more than one point on the x axis, so you must choose (in the Nth Minimizer field) which minimum value you want to see. The calculator returns the value of the Nth Minimizer, counting from the left, that is, toward increasing X-axis values. If you enter a negative integer, the direction of search is reversed toward decreasing X-axis values (counting from the right).

This function is available only in the SKILL mode.

- *signal* is the name of the signal.
- *Nth Minimizer* is the minimum value of the expression.
**xval**

Returns the vector consisting of the X-axis values of the points in the signal.

**Example**

The following diagram illustrates this function.
y**x

Returns the $y^x$ value. This function is available only in the SKILL mode.
ymax

Computes the maximum Y-axis value of the expression y=f(x). This function is available only in the SKILL mode.
**ymin**

Computes the minimum Y-axis value of the expression \( y = f(x) \). This function is available only in the SKILL mode.
yval

Returns a vector consisting of the Y-axis values of the points in the signal. This function can also calculate the ordinate value at a specified abscissa value. This function is available only in the MDL mode.

Example

\[ yvalOut = \max ( \ yval( \ V(out) ) ) \]
Constants

This chapter lists constants and their definitions for the SKILL and MDL modes.

Table E-1  Constants in the SKILL Mode

<table>
<thead>
<tr>
<th>Constant</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boltzmann</td>
<td>1.380622e-23</td>
</tr>
<tr>
<td>charge</td>
<td>1.6021917e-19</td>
</tr>
<tr>
<td>degPerRad</td>
<td>57.2957795130823</td>
</tr>
<tr>
<td>epp0</td>
<td>8.854e-12</td>
</tr>
<tr>
<td>pi</td>
<td>3.14159265358979323846</td>
</tr>
<tr>
<td>sqrt2</td>
<td>1.41421356237309504880</td>
</tr>
<tr>
<td>twoPi</td>
<td>6.28318530717958647688</td>
</tr>
</tbody>
</table>

Table E-2  Constants in the MDL Mode

**Integer Constants**

<table>
<thead>
<tr>
<th>Constant</th>
<th>Definition</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>Boolean true</td>
<td>1</td>
</tr>
<tr>
<td>no</td>
<td>Boolean false</td>
<td>0</td>
</tr>
</tbody>
</table>

**Real Mathematical Constants**

<table>
<thead>
<tr>
<th>Constant</th>
<th>Definition</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>pi</td>
<td>π</td>
<td>3.14159265</td>
</tr>
<tr>
<td>e</td>
<td>e</td>
<td>2.71828183</td>
</tr>
<tr>
<td>inf</td>
<td>∞</td>
<td>infinity</td>
</tr>
<tr>
<td>nan</td>
<td>Not a number (result of an invalid operation)</td>
<td>NaN</td>
</tr>
</tbody>
</table>
## Real Physical Constants

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>q</td>
<td>Charge of an electron</td>
<td>$1.6021918\times10^{-19}$ C</td>
</tr>
<tr>
<td>c</td>
<td>Speed of light</td>
<td>$2.99792458\times10^8$ m/s</td>
</tr>
<tr>
<td>k</td>
<td>Boltzmann’s constant</td>
<td>$1.3806226\times10^{-23}$ J/K</td>
</tr>
<tr>
<td>h</td>
<td>Planck’s constant</td>
<td>$6.6260755\times10^{-34}$ J-s</td>
</tr>
<tr>
<td>eps0</td>
<td>Permittivity of a vacuum</td>
<td>$8.85418792394420013968\times10^{-12}$ F/m</td>
</tr>
<tr>
<td>epsrsi</td>
<td>Relative permittivity of silicon</td>
<td>11.7</td>
</tr>
<tr>
<td>u0</td>
<td>Permeability of a vacuum</td>
<td>$\pi \times 4.0\times10^{-7}$ H/m</td>
</tr>
<tr>
<td>celsius0</td>
<td>0 celsius</td>
<td>273.15 K</td>
</tr>
<tr>
<td>micron</td>
<td></td>
<td>$10^{-6}$ m</td>
</tr>
<tr>
<td>angstrom</td>
<td></td>
<td>$10^{-10}$ m</td>
</tr>
<tr>
<td>avogadro</td>
<td>Avogadro’s number</td>
<td>$6.022169\times10^{23}$</td>
</tr>
<tr>
<td>logic0</td>
<td>The value of logic 0</td>
<td>0</td>
</tr>
<tr>
<td>logic1</td>
<td>The value of logic 1</td>
<td>5</td>
</tr>
</tbody>
</table>
Index

Numerics

1/x 292
10**x 293
angle 297
argmax 298
argmin 299
asin 300
asinh 301
assigning keys 148
atan 302
atanh 303
avg 304
b1f 305
bw (bandwidth) 306
categories
  MDL 148
  SKILL 145
ceil 308
cfft 309
clip 310
compression 312
compressionVRI 313
conj 314
convolve 315
cos 316
cplx 318
cross 319
crosscor 321
crosses 322
d2r 324
db10 325
db20 326
dbm 327
defining 146
delay 328
deltax 329
deriv 331
dft 332
dftbb 335
dutycycle 337
dutycycles 338
exp 340
eyeDiagram 341
falltime 342
fft 344
filtering 149
flip 346
floor 347
foo 348

A

abs 294
acos 295
acosh 296
adding labels 89
angle 297
argmax 298
argmin 299
asin 300
asinh 301
atan 302
atanh 303
avg 304
b1f 305
bw (bandwidth) 306
categories
  MDL 148
  SKILL 145
ceil 308
cfft 309
clip 310
compression 312
compressionVRI 313
conj 314
convolve 315
cos 316
cplx 318
cross 319
crosscor 321
crosses 322
d2r 324
db10 325
db20 326
dbm 327
defining 146
delay 328
deltax 329
deriv 331
dft 332
dftbb 335
dutycycle 337
dutycycles 338
exp 340
eyeDiagram 341
falltime 342
fft 344
filtering 149
flip 346
floor 347
foo 348

B

b1f 305
bw (bandwidth) 306

C

caculator
  expressions, building
    example 142

calculator
  constants 149
  expressions
    building, algebraic mode 142
    building, RPN mode 140
    printing 149
  functions
    1/x 292
    10**x 293
    abs 294
    acos 295
    acosh 296
WaveScan User Guide

fourEval 349
freq 350
freq_jitter 352
ga 354
gac_freq 355
gac_gain 356
gainBwProd 357
gainmargin 358
getAsciiWave 359
gmax 360
gmin 361
gmsg 361, 362
gp 363
gpc_freq 364
gpc_gain 365
groupdelay 366
gt 368
gumx 369
harmonic 370
harmonicFreq 371
histo 372
ifft 373
iinteg 375
im 377
int 379
integ 380
ipn 381
ipnVRI 382
kf 383
ln 384
log10 385
lsb 386
lshift 387
mag 388
max 389
min 390
mod 391
movingavg 392
nc_freq 393
nc_gain 394
nf 395
nfmin 396
overshoot 397
period_jitter 399
ph 400
phasemargin 401
phaseNoise 402
pow 403
pp 404
psd 405
psdbb 407
pzbode 408
pzfilter 409
r2d 411
re 412
real 413
risetime 414
rms 417
round 422
s11 423
s12 424
s21 425
s22 426
sample 427
settlingTime 429
sign 431
sin 432
sinh 433
slewrate 434
snr 435
spectralPower 437
sqrt 438
ssb 439
sthist 440
stddev 441
tan 442
tangentLine 443
tanh 444
thd 445
trim 447
value 449
window 450
x**2 458
xmax 459
xmin 460
xval 461
y**x 462
ymin 463
yval 465
memories

 commencement
 creating
 deleting
 editing
 loading
 saving
 undeleting
 menu descriptions

 status bar area

June 2004 470 Product Version 5.1.41
tool bar description 163
variables
copying 158
creating 157
deleting 158
editing 158
loading 160
saving 159
undeleting 159
cdsenv file, variables
application
envdirectory 197
exitdialog 196
graphattributesfile 195
readstate 193
statefile 192
writestate 194
axis
foreground 237
significantDigits 238
axis, dependent
forceOrigin 234
logScale 233
axis, independent
forceOrigin 236
logScale 235
calculator
height 270
notation 272
rpnMode 264
significantDigits 271
stackSize 265, 266
useSkill 273
width 269
xLocation 267
yLocation 268
circular graph
background 215
foreground 217
circular scale
admittForeground 242
cursorScale 244
impedForeground 241
polarForeground 239
significantDigits 243
filter
textFilterType 205
graph
cursorOn 223
defaultSubtitle 222
displayDate 221
dName 229
fontSize 231
fontStyle 230
majGridsOn 224
majorForeground 226
minGridsOn 225
minorForeground 227
snapOn 228
stripChartOn 232
subTitle 220
suffix 219
visibleLegendRows 218
graph frame
height 208
width 206
label
direction 262
dName 263
fontName 263
fontStyle 263
foreground 260
significantDigits 263
marker
arrowOn 256
dName 257
fontName 259
fontStyle 258
foreground 254
rectangular graph
background 210
foreground 212
visibleDigitalRows 213
visibleStripChartRows 214
Results Browser
dataDirHome 204
historyLength 202
plotstyle 203
table
cellPadding 277
fileSuffix 275
height 274
location 280
maxColumnWidth 278
notation 279
significantDigits 281
stringColumnWidth 276
width 273
trace
dToAConversion 251
dToAUnit 252
fontStyle 251
<table>
<thead>
<tr>
<th>HarmonicFreq 371</th>
</tr>
</thead>
<tbody>
<tr>
<td>hiding</td>
</tr>
<tr>
<td>objects 58</td>
</tr>
<tr>
<td>histo 372</td>
</tr>
<tr>
<td>horizontal cursor 85</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>ifft 373</td>
</tr>
<tr>
<td>integ 375</td>
</tr>
<tr>
<td>im 377</td>
</tr>
<tr>
<td>int 379</td>
</tr>
<tr>
<td>integ 380</td>
</tr>
<tr>
<td>ipn 381</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>K</th>
</tr>
</thead>
<tbody>
<tr>
<td>kf 383</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>labels adding 89</td>
</tr>
<tr>
<td>editing 90</td>
</tr>
<tr>
<td>moving 93</td>
</tr>
<tr>
<td>ln 384</td>
</tr>
<tr>
<td>log10 385</td>
</tr>
<tr>
<td>lsb 386</td>
</tr>
<tr>
<td>lshift 387</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>mag 388</td>
</tr>
<tr>
<td>markers delta marker placing 97</td>
</tr>
<tr>
<td>editing 100</td>
</tr>
<tr>
<td>moving 101</td>
</tr>
<tr>
<td>max 389</td>
</tr>
<tr>
<td>min 390</td>
</tr>
<tr>
<td>mod 391</td>
</tr>
<tr>
<td>moving columns 173</td>
</tr>
<tr>
<td>labels 93</td>
</tr>
<tr>
<td>markers 101</td>
</tr>
<tr>
<td>movingavg 392</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>nc_freq 393</td>
</tr>
<tr>
<td>nc_gain 394</td>
</tr>
<tr>
<td>newlink status 164</td>
</tr>
<tr>
<td>nf 395</td>
</tr>
<tr>
<td>nfmin 396</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>O</th>
</tr>
</thead>
<tbody>
<tr>
<td>objects deleting 59</td>
</tr>
<tr>
<td>hiding 58</td>
</tr>
<tr>
<td>revealing 58</td>
</tr>
<tr>
<td>swapping 59</td>
</tr>
<tr>
<td>opening Results Browser 24</td>
</tr>
<tr>
<td>overshoot 397</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>panning graphs 62</td>
</tr>
<tr>
<td>period_jitter 399</td>
</tr>
<tr>
<td>ph 400</td>
</tr>
<tr>
<td>phasemargin 401</td>
</tr>
<tr>
<td>phaseNoise 402</td>
</tr>
<tr>
<td>pow 403</td>
</tr>
<tr>
<td>pp 404</td>
</tr>
<tr>
<td>printing graphs 105</td>
</tr>
<tr>
<td>tables 176</td>
</tr>
<tr>
<td>psd 405</td>
</tr>
<tr>
<td>psdbb 407</td>
</tr>
<tr>
<td>pzbode 408</td>
</tr>
<tr>
<td>pzfilter 409</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>r2d 411</td>
</tr>
<tr>
<td>re 412</td>
</tr>
<tr>
<td>real 413</td>
</tr>
<tr>
<td>Report Table window about the 42, 166 menu descriptions 183</td>
</tr>
<tr>
<td>restoring session 22</td>
</tr>
<tr>
<td>Results Browser icon descriptions 46 menu descriptions 24</td>
</tr>
</tbody>
</table>
opening 24
results, selecting in 26
revealing objects 58
risetime 414
rms 417
round 422

S

s12 424
sample 427
Save Session Dialog, about the 22
saving graphs 109
session 21
traces 103
scale attributes, editing 80
session saving 22
settlingTime 429
shell environment variables
  CDS_wavescan_JRE_DIR 284
  memory 285
sign 431
signals plotting 28
sin 432
sinh 433
slewrate 434
snr 435
spectralPower 437
sqrt 438
ssb 439
stathisto 440
stddev 441
supported data formats 16
swapping objects 59

T

tables
  attributes, formatting 170
columns
    sorting 176
    width, changing 173
columns, moving 173
creating 42
printing 176
transposing rows and columns 173
tan 442
tangent Line 443
tanh 444
thd 445
trace
  attributes, editing 71
traces saving 103
trim 447

V

value 449
vertical cursor 85

W

Wavescan windows 16
window 450

X

x**2 458
xmax 459
xmin 460
xval 461

Y

y**x 462
ymax 463
ymin 464
yval 465

Z

zooming graphs 59