

Simulation Environment Help

**Product Version ICADVM20.1
October 2020**

© 2020 Cadence Design Systems, Inc. All rights reserved.
Printed in the United States of America.

Cadence Design Systems, Inc. (Cadence), 2655 Seely Ave., San Jose, CA 95134, USA.

Open SystemC, Open SystemC Initiative, OSCI, SystemC, and SystemC Initiative are trademarks or registered trademarks of Open SystemC Initiative, Inc. in the United States and other countries and are used with permission.

Trademarks: Trademarks and service marks of Cadence Design Systems, Inc. contained in this document are attributed to Cadence with the appropriate symbol. For queries regarding Cadence's trademarks, contact the corporate legal department at the address shown above or call 800.862.4522. All other trademarks are the property of their respective holders.

Restricted Permission: This publication is protected by copyright law and international treaties and contains trade secrets and proprietary information owned by Cadence. Unauthorized reproduction or distribution of this publication, or any portion of it, may result in civil and criminal penalties. Except as specified in this permission statement, this publication may not be copied, reproduced, modified, published, uploaded, posted, transmitted, or distributed in any way, without prior written permission from Cadence. Unless otherwise agreed to by Cadence in writing, this statement grants Cadence customers permission to print one (1) hard copy of this publication subject to the following conditions:

1. The publication may be used only in accordance with a written agreement between Cadence and its customer.
2. The publication may not be modified in any way.
3. Any authorized copy of the publication or portion thereof must include all original copyright, trademark, and other proprietary notices and this permission statement.
4. The information contained in this document cannot be used in the development of like products or software, whether for internal or external use, and shall not be used for the benefit of any other party, whether or not for consideration.

Disclaimer: Information in this publication is subject to change without notice and does not represent a commitment on the part of Cadence. Except as may be explicitly set forth in such agreement, Cadence does not make, and expressly disclaims, any representations or warranties as to the completeness, accuracy or usefulness of the information contained in this document. Cadence does not warrant that use of such information will not infringe any third party rights, nor does Cadence assume any liability for damages or costs of any kind that may result from use of such information.

Restricted Rights: Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or its successor.

Contents

<u>Preface</u>	9
<u>Scope</u>	10
<u>Licensing Requirements</u>	10
<u>Related Documentation</u>	10
<u>What's New and KPNS</u>	10
<u>Installation, Environment, and Infrastructure</u>	10
<u>Technology Information</u>	10
<u>Virtuoso Tools</u>	11
<u>Additional Learning Resources</u>	12
<u>Video Library</u>	12
<u>Virtuoso Videos Book</u>	12
<u>Rapid Adoption Kits</u>	13
<u>Help and Support Facilities</u>	13
<u>Customer Support</u>	14
<u>Feedback about Documentation</u>	14
<u>Typographic and Syntax Conventions</u>	15
<u>1</u>	
<u>Setting Up SE Help</u>	17
<u>About SE Help</u>	17
<u>Finding the Information You Want</u>	18
<u>Before You Can Run a Simulation</u>	19
<u>Displaying the Simulation Menu</u>	19
<u>Initializing a New Run Directory in the Graphical Environment</u>	19
<u>Setting Up Simulation in the UNIX Environment</u>	21
<u>Running si in Replay Mode</u>	22
<u>Setting Up Remote Simulation</u>	23
<u>Customizing Simulation</u>	23
<u>Using the Simulation Run Control File To Customize SE</u>	23
<u>Customizing Netlisting</u>	25
<u>Customizing Scale Factors</u>	25

2

Creating the Input Stimulus 27

Creating the Input Stimulus in the Control File 27

Using Substitution Functions 28

3

Customizing Netlisting 31

SE Netlisting 31

Specifying a Hierarchy of Netlisting Views 32

Switching Views 32

Selecting a Netlisting View from a Hierarchy 32

Overriding Default View and Stop Lists 32

Controlling Renetlisting 33

4

Running a Simulation 35

Choosing Simulation Run Options 35

Simulation in the Graphical Environment 36

Simulation in the UNIX Environment 38

Full Simulation 38

Simulation in Steps 39

simInitRunDir 39

netlist 39

simin 40

runsim 40

exit 40

Simulation in Batch Mode 40

Interactive Simulation 41

Netlist and Simulate Form 43

Simulation Environment Options Form 44

5

<u>Displaying Results</u>	45
<u>Displaying Waveform Results in the Graphical Environment</u>	45
<u>Displaying Waveform Results in Register Form</u>	46
<u>Displaying Text Results</u>	47
<u>Displaying Netlisting Errors for Specific Nets or Instances</u>	47
<u>Displaying Other Netlisting Errors (SILOS II Only)</u>	47
<u>Displaying a Specified Text File</u>	48

6

<u>Controlling Job Status</u>	49
<u>Accessing the Job Monitor Form</u>	49
<u>Checking Current Simulation Status</u>	50
<u>Adjusting Job Priority</u>	50
<u>Terminating a Simulation</u>	51
<u>Interrupting or Restarting a Simulation</u>	52
<u>Editing the Job Monitor Form</u>	52

7

<u>SE Functions Reference</u>	53
<u>Initialize</u>	53
<u>Initialize Forms</u>	54
<u>Using Initialize for a New Run Directory</u>	54
<u>Using Initialize for an Existing Run Directory</u>	55
<u>Top-Level SKILL Command</u>	55
<u>Options</u>	55
<u>Prerequisites</u>	55
<u>Options Form</u>	56
<u>Using Options</u>	57
<u>Stimulus – Edit File</u>	57
<u>Prerequisites</u>	57
<u>Edit File Form</u>	57
<u>Netlist/Simulate</u>	58
<u>Prerequisites</u>	58

Simulation Environment Help

<u>Netlist/Simulate Form</u>	59
<u>Using Netlist/Simulate</u>	60
<u>Top-Level SKILL Command</u>	60
<u>Interactive</u>	60
<u>Prerequisites</u>	61
<u>Using Interactive</u>	61
<u>Top-Level SKILL Command</u>	61
<u>Show Outputs – Show Run Log</u>	61
<u>Prerequisites</u>	61
<u>Using Show Run Log</u>	62
<u>Show Outputs – Show Output</u>	62
<u>Prerequisites</u>	62
<u>Using Show Output</u>	62
<u>Show Outputs – Show Global Error</u>	62
<u>Prerequisites</u>	63
<u>Using Show Global Errors</u>	63
<u>Show Outputs – Highlight Errors</u>	63
<u>Prerequisites</u>	63
<u>Using Highlight Errors</u>	63
<u>Show Outputs – Show Run File</u>	64
<u>Prerequisites</u>	64
<u>Show Run File Form</u>	64
<u>Using Show Run File</u>	64
<u>Show Waveforms</u>	65
<u>Prerequisites</u>	65
<u>Show Waveforms Form</u>	65
<u>Using Show Waveforms</u>	65
<u>Top-Level SKILL Command</u>	66
<u>Show Registers</u>	66
<u>Prerequisites</u>	66
<u>Show Registers Form</u>	66
<u>Using Show Registers</u>	67
<u>Job Monitor</u>	67
<u>Prerequisites</u>	67
<u>Job Monitor Form</u>	67
<u>Using Job Monitor</u>	68

Simulation Environment Help

<u>Top-Level SKILL Command</u>	69
--------------------------------------	----

8

<u>Sample Files</u>	71
<u>Sample control File</u>	71
<u>Sample si.inp File</u>	72
<u>Sample si.inp File Generated for Cadence SILOS II</u>	73

Simulation Environment Help

Preface

This document describes the Virtuoso[®] Simulation Environment (SE). This environment lets you run simulations from the graphical interface or from a command line. Virtuoso SE supports user-defined simulators and the following standard simulators:

- System HILO
- HSPICE
- Verilog-XL Simulator

This document is aimed at designers who want to netlist and simulate designs maintained in the Virtuoso design environment and assumes that you are familiar with:

- The Virtuoso design environment and application infrastructure mechanisms designed to support consistent operations between all Cadence tools.
- The applications used to design and develop integrated circuits in the Virtuoso design environment, notably Virtuoso Schematic Editor.

This preface contains the following topics:

- [Scope](#)
- [Licensing Requirements](#)
- [Related Documentation](#)
- [Additional Learning Resources](#)
- [Customer Support](#)
- [Feedback about Documentation](#)
- [Typographic and Syntax Conventions](#)

Scope

Unless otherwise noted, the functionality described in this guide can be used in both mature node (for example, IC6.1.8) and advanced node (for example, ICADVM20.1) releases.

Label	Meaning
(ICADVM20.1 Only)	Features supported only in the ICADVM20.1 advanced nodes and advanced methodologies release.
(IC6.1.8 Only)	Features supported only in mature node releases.

Licensing Requirements

For information on licensing in the Virtuoso Simulation Environment, see [*Virtuoso Software Licensing and Configuration Guide*](#).

Related Documentation

What's New and KPNS

- [*Simulation Environment What's New*](#)
- [*Simulation Environment Known Problems and Solutions*](#)

Installation, Environment, and Infrastructure

- [*Cadence Installation Guide*](#)
- [*Virtuoso Design Environment User Guide*](#)
- [*Virtuoso Design Environment SKILL Reference*](#)
- [*Cadence Application Infrastructure User Guide*](#)

Technology Information

- [*Virtuoso Technology Data User Guide*](#)

- [*Virtuoso Technology Data ASCII Files Reference*](#)
- [*Virtuoso Technology Data SKILL Reference*](#)

Virtuoso Tools

IC6.1.8 Only

- [*Virtuoso Layout Suite L User Guide*](#)
- [*Virtuoso Layout Suite XL User Guide*](#)
- [*Virtuoso Layout Suite GXL Reference*](#)

ICADVM20.1 Only

- [*Virtuoso Layout Viewer User Guide*](#)
- [*Virtuoso Layout Suite XL: Basic Editing User Guide*](#)
- [*Virtuoso Layout Suite XL: Connectivity Driven Editing Guide*](#)
- [*Virtuoso Layout Suite EXL Reference*](#)
- [*Virtuoso Concurrent Layout User Guide*](#)
- [*Virtuoso Design Planner User Guide*](#)
- [*Virtuoso Multi-Patterning Technology User Guide*](#)
- [*Virtuoso Placer User Guide*](#)
- [*Virtuoso Simulation Driven Interactive Routing User Guide*](#)
- [*Virtuoso Width Spacing Patterns User Guide*](#)
- [*Virtuoso RF Solution Guide*](#)
- [*Virtuoso Electromagnetic Solver Assistant User Guide*](#)

IC6.1.8 and ICADVM20.1

- [*Virtuoso Abstract Generator User Guide*](#)
- [*Virtuoso Custom Digital Placer User Guide*](#)
- [*Virtuoso Design Rule Driven Editing User Guide*](#)

- [*Virtuoso Electrically Aware Design Flow Guide*](#)
- [*Virtuoso Floorplanner User Guide*](#)
- [*Virtuoso Fluid Guard Ring User Guide*](#)
- [*Virtuoso Interactive and Assisted Routing User Guide*](#)
- [*Virtuoso Layout Suite SKILL Reference*](#)
- [*Virtuoso Module Generator User Guide*](#)
- [*Virtuoso Parameterized Cell Reference*](#)
- [*Virtuoso Pegasus Interactive User Guide*](#)
- [*Virtuoso Space-based Router User Guide*](#)
- [*HDL Import and Netlist-to-Schematic Conversion SKILL Reference*](#)

Additional Learning Resources

Video Library

The [Video Library](#) on the Cadence Online Support website provides a comprehensive list of videos on various Cadence products.

To view a list of videos related to a specific product, you can use the *Filter Results* feature available in the pane on the left. For example, click the *Virtuoso Layout Suite* product link to view a list of videos available for the product.

You can also save your product preferences in the Product Selection form, which opens when you click the *Edit* icon located next to *My Products*.

Virtuoso Videos Book

You can access certain videos directly from Cadence Help. To learn more about this feature and to access the list of available videos, see [Virtuoso Videos](#).

Rapid Adoption Kits

Cadence provides a number of [Rapid Adoption Kits](#) that demonstrate how to use Virtuoso applications in your design flows. These kits contain design databases and instructions on how to run the design flow.

In addition, Cadence offers the following training courses of interest:

- [Virtuoso Analog Design Environment](#)
- [Virtuoso Schematic Editor](#)
- [Spectre Simulations Using Virtuoso ADE](#)
- [Virtuoso UltraSim Full-Chip Simulator](#)
- [Virtuoso Simulation for Advanced Nodes](#)

Cadence also offers the following training courses on the SKILL programming language, which you can use to customize, extend, and automate your design environment:

- [SKILL Language Programming Introduction](#)
- [SKILL Language Programming](#)
- [Advanced SKILL Language Programming](#)

To explore the full range of training courses provided by Cadence in your region, visit [Cadence Training](#) or write to training_enroll@cadence.com.

Note: The links in this section open in a separate web browser window when clicked in Cadence Help.

Help and Support Facilities

Virtuoso offers several built-in features to let you access help and support directly from the software.

- The Virtuoso *Help* menu provides consistent help system access across Virtuoso tools and applications. The standard Virtuoso *Help* menu lets you access the most useful help and support resources from the Cadence support and corporate websites directly from the CIW or any Virtuoso application.
- The Virtuoso Welcome Page is a self-help launch pad offering access to a host of useful knowledge resources, including quick links to content available within the Virtuoso installation as well as to other popular online content.

The Welcome Page is displayed by default when you open Cadence Help in standalone mode from a Virtuoso installation. You can also access it at any time by selecting *Help – Virtuoso Documentation Library* from any application window, or by clicking the *Home* button on the Cadence Help toolbar (provided you have not set a custom home page).

For more information, see Getting Help in *Virtuoso Design Environment User Guide*.

Customer Support

For assistance with Cadence products:

- **Contact Cadence Customer Support**

Cadence is committed to keeping your design teams productive by providing answers to technical questions and to any queries about the latest software updates and training needs. For more information, visit <https://www.cadence.com/support>.

- **Log on to Cadence Online Support**

Customers with a maintenance contract with Cadence can obtain the latest information about various tools at <https://support.cadence.com>.

Feedback about Documentation

You can contact Cadence Customer Support to open a service request if you:

- Find erroneous information in a product manual
- Cannot find in a product manual the information you are looking for
- Face an issue while accessing documentation by using Cadence Help

You can also submit feedback by using the following methods:

- In the Cadence Help window, click the *Feedback* button and follow instructions.
- On the Cadence Online Support Product Manuals page, select the required product and submit your feedback by using the *Provide Feedback* box.

Typographic and Syntax Conventions

The following typographic and syntax conventions are used in this manual.

<i>text</i>	Indicates names of manuals, menu commands, buttons, and fields.
text	Indicates text that you must type exactly as presented. Typically used to denote command, function, routine, or argument names that must be typed literally.
<i>z_argument</i>	Indicates text that you must replace with an appropriate argument value. The prefix (in this example, <i>z_</i>) indicates the data type the argument can accept and must not be typed.
	Separates a choice of options.
{ }	Encloses a list of choices, separated by vertical bars, from which you must choose one.
[]	Encloses an optional argument or a list of choices separated by vertical bars, from which you may choose one.
[?argName <i>t_arg</i>]	Denotes a <i>key argument</i> . The question mark and argument name must be typed as they appear in the syntax and must be followed by the required value for that argument.
...	Indicates that you can repeat the previous argument.
	Used with brackets to indicate that you can specify zero or more arguments.
	Used without brackets to indicate that you must specify at least one argument.
, ...	Indicates that multiple arguments must be separated by commas.
=>	Indicates the values returned by a Cadence® SKILL® language function.
/	Separates the values that can be returned by a Cadence SKILL language function.

If a command-line or SKILL expression is too long to fit within the paragraph margins of this document, the remainder of the expression is moved to the next line and indented. In code excerpts, a backslash (\) indicates that the current line continues on to the next line.

Simulation Environment Help

Preface

Setting Up SE Help

In this chapter, you can find information about

- [About SE Help](#) on page 17
- [Finding the Information You Want](#) on page 18
- [Before You Can Run a Simulation](#) on page 19
- [Displaying the Simulation Menu](#) on page 19
- [Initializing a New Run Directory in the Graphical Environment](#) on page 19
- [Setting Up Simulation in the UNIX Environment](#) on page 21
- [Setting Up Remote Simulation](#) on page 23
- [Customizing Simulation](#) on page 23
- [Customizing Scale Factors](#) on page 25
- [Customizing Netlisting](#) on page 25

For information on SE SKILL APIs, see the [*Digital Design Netlisting and Simulation SKILL Reference*](#).

About SE Help

SE allows you to run simulations from the graphical interface or from a command line. In the graphical environment, you use SE menus and forms. In the nongraphical environment, you use SE commands in a UNIX[®] xterm window running *si* or you use SKILL commands in the [Command Interpreter Window \(CIW\)](#).

SE supports user-defined simulators and the following standard simulators:

- System HILO
- HSPICE

Simulation Environment Help

Setting Up SE Help

■ Verilog-XL Simulator

Finding the Information You Want

If you have used Simulation Environment before, but need detailed information about a specific Simulation Environment feature, click any menu choice in the following diagram to go to reference information about that choice.

Misc->Probe->	Simulation	
	Initialize...	
	Options...	
	Application Options...	
	Stimulus	
	Netlist/Simulate...	
	Interactive...	
	Show Outputs->	Show Stimulus Wavefor ms -
	Show Waveforms...	Edit File... -
	Show Registers...	Show Run Log-> Show Foreground Run Log
	Job Monitor...	Show Output Show Background Ru n Log
		Show Global Errors
		Highlight Errors
		Show Run File...

Application Options

This menu choice lets you view forms that are specific to your simulator. You must write the necessary SKILL code to activate this menu choice.

Show Foreground Run Log

This command lets you view the run log of the currently running foreground job.

Show Background Run Log

This command lets you view the run log of the currently running background job.

Before You Can Run a Simulation

After you complete your design, you extract it, correct errors, and save the design for simulation input. You must correct all errors you find during extraction before you simulate your design. For information on extraction, refer to *Virtuoso Schematic Editor L User Guide*.

Displaying the Simulation Menu

To bring up the Simulation menu when you are displaying a schematic

- In the Schematic window, choose *Launch — Plugins — Simulation — Other*.

The system changes the menu banner to include the *Simulation* menu. This menu has the commands you need to simulate your design.

Initializing a New Run Directory in the Graphical Environment

The first step in simulation is setting up the simulation environment. When you initialize the environment, you specify the following:

- Design
- Simulator
- Simulation run directory

Initially, the system turns off all commands on the Simulation menu except Initialize. After you run the Initialize command, you can use the remaining simulation commands.

Simulation Environment Help

Setting Up SE Help

To initialize a new simulation run directory

1. In the Schematic window, select Simulation – Initialize.

The following form appears:



2. Type the name of the simulation run directory.

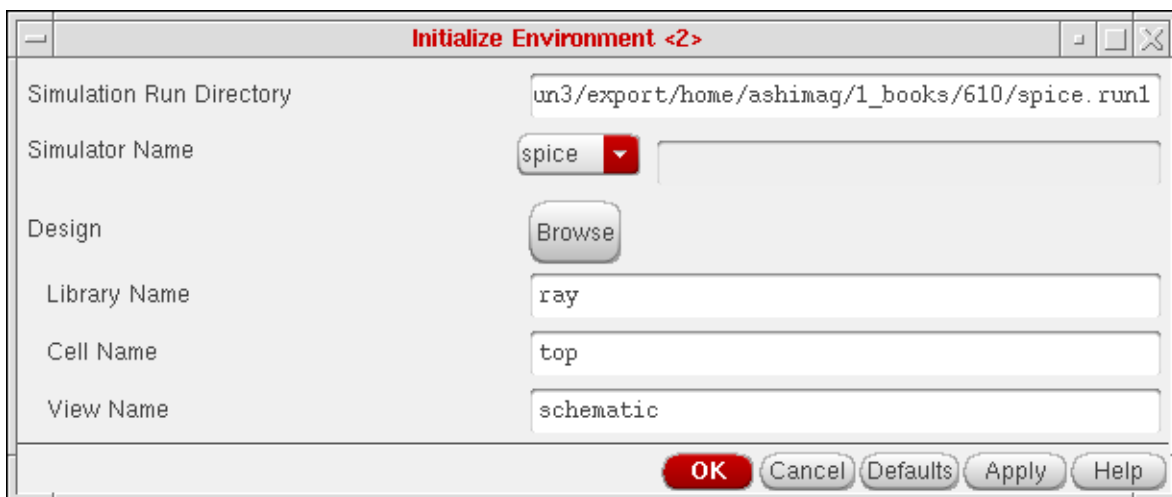
You can type either a full or a relative path. If you type a relative path, the system puts the run directory under the directory in which you started the software. The default run directory is *spice.run1*.

The system stores all simulation input and output files in the simulation run directory.

As the system initializes the environment, it lists the files it has loaded and any overridden variables in the *si.foregnd.log* file

3. Click OK.

The following form replaces the first [Initialize form](#):



The values on this form are the current window and default SE values. You can edit these fields by typing in the form or by using the [Browser](#).

4. From the Simulator Name cyclic field, select the simulator.

Simulation Environment Help

Setting Up SE Help

If you use a simulator that is not listed, select *other* and type the name of the simulator in the adjoining text entry field. You must select *other* before you can type in the text entry field.

5. Enter the name of the library containing the top level of your design.
6. Enter the cell name of the top level of your design.
7. Enter the view name of your design (for example, schematic).
8. Click *OK*.

Setting Up Simulation in the UNIX Environment

Cadence recommends that you run simulation using the menus and forms in the graphical environment. However, you can also run a simulation using SE commands in the *CIW* or in UNIX using the *si* binary. With these commands, you can run simulation in either interactive or batch mode.

Before you start a simulation in the UNIX environment, you must create the following files in your simulation run directory:

- si.env
- control

If you run simulation in UNIX, you must create the *si.env* file. (This file is created automatically when you use the graphical environment.) The *si.env* file tells SE which design to simulate and what simulator to use. The following table lists the variables you must define in the *si.env* file. Each interface might store additional specific variables in the *si.env* file

Simulation Environment Help

Setting Up SE Help

Required Properties in si.env.

Variable	Description
<u>simSimulator</u>	Simulator to be run
<u>simLibName</u>	Name of the library containing the top-level cellview
<u>simCellName</u>	Name of the top-level cell to be simulated
<u>simViewName</u>	View name of the top-level cell to be simulated

Optional Property in si.env

Variable	Description
<u>simHost</u>	Host name of remote simulator

The following is a sample `si.env` file:

```
simLibName = "testLib"
simCellName = "74169"
simViewName = "schematic"
simSimulator = "silos"
simHost = "cds642"
```

To set up SE to run simulations in the UNIX environment, follow the steps below:

1. Change to the directory that will contain the simulation run directory.
2. Create the run directory using the UNIX command `mkdir directoryname`, where `directoryname` is the name of the simulation run directory. For example, if your simulation run directory is `spice.run1`, type the following:

```
mkdir spice.run1
```

3. Change to the newly created directory.

```
cd spice.run1
```

4. Create the simulation environment file [*si.env*](#).

Running si in Replay Mode

The `simIlSleep()` function allows you to specify time for which you need to suspend the running process. It is recommended to use the `simIlSleep()` function, instead of the `ipcSleep()` and `sleep()` functions, while running `si` in the replay mode.

Note: The `ipcSleep()` and `sleep()` functions no longer work with the replay mode.

For details on the `simIlSleep()` function, see [*simIlSleep*](#) in the *Digital Design Netlisting and Simulation SKILL Reference*.

Setting Up Remote Simulation

You can set up the system to run remote simulation using the Verilog-XL and HSPICE simulators. The local machine and the remote host must both run the X Window System™.

To set up remote simulation, perform the following steps to define the necessary variables (you can set these variables using the [Options command](#) on the Simulation menu by typing them in the [CIW](#) or in the [.simrc file](#)):

1. Set the SE variable [simHost](#) to the name of the remote workstation.

For example:

```
simHost = "cds17"
```

2. Set the SE variable [simHostDiffers](#) to true (t) if the host computer has a different binary storage format than the local computer.

For example:

```
simHostDiffers = t
```

After you set these variables, you can run a remote simulation and view the results as if you were running the simulation locally.

Customizing Simulation

Using the Simulation Run Control File To Customize SE

When you initialize SE, it first loads the [si.env file](#). This file tells SE what design to simulate and what simulator to use.

SE then loads the simulation run control file `.simrc` if it exists. The `.simrc` file lets you override any netlisting or simulation environment variables. SE searches the following directories and loads the first `.simrc` file it finds:

```
$SIMRC/.simrc
$ossSimUserSiDir/.simrc
install_dir/tools/dfII/local/.simrc
./simrc
~/simrc
```

Simulation Environment Help

Setting Up SE Help

If you set a variable in `.simrc` that also sets options in the graphical environment (with the [Options command](#)), SE uses the `.simrc` file settings and ignores the Options settings.

The `.simrc` file must be in SKILL syntax. The following is a sample `.simrc` file. The first line overrides the default view list used for view switching with SILOS. The second line overrides the default stopping list that stops hierarchy expansion for SILOS.

```
hspiceSimViewList = ("hspice" "cmos_sch" "schematic")
hspiceSimStopList = ("hspice" "cmos_sch")
```

The following table shows you some SE variables you can set in your `.simrc` file to customize simulation. You can also see the [Open Simulation System Reference](#) for a complete list of SE variables

Variable	Description
<code>simSimulator</code>	Simulator to run
<code>simControlFile</code>	Path of default control file
<code>simDefaultControl</code>	Name of default <i>control</i> file if stored in <i>install_dir/etc/s</i>
<code>simTimeUnit</code>	Scaling factor for delay times. This value should match the first argument of the <i>deftiming</i> command.
<code>simCapUnit</code>	Scaling factor for capacitance
<code>simNlpGlobalLibName</code>	Name of library containing global formatting instructions for flat netlister
<code>simNlpGlobalCellName</code>	Name of cell containing global formatting instructions for flat netlister
<code>simNlpGlobalViewName</code>	Name of view of the cell containing global formatting instructions for flat netlister
<code>simNotIncremental</code>	Specifies incremental netlisting when set to <i>nil</i> .
<code>simReNetlistAll</code>	Specifies non-incremental netlisting
<code>simNetlistHier</code>	Specifies hierarchical netlisting

For a description of SKILL syntax and further information about SKILL functions, see the [Cadence SKILL Language User Guide](#) and the [Cadence SKILL Language Reference Manual](#).

Customizing Netlisting

The way the design hierarchy is traversed to produce the netlist and the syntax of the netlist depends on your simulator. For example, you might want the netlist for a Verilog simulation to be at the logic gate level because Verilog can simulate primitives such as AND gates and AOIs. You might want the netlist for a SPICE simulation of the same design to be at the transistor level because SPICE cannot simulate logic gates.

Click the topics below to go to information about [customizing netlisting](#).

- [SE Netlisting](#)
- [Specifying a Hierarchy of Netlisting Views](#)
- [Selecting a Netlisting View from a Hierarchy](#)
- [Overriding Default View and Stop Lists](#)
- [Controlling Renetlisting](#)

Customizing Scale Factors

The netlister can scale time and capacitance values. The scale factors for time and capacitance are defined by two SE variables: `simTimeUnit` and `simCapUnit`. With both variables, the value to be scaled is divided by the scale factor. The default value of variable `simTimeUnit` is 1e-9 (nanoseconds), and the default value of variable `simCapUnit` is 1e-15 (femtofarads). You can customize the scale factors by typing new `simTimeUnit` and `simCapUnit` values in your `.simrc` file.

Simulation Environment Help

Setting Up SE Help

Creating the Input Stimulus

In this chapter, you can find information about

- [Creating the Input Stimulus in the Control File](#) on page 27
- [Using Substitution Functions](#) on page 28

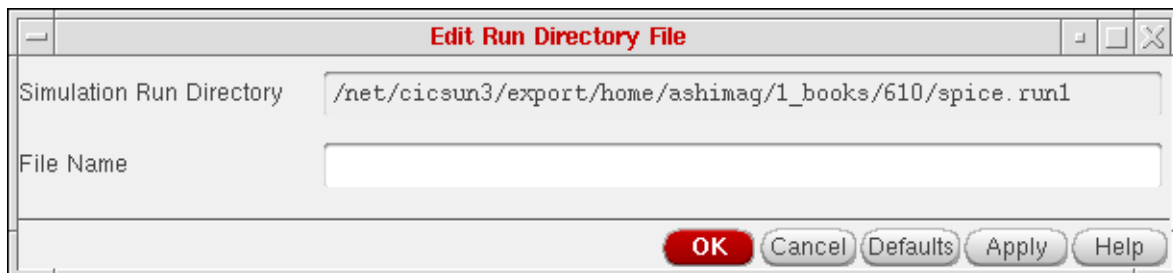
For information on SE SKILL APIs, see the [Digital Design Netlisting and Simulation SKILL Reference](#).

Creating the Input Stimulus in the Control File

To create an input stimulus:

1. In the Schematic window, select *Simulation – Stimulus – Edit File*.

The following form appears:



The simulation run directory is the one you specified with the [Initialize command](#). You cannot change the simulation run directory by editing this form. To change this directory, click *Cancel* and select the Initialize command again.

2. In the File Name field, type the name of the simulation run directory file you want to edit.

You do not need to type a full path.

Type `control` to edit the control file created in the simulation run directory by the *Initialize* command.

Simulation Environment Help

Creating the Input Stimulus

Note: You can edit or create any file in the simulation run directory by editing the File Name field.

3. Click OK.

The system brings up a new window displaying the file you specified.

4. Using a text editor, edit the file in the syntax of your analysis tool.

You can type all your changes directly into the control file or you can type SE substitution functions to translate names and merge in other files. For example, to merge an input stimulus file named `simsub.inp` into the `control` file, precede the file name with an exclamation point (!) and enclose the text in square brackets.

```
[!simsub.inp]
```

If `simsub.inp` is not in the current simulation run directory, type the full path as in the following example:

```
[!/<level1InstName/.../levelnInstName>/simsub.inp]
```

Click the following highlighted text to view a sample control file for the SILOS II simulator.

5. When you finish editing the file, exit the text editor. The system automatically closes the window.

Using Substitution Functions

If needed, SE maps the names you assign on your schematic to names that are valid in the syntax of your simulator. When you refer to a net name in your control file or any included file, you must use a the appropriate substitution to ensure correct translation.

The following table summarizes the substitution functions provided by SE:

Function syntax	Description
[#netname]	Replaces [#netname] with netlister-assigned name of the net corresponding to <i>netname</i>
[\$instname]	Replaces [\$instname] with netlister-assigned name of the instance corresponding to <i>instname</i>
[!filename]	Replaces [!filename] with the contents of the file named <i>filename</i> and continue to do substitutions in the included file
[?filename]	Same as [!filename] except no error message is given if the file does not exist

Simulation Environment Help

Creating the Input Stimulus

Function syntax	Description
[n!filename]	Replaces [n!filename] with the contents of the file named <i>filename</i> and does not make substitutions in the included file
[n?filename]	Same as [n!filename] except no error message is generated if the file does not exist and no substitutions are made in the included file

Simulation Environment Help

Creating the Input Stimulus

Customizing Netlisting

In this chapter, you can find information about

- [SE Netlisting](#) on page 31
- [Specifying a Hierarchy of Netlisting Views](#) on page 32
- [Selecting a Netlisting View from a Hierarchy](#) on page 32
- [Overriding Default View and Stop Lists](#) on page 32
- [Controlling Renetlisting](#) on page 33

For information on SE SKILL APIs, see the [*Digital Design Netlisting and Simulation SKILL Reference*](#).

SE Netlisting

When SE netlists your design, it takes your design hierarchy (extracted schematic or layout) and the simulator primitive component library and generates a network description containing all instances, nets, and models in an appropriate format for your simulator.

When you extract and save your schematic, the system takes the connectivity information from the drawing and saves it to the disk. The netlister uses this data, the simulation data in the Cadence `basic` and `sample` libraries, and the modeling data to create the netlist for simulation.

Some Cadence netlisters “flatten” hierarchy and produce an expanded description of the design. For HSPICE, the netlister can produce either a flattened or a hierarchical netlist. The Verilog-XL netlister produces only a hierarchical netlist.

The following sections describe SE variables you can set in your `.simrc` file to customize netlisting.

Specifying a Hierarchy of Netlisting Views

Switching Views

simViewList – This is the “view switch list.” Each of the views in this list is searched for, in order, under the cell. The first view found is “switched” with the symbol.

When a device referenced in the schematic is located, the instance of the symbol must be associated with its corresponding schematic or simulator primitive. This process is called switching views. The list of valid views to be used for switching is defined by the variable `simViewList` for the target simulator. If no valid view is found, an error message is generated because no netlist can be produced.

Selecting a Netlisting View from a Hierarchy

simStopList – This is the “stopping point view list” for the target simulator. Any view found in the simViewList is checked to see if it is in this list. If the view is found in `simStopList`, expansion stops. If it is not found, the next view found in `simViewList` is expanded.

A stopping view is the most detailed description of devices for a simulation. If a view located in the “view switch list” is specified in `simStopList`, expansion stops and the connectivity information for the instance is written to the netlist file by the SE netlist function. If a view is not specified in `simStopList`, expansion continues to locate the instance.

Overriding Default View and Stop Lists

To override the default view list or stop list, type the new list for the current simulator in the .simrc file. This sets the internal list for the current simulator to the `.simrc` value. You must set the view and stop lists separately for each simulator.

The following table summarizes the `simViewList` and `simStopList` variables that correspond to the internal variables listed for each simulator.

simSimulator	simViewList	simStopList
SPICE	<code>spiceSimViewList</code>	<code>spiceSimStopList</code>
HSPICE	<code>hspiceSimViewList</code>	<code>hspiceSimStopList</code>

The following example shows how the netlister uses the view and stop list variables to control its traversal of the design hierarchy:

```
spiceSimViewList = '("spice" "cmos.sch" "schematic")  
spiceSimStopList = '("spice")
```

The specifications tell the netlister to stop when it finds a view named “spice” and write the device into the netlist file. If a “spice” view does not exist, the netlister looks for the “cmos.sch” view to expand. If neither a “spice” nor a “cmos.sch” view exists, the netlister looks for the “schematic” view. If none of these views exist, the netlister generates an error message.

Controlling Renetlisting

Incremental netlisting variables let you reduce netlisting time by eliminating unnecessary netlisting.

simNotIncremental – Set this variable to `nil` to tell the system to netlist only the parts of your design modified since the last netlisting of the design. The default is `nil`.

simReNetlistAll – Set this variable to `t` to generate a new netlist on all the cellviews in your entire design. The default is `nil`.

simNetlistHier – Set this variable to `t` to run the hierarchical netlister.

Simulation Environment Help

Customizing Netlisting

Running a Simulation

In this chapter, you can find information about

- [Choosing Simulation Run Options](#) on page 35
- [Simulation in the Graphical Environment](#) on page 36
- [Simulation in the UNIX Environment](#) on page 38
- [Full Simulation](#) on page 38
- [Simulation in Steps](#) on page 39
- [Simulation in Batch Mode](#) on page 40
- [Interactive Simulation](#) on page 41

For information on SE SKILL APIs, see the [*Digital Design Netlisting and Simulation SKILL Reference*](#).

Choosing Simulation Run Options

You can use the Options command before you run a simulation to specify whether the

- System creates a [hierarchical](#) or [flat](#) netlist
- Simulation runs on a remote machine

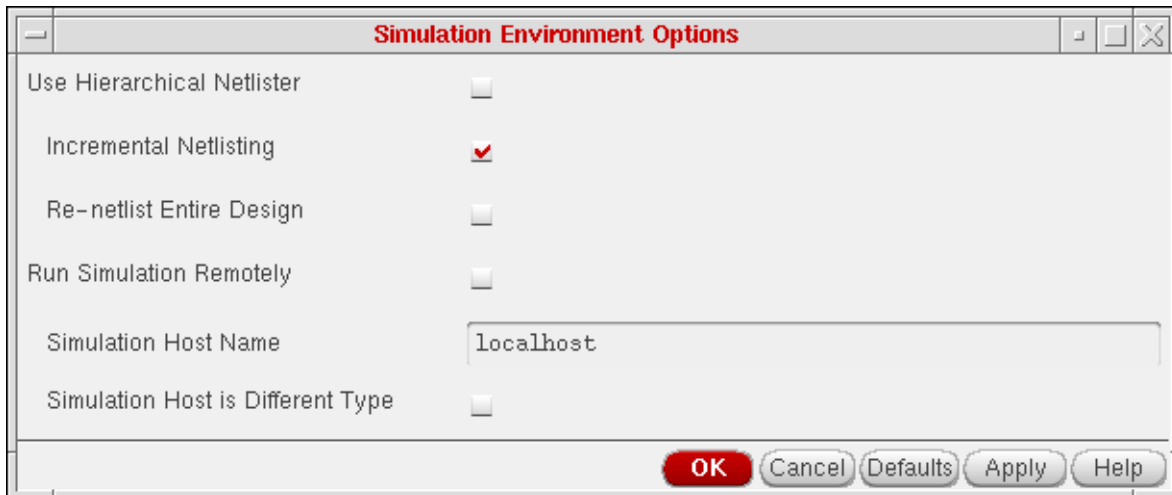
To specify the options you want for a simulation

1. In the Schematic window, select *Simulation – Options*.

Simulation Environment Help

Running a Simulation

2. Set the options you want.



3. Click *OK*.

Simulation in the Graphical Environment

To generate a netlist, run a simulation using an existing netlist, or run a complete simulation after generating a new netlist

1. In the Schematic window, select *Simulation – Netlist/Simulate*.

Simulation Environment Help

Running a Simulation

The following form appears:

The screenshot shows a dialog box titled "Netlist and Simulate". It contains the following fields and controls:

- Simulation Run Directory:** A text box containing the path "net/cicsun3/export/home/ashimag/1_books/610/spice.run1".
- Design:** A button labeled "Browse".
- Library Name:** A text box containing "ray".
- Cell Name:** A text box containing "top".
- View Name:** A text box containing "schematic".
- Simulator Name:** A dropdown menu showing "other" and an adjacent empty text box.
- Run Actions:** Two checkboxes, "netlist" and "simulate", both of which are checked.
- Run in Background:** A checked checkbox.
- Job Priority (0-20):** A slider control set to the value 10.
- Buttons:** "OK", "Cancel", "Defaults", "Apply", and "Help" are located at the bottom right of the dialog.

Most fields in this form are set to the values you specified with the Initialize command. You can edit form fields by typing or by using the Browser.

You cannot change the simulation run directory by editing this form. To change this directory, click *Cancel* and select the *Initialize* command again.

2. Type in the values and select the options you want.
3. Click OK.

If you click *Cancel*, the simulation is not run.

When a background simulation is completed, a dialog box appears with the completion status of the simulation.

Note: You cannot run more than one simulation concurrently in the same run directory. If you attempt to do so, a dialog box tells you that a simulation is already running in the directory. You can terminate the currently running job and start a new job.

Simulation in the UNIX Environment

To run a simulation in the UNIX environment

1. Change to the directory that contains the `cds.lib` file. (You can also specify the `cds.lib` file at the command line with the `-cdslib` option.)

The `cds.lib` file contains the library path to the design. If you invoke `si` in a directory that does not contain the `cds.lib` file, specify a fully-qualified path to the `cds.lib` file following the `-cdslib` option.

```
si /mnt/temp/spice.run1 -cdslib /mnt/dave/cds.lib
```

2. Type the `si` command followed by the full system path to the simulation run directory. For example, if the path to your run directory is `/mnt/temp/spice.run1`, type

```
si /mnt/temp/spice.run1
```

When the system has been initialized, SE displays a command prompt (`>`).

You can type the `si` command, with any or both of two options, `noenv` and `difftest`. You use `noenv` if you do not want the environment file to be specified by `si`. In this case, the default environment file will be picked up by OSS. For example:

```
si -noenv
```

You use the `difftest` option to strip off any date stamps that appear in the `si.log` file or in the standard output during netlisting. For example:

```
si -difftest -cdslib ./cds.lib -batch -command netlist
```

You can now type the SE commands to run any of these:

- ☐ Full simulation
- ☐ Simulation in steps
- ☐ Simulation in batch mode

Full Simulation

After you start SE, you can use the `sim` command to run a full simulation automatically.

```
> sim
```

The `sim` command initializes simulation variables appropriate for your simulator (specified by the simSimulator variable) and runs SE commands in the following order:

- simInitRunDir
- netlist

- simin
- runsim
- exit

Simulation in Steps

You can type the following commands to run a simulation in steps. Use these commands in the order listed below.

- simInitRunDir
- netlist
- simin
- runsim
- exit

simInitRunDir

The `simInitRunDir` command sets up your simulation run directory. It performs the following actions:

Copies a default control file into the simulation run directory

- Generates a default input stimulus file

Creates a waveform (*raw*) directory in the simulation run directory.

Each interface might have additional initialization procedures. Some interfaces create input files in addition to the basic `control` file.

netlist

The `netlist` command produces a text description of the design specified by the simLibName, simCellName, and simViewName variables in a file named *netlist*. The *netlist* file contains the elements, signals, models, and their interconnections in the format required by the target simulator.

simin

The `simin` command translates names in the control file from user-assigned names in your schematic to netlist-generated names acceptable to the target simulator. This function also merges all input stimulus and command files specified in the `control` file into the simulator input file `si.inp`. You can only run this command if a netlist has already been generated.

runsim

The `runsim` command runs the simulator. This command performs the following actions:

- Runs the specified simulator using the si.inp file as its input
- Translates waveform output from the simulator to the Cadence Waveform Storage Format (if necessary)
- Translates names in the text simulator output back to the user-assigned names in your design

You can only use this command after running the `netlist` and `simin` commands to generate the simulator input files.

exit

Use the `exit` command to exit SE.

Simulation in Batch Mode

You can run a simulation in batch mode by starting the si program with the `-batch` option.

```
si -batch [run_directory_name]
```

SE initializes the environment and runs the sim command. The `sim` command automatically runs the following SE commands:

simCheckVariables

simInitRunDir

netlist

simin

runsim

Simulation Environment Help

Running a Simulation

exit

For example, if your simulation run directory is `/mnt/dave/aluSimulations/silos.run1`, and you want to run a batch simulation with a log of events, enter the following command:

```
si -batch /mnt/dave/aluSimulations/silos.run1 >&  
/mnt/dave/aluSimulations/silos.run1/si.log &
```

The simulation run directory must already exist and contain the `si.env` file. Redirect the messages from SE into an `si.log` file in the simulation run directory so that you have a log of events during the run. This log is created automatically when you run simulations in the background from the Cadence graphical environment.

You can also run a single SE command in batch mode with the `-command` option.

```
si -batch -command command_name run_directory_name
```

For example, if your run directory name is `/mnt2/deborah/spice.run1`, and you want to generate a netlist, enter the following command:

```
si -batch -command netlist /mnt2/deborah/spice.run1
```

Note: For using the Spectre simulator, type `nl` as the parameter for `-command`.

Interactive Simulation

If your simulation interface supports interactive simulation, you can run simulation interactively using the following procedure. You must run the Initialize command before you use the Interactive command.

1. In the Schematic window, select Simulation – Interactive.
2. Choose to netlist your design, or not, and click *OK*.

By default, the system opens the following three windows:

- ☐ A window for displaying waveforms produced during the simulation (at the top half of the screen)
- ☐ A window for editing the design you are simulating (at the bottom right)
- ☐ A window for textual interaction with the simulator (at the bottom left)

Commands in the simulator window vary depending on the simulator you use.

If you prefer to set the window placement yourself, in the CIW, select Options – User Preferences and set the Place Manually button.

Simulation Environment Help

Running a Simulation

3. When you finish simulation, select Finish Interactive from the SE menu of the simulator window.

Netlist and Simulate Form

Simulation Run Directory specifies the simulation run directory. The field is read-only.

Design fields let you find or specify the design.

Browse brings up the TDM Library Browser when you click the button.

Library Name specifies the name of the library containing the top level of your design.

Cell Name specifies the cell name of the top level of your design.

View Name specifies the view name of your design.

Simulator Name specifies the name of the simulator. Select *other* if you want to type in the name of an unlisted simulator.

Run Actions buttons specify whether you want to netlist, simulate, or both.

netlist creates a netlist.

simulation runs a simulation.

Run in Background runs the job in the background and lets you use the graphical user interface while the job is running.

Job Priority adjusts the priority for background jobs only. Change the priority by clicking and dragging the bar in the field to the left or right. The lower the number in this field, the faster the background job runs and the slower the system performs other tasks.

Simulation Environment Options Form

Use Hierarchical Netlist creates a hierarchical, rather than a flat, netlist. This field is ignored if both types of netlists are not supported.

Incremental Netlisting limits the new hierarchical netlisting to parts of the design you modified since you last created a netlist. The field has no effect unless you use the hierarchical netlister.

Re-netlist Entire Design creates a new netlist for all cellviews in your design.

Run Simulation Remotely runs the simulation on a remote file server. You specify the remote file server in the Simulation Host Name field.

Simulation Host Name specifies the name of the remote simulator.

Simulation Host Is Different Type specifies that a computer running a remote simulation has a different binary storage format than your local computer. You must set this correctly for remote simulations to view waveform results.

Displaying Results

In this chapter, you can find information about

- [Displaying Waveform Results in the Graphical Environment](#) on page 45
- [Displaying Waveform Results in Register Form](#) on page 46
- [Displaying Text Results](#) on page 47
- [Displaying Netlisting Errors for Specific Nets or Instances](#) on page 47
- [Displaying Other Netlisting Errors \(SILOS II Only\)](#) on page 47
- [Displaying a Specified Text File](#) on page 48

For information on SE SKILL APIs, see the [Digital Design Netlisting and Simulation SKILL Reference](#).

Displaying Waveform Results in the Graphical Environment

To display the waveforms produced during a simulation

1. In the Schematic window, select – Simulation – Show Waveforms.

The following form appears:



Simulation Environment Help

Displaying Results

The simulation run directory is the one you specified with the Initialize command. You cannot change this directory by editing this form. To change the simulation run directory, click *Cancel* and select the Initialize command again.

2. In the Waveform File Name field, type the name of the waveform file to be displayed.

The system looks for this file in a directory called *raw* in the simulation run directory. The default waveform file is *waves*. Most of the standard interfaces do not require that you change this value. If you are using a tool that produces multiple waveform files, such as the Cadence HSPICE interface, you need to change this field to see the waveforms in each waveform file.

3. Click OK.

The system opens the waveform window and displays information for the simulation run and the waveform file you specified. You can now execute any of the waveform commands to manipulate and display specific waveforms.

Displaying Waveform Results in Register Form

To display in register form the waveforms produced during a simulation

1. In the Schematic window, select – Simulation – Show Registers.

The Show Registers form appears:



Simulation Run Directory is a read-only field showing the path to the current simulation run directory. You cannot change this directory by editing this form. To change the simulation run directory, click *Cancel* and select the Initialize command again.

The Waveform File Name field specifies the name of the waveform file to be displayed. The system expects this file to be in the *raw* directory in the simulation run directory. The default waveform file is *waves*. Most of the standard interfaces do not require that you change this field. If you are using a tool that produces multiple waveform files, such as the Cadence HSPICE interface, you need to change this field to see the waveforms in each waveform file.

2. Fill out the form and click OK.

The system opens the register display window.

Displaying Text Results

To see the text output (`si.out`) from the simulator

1. In the Schematic window, select – Simulation – Show Outputs – Show Output.

A window appears showing the `si.out` file from the current simulation run directory. This file contains the text output of the simulator, including any simulator error messages.

2. To close the window, select File – Close Window.

Displaying Netlisting Errors for Specific Nets or Instances

To highlight errors associated with a specific net or instance

- In the Schematic window, select Simulation – Show Outputs – Highlight Errors. The system places a probe on any object in your design that has an error. For example, the system might place a probe on a gate with an undriven input if your simulator does not support these. The system might place probes lower in the design hierarchy and then reflect them up the hierarchy to the instance containing them. Once probes are placed, you can use the standard probing functions to manipulate and remove them.

Note: If an error cannot be isolated to a particular net or instance, the system does not place a probe. Use the Show Global Error command to see this type of error.

Displaying Other Netlisting Errors (SILOS II Only)

To display errors that are not associated with a particular net or instance

1. In the Schematic window, select – Simulation – Show Outputs – Show Global Error.

A window appears showing the error messages generated by the flat netlister or by your analysis tool. If no error file exists, a dialog box informs you that there were no global errors.

2. To close the window, select File – Close Window.

Note: The Show Global Error command is only available with *Cadence Silos II*.

Displaying a Specified Text File

To display any text file in the current simulation run directory

1. In the Schematic window, select *Simulation – Show Outputs – Show Run File*.

The following form appears:



You cannot change the simulation run directory by editing this form. To change this directory, click *Cancel* and select the Initialize command again.

2. Type the name of the file you want to see in the *File Name* field and click *OK*.

A window appears showing the file you specified. The system expects this file to be in the simulation run directory, so you do not need to type the full path to the file.

3. To close the window, select *File – Close Window*.

Controlling Job Status

This chapter covers the following topics:

- [Accessing the Job Monitor Form](#) on page 49
- [Checking Current Simulation Status](#) on page 50
- [Adjusting Job Priority](#) on page 50
- [Terminating a Simulation](#) on page 51
- [Interrupting or Restarting a Simulation](#) on page 52
- [Editing the Job Monitor Form](#) on page 52

For information on SE SKILL APIs, see the [Digital Design Netlisting and Simulation SKILL Reference](#).

Accessing the Job Monitor Form

You can use the Job Monitor command to perform the following operations for a background analysis job:

- Check the job status
- Monitor output
- Suspend the job
- Change the execution priority
- Stop the job

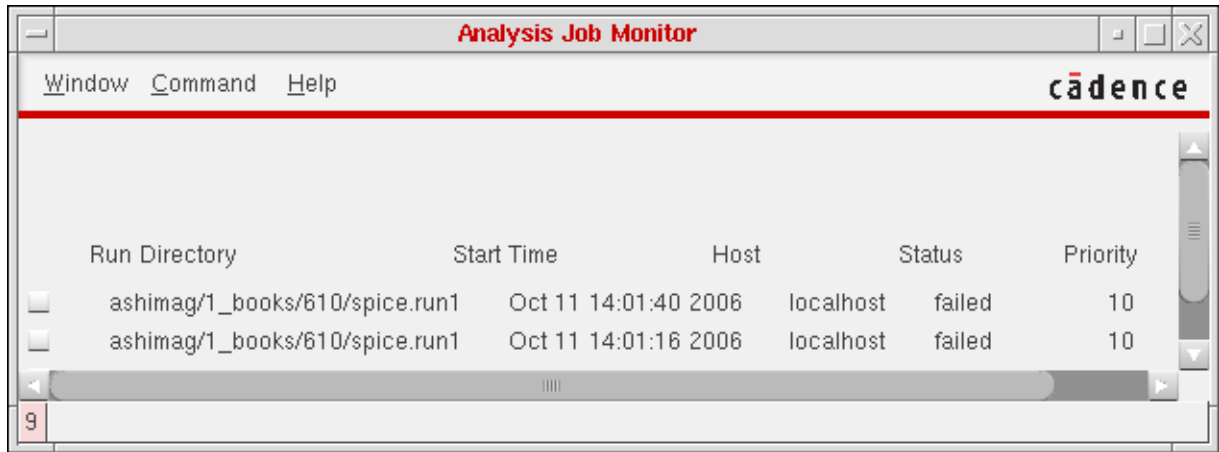
To perform any of these functions

1. In the Schematic window, select *Simulation – Job Monitor*.

Simulation Environment Help

Controlling Job Status

The following form window appears:



SE background analysis jobs appear in this form window whether or not you started them using the Simulation menu.

2. Click the jobs for which you want to run a Job Monitor command.
3. Select the appropriate command from the Command menu.
4. To close the Job Monitor form window, select *Window – Close*.

Checking Current Simulation Status

To bring up a window showing the run log (`si.log`) for the selected jobs

- In the Job Monitor form window, select *Command – Show Run Log*.

To close the run log window

- In the Run Log window, select *File – Close Window*.

Note: You can also display the run log of a background job with the *Simulation – Show Outputs – Show Run Log* commands. The *Show Foreground Run Log* option shows the log for a foreground job, and the *Show Background Run Log* option shows the log for a background job.

Adjusting Job Priority

To change the priority of the selected jobs

Simulation Environment Help

Controlling Job Status

1. In the Job Monitor form window, select *Command – Set Priority*.

The following form appears:



2. Click and drag the bar in the Job Priority field.

Try to set an appropriate job priority before you start the job. Once a job has started, you can only decrease the priority of a job unless you have root access. (This is a UNIX requirement.) For example, to decrease the priority of a job set at 10, change it to 12. This slows down the background job and increases the speed of your local Cadence graphics shell.

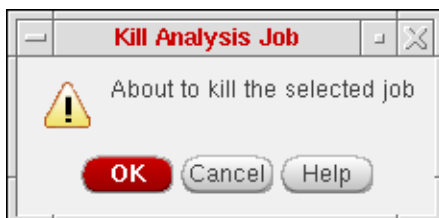
3. Click OK.

Terminating a Simulation

To terminate the selected jobs

- In the Job Monitor form window, select *Command – Kill*.

A dialog box asks you to confirm the job terminations. If you kill a job, you cannot resume it with the Continue command.



Interrupting or Restarting a Simulation

To suspend the selected jobs

- In the Job Monitor form window, select *Command – Suspend*. The selected job is listed as suspended.

To resume the jobs you suspended with the Suspend command

- In the Job Monitor form window, select *Command – Continue*. The selected job is listed as running.

Editing the Job Monitor Form

To remove the selected jobs from the Job Monitor form

- In the Job Monitor form window, select *Command – Remove Entry*. A dialog box asks you to confirm the deletion.



Jobs you select are not terminated, but you can no longer change their priorities or terminate them. You can still see output from these simulations (with the Show Run Log command) if you first initialize the environment and specify their simulation run directories.

SE Functions Reference

in this chapter you can find information about

- [Initialize](#) on page 53
- [Options](#) on page 55
- [Stimulus – Edit File](#) on page 57
- [Netlist/Simulate](#) on page 58
- [Interactive](#) on page 60
- [Show Outputs – Show Run Log](#) on page 61
- [Show Outputs – Show Output](#) on page 62
- [Show Outputs – Show Global Error](#) on page 62
- [Show Outputs – Highlight Errors](#) on page 63
- [Show Outputs – Show Run File](#) on page 64
- [Show Waveforms](#) on page 65
- [Show Registers](#) on page 66
- [Job Monitor](#) on page 67

For information on SE SKILL APIs, see the [*Digital Design Netlisting and Simulation SKILL Reference*](#).

Initialize

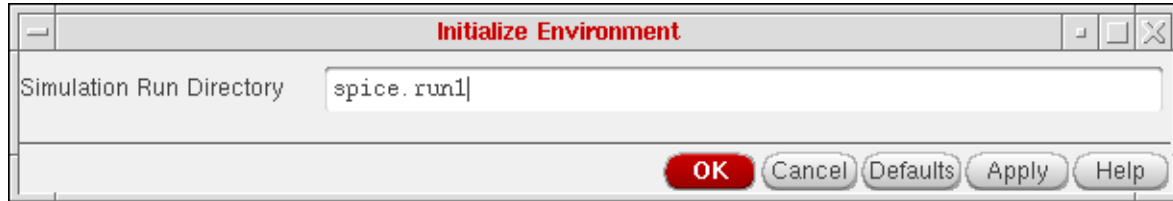
Simulation



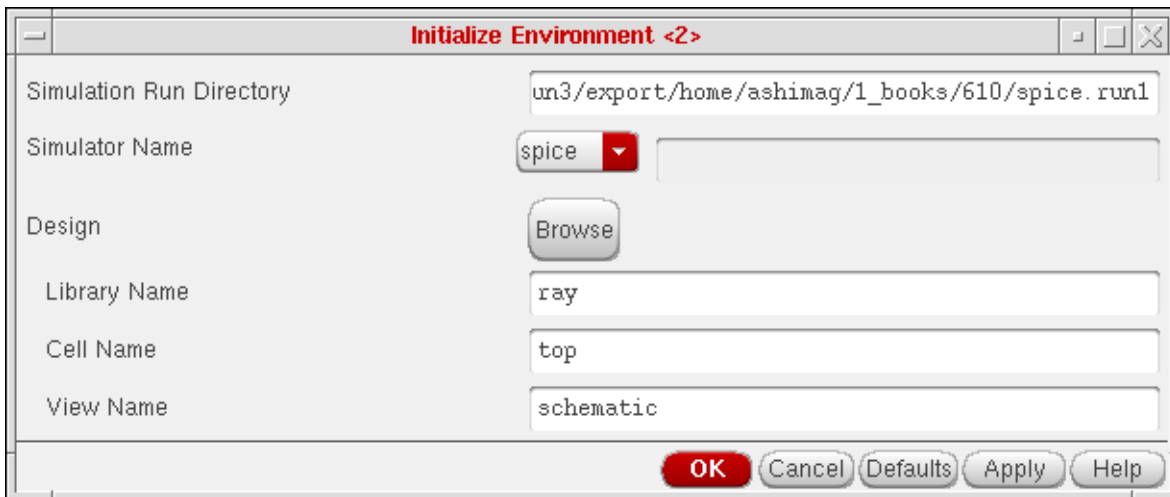
Initialize...

Initializes the environment for simulation.

Initialize Forms



Simulation Run Directory specifies the directory in which the system stores simulation input and output files.



Simulator Name specifies your analysis tool. This field shows the analysis tools available. If you want to use a tool that is not listed, select *other* and type the name of that tool in the adjoining text entry field. You can only type in the text entry field if you have selected *other*.

Library Name specifies the library containing the top level of your design.

Cell Name specifies the top-level cell to simulate, for example, *inverter*.

View Name specifies the name of the view to simulate, for example, *schematic*.

Using Initialize for a New Run Directory

1. Select *Initialize*.
2. Type the name of the *Simulation Run Directory*.

You can type either a full or a relative path. If you type a relative path, the system puts the run directory under the directory in which you started the software. The default is *spice.run1*.

3. Click *OK*.

The *InitializeEnvironment* form appears.

If this is the first time you have initialized SE in this session, the values on this form are taken from the current window and from default SE values. You can edit Design fields by typing or by using the Browser.

4. Fill out this form and click *OK*.

Using Initialize for an Existing Run Directory

1. Select *Initialize*.
2. Type the name of the *Simulation Run Directory* and click *OK*.

Top-Level SKILL Command

```
simInitEnv( )
```

Options

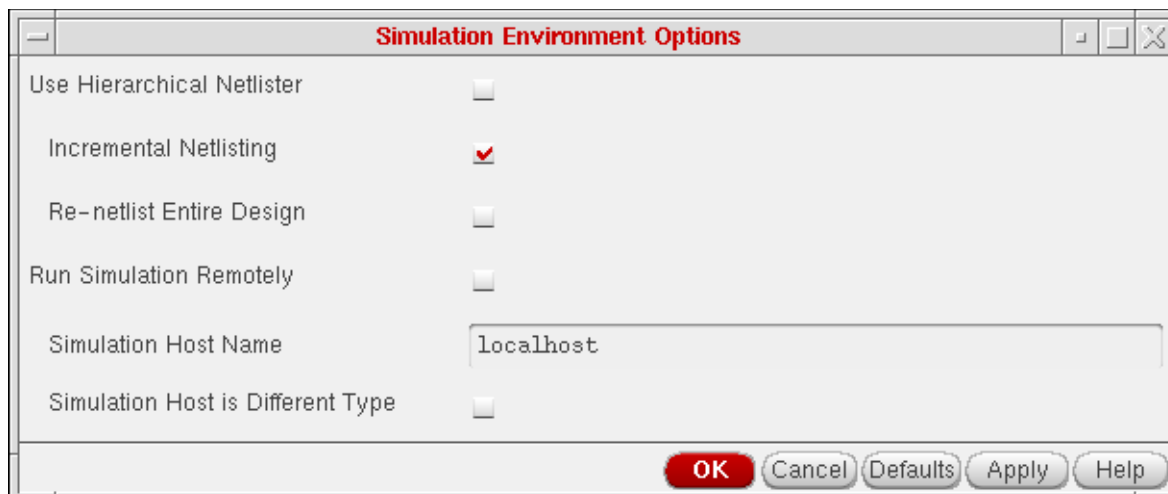
Simulation ⇒ **Options...**

Sets simulation run options that do not appear on the Netlist/Simulate form. These options determine whether the system creates a flat or hierarchical netlist, runs remote simulation, or runs the *simdiff* command during the simulation process. The fields on the *Options* form have the default values or the values you specified when you last ran a simulation in the current simulation run directory.

Prerequisites

You must run *Initialize* before you use *Options*.

Options Form



The screenshot shows a dialog box titled "Simulation Environment Options". It contains several settings:

- Use Hierarchical Netlister**: ☐
- Incremental Netlisting**: ☒
- Re-netlist Entire Design**: ☐
- Run Simulation Remotely**: ☐
- Simulation Host Name**: A text field containing "localhost".
- Simulation Host is Different Type**: ☐

At the bottom right, there are five buttons: **OK** (highlighted in red), **Cancel**, **Defaults**, **Apply**, and **Help**.

Use Hierarchical Netlister specifies whether the system creates a hierarchical netlist. For tools and interfaces that do not support both types of netlists, this field is ignored.

Incremental Netlisting specifies whether the system netlists only the parts of your design you modified since you last netlisted.

Re-netlist Entire Design specifies whether the system runs a new netlist for all cellviews in your design.

Run Simulation Remotely specifies whether the simulation runs on a remote file server or on your local computer.

- ☒ Runs simulation on a different computer. *Simulation Host Name* and *Simulation Host is Different Type* specify the computer that will be used for remote simulation.
- ☐ Runs simulation on your computer. *Simulation Host Name* and *Simulation Host is Different Type* have no effect.

Simulation Host Name specifies the computer on which the simulation runs. The default is *localhost*, which is your machine. Type the name of the computer on which you want to run remote simulation. This field requires that *Run Simulation Remotely* is turned on.

Simulation Host is Different Type specifies whether the computer on which you run the simulation has a different binary storage format.

- ☐ Indicates the storage format is the same.
- ☒ Indicates the storage format is different. If you do not set this field correctly, you will be unable to look at your waveform results after the simulation finishes. This field requires that *Run Simulation Remotely* is turned on.

Using Options

1. Select *Options*.
2. Fill out the form that appears and click *OK*.

Stimulus – Edit File

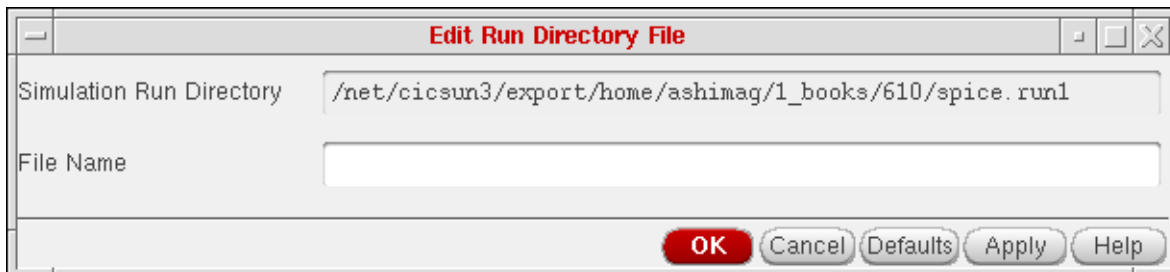


Lets you edit any file in the current simulation run directory.

Prerequisites

You must run *Initialize* before you use *Edit File*.

Edit File Form



Simulation Run Directory specifies the name of the current simulation run directory. You cannot change the directory by editing this form. To change the directory, click *Cancel* and select the *Initialize* command again.

File Name specifies the name of the file you want to edit in the current simulation run directory. You can edit or create any file in the simulation run directory with this command.

Using Edit File

1. Select *Edit File*.
2. Fill out the form that appears and click *OK*.

The system displays the file in a window. (You can change the text editor by modifying the UNIX *EDITOR* shell variable or the SKILL *editor* variable.) After editing the file, exit the editor. The system closes the window automatically.

Netlist/Simulate



Generates a netlist, runs a simulation using an existing netlist, or runs a complete simulation, including generating a new netlist.

Prerequisites

You must run *Initialize* before you use *Netlist/Simulate*.

Netlist/Simulate Form



The screenshot shows a dialog box titled "Netlist and Simulate". It contains several fields and controls:

- Simulation Run Directory:** A text field containing the path "net/cicsun3/export/home/ashimag/1_books/610/spice.run1".
- Design:** A button labeled "Browse".
- Library Name:** A text field containing "ray".
- Cell Name:** A text field containing "top".
- View Name:** A text field containing "schematic".
- Simulator Name:** A dropdown menu showing "other" with a red arrow, and an adjacent empty text field.
- Run Actions:** Two checkboxes, "netlist" and "simulate", both of which are checked.
- Run in Background:** A checked checkbox.
- Job Priority (0-20):** A slider control with the value "10" displayed above it.
- Buttons:** At the bottom right, there are five buttons: "OK" (highlighted in red), "Cancel", "Defaults", "Apply", and "Help".

Simulation Run Directory displays the path to the simulation run directory. The simulation run directory contains simulation input and output files. This is the same directory you specified when you ran the *Initialize* command. This field is read-only. You cannot modify it on this form. To change the simulation run directory, you must run the *Initialize* command.

Library Name specifies the library that contains the top-level cellview you want to simulate.

Cell Name specifies the cell name of the top-level cellview to simulate.

View Name specifies the view name of the top-level cellview to simulate.

Simulator Name specifies your analysis tool. This field shows the analysis tools available. If you want to use a tool that is not listed, select *other* and type the name of that tool in the adjoining text entry field. You can only type in the text entry field if you have selected *other*.

Run Actions (Netlist/Simulate) specifies whether to run just the netlister, just a simulation, or both.

Run in Background determines whether to run the simulation in the background.

-
- ☒ Runs the simulation in the background. While the simulation is running, you can continue to use the graphics shell to monitor the simulation or perform other editing or analysis tasks. If you do not run the simulation in the background, you cannot use the graphical environment until the simulation is done.
 - ☐ Runs the simulation in the foreground, which is most useful for small designs.
-

Job Priority specifies the UNIX priority of the background simulation. The lower the number in this field, the faster the background simulation and the slower your system performance on other tasks. You cannot modify the priority of a foreground simulation. Unless you have root access privilege, you cannot lower the priority of a job once it is started.

Using Netlist/Simulate

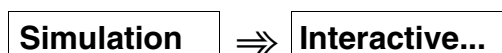
1. Select *Netlist/Simulate*.
2. Fill out the form that appears and click *OK*.
 - ☐ If you click *Cancel*, the simulation is not run.
 - ☐ If you try to run a simulation using a simulation run directory in which a simulation is running, the system informs you a simulation is running and asks you whether you want to disregard the existing job and start a new job.

When the simulation is done, a dialog box appears giving the completion status of the simulation.

Top-Level SKILL Command

```
simRunNetAndSim( )
```

Interactive



Starts an interactive simulation session.

Prerequisites

You must run `Initialize` before you use `Interactive`.

Using Interactive

1. Select `Interactive`.
2. Select or de-select *Netlist your design first?* and click *OK*.

The system opens three windows: one for displaying waveforms produced during the simulation, one for editing the design you are simulating, and one for textual interaction with the simulator.

Because simulators are different, the menus and forms in the simulator window vary by simulator. Refer to the manual on your simulation interface for information on simulator-specific commands.

Top-Level SKILL Command

```
iseStartInteractive( )
```

Show Outputs – Show Run Log



Displays the run log (`si.log`) of a background simulation or the run log (`si.foregnd.log`) of a foreground simulation. Both run logs contain the output from the simulation environment. It lists the simulation steps and their completion status. It also lists any error messages from SE, including the netlisters.

Prerequisites

You must run *Initialize* before you use *Show Run Log*.

Using Show Run Log

1. Select
Show Run Log – Show Foreground Run Log or
Show Run Log – Show Background Run Log.

A window appears showing the `si.log` file of the foreground or background job.

2. Select the *Close Window* command from the File menu to close the window.

Show Outputs – Show Output



Displays the `si.out` file from the simulation run directory. This file contains the text output of the simulator, for example, error messages.

Prerequisites

You must run *Initialize* before you use *Show Output*.

Using Show Output

1. Select *Show Output*.
A window appears showing the `si.out` file.
2. Select the *Close Window* command from the File menu to close the window.

Show Outputs – Show Global Error



Displays global errors (`global.err`) found during netlisting or simulation with interfaces that support this feature. Global errors are not associated with a particular net or instance.

Prerequisites

You must run *Initialize* before you use *Show Global Error*.

Using Show Global Errors

1. Select *View Global Error*.

A window appears showing the error messages generated by the flat netlister or by your analysis tool. If no error file exists, a dialog box informs you that there were no global errors.

You can see global errors generated by the simulator only if you use Cadence SILOS II.

2. Select the *Close Window* command from the File menu to close the window.

Show Outputs – Highlight Errors



Places a probe on any object in your schematic that has an error. For example, the system might place a probe on a gate with an undriven input if your simulator does not support these. The system might place probes lower in the design hierarchy and reflect them up the hierarchy to the instance that contains them.

After the probes have been placed, you can use the standard probing functions to manipulate and remove them. If an error cannot be isolated to a particular net or instance, the system does not place a probe on it. Use the *Show Global Error* command to see this type of error.

Prerequisites

You must run *Initialize* before you use *Highlight Errors*.

Using Highlight Errors

Select *Highlight Errors*.

The system places probes on objects in your design with errors.

Show Outputs – Show Run File

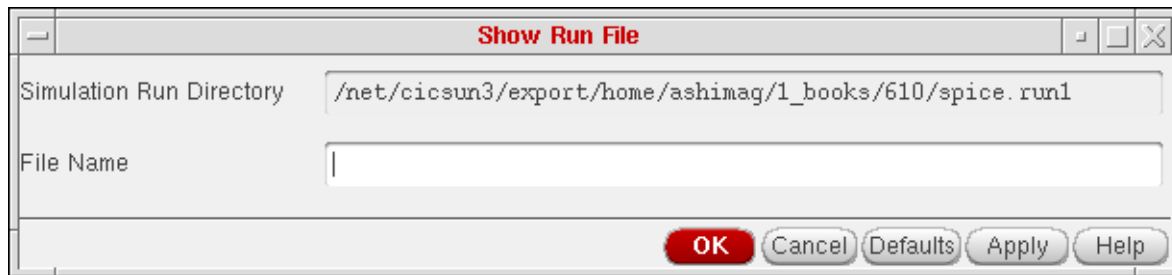


Lets you look at any text file in the current simulation run directory.

Prerequisites

You must run `Initialize` before you use *Show Run File*.

Show Run File Form



Simulation Run Directory is a read-only field displaying the path to the current simulation run directory. To change this directory, click *Cancel* and select the *Initialize* command again.

File Name specifies the name of the file to display. The system expects the file to be in the simulation run directory.

Using Show Run File

1. Select *Show Run File*.
2. Fill out the form and click *OK*.
3. Select *Close Window* from the File menu to close the window.

Show Waveforms

Simulation ⇒ Show Waveforms

Lets you display the waveforms produced during a simulation.

Prerequisites

You must run *Initialize* before you use *Show Waveforms*.

Show Waveforms Form



Simulation Run Directory is a read-only field showing the path to the current simulation run directory. You cannot change the directory by editing this form. To change the directory, click *Cancel* and select the *Initialize* command again.

Waveform File Name specifies the name of the waveform file to display. The system expects this file to be in the *raw* directory in the simulation run directory. The default waveform file is *waves*. Most of the standard interfaces do not require that you change this field. If you are using a tool that produces multiple waveform files, such as the Cadence HSPICE interface, you need to change this field to look at the waveforms for each analysis run.

Using Show Waveforms

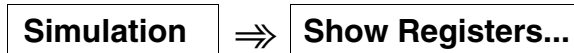
1. Select *Show Waveforms*.
2. Fill out the form that appears and click *OK*.

The system opens the waveform window and displays information on the simulation run and the waveform file you specified. You can now execute any of the waveform commands to manipulate and display specific waveforms.

Top-Level SKILL Command

`simWaveOpen ()`

Show Registers



Lets you display the waveforms produced during a simulation in register form.

Note: Show Registers works only with icfb and icds work benches.

Prerequisites

You must run *Initialize* before you use *Show Registers*.

Show Registers Form

A screenshot of a dialog box titled "Show Registers". It has two text input fields. The first field is labeled "Simulation Run Directory" and contains the path "/net/cicsun3/export/home/ashimag/1_books/610/spice.run1". The second field is labeled "Waveform File Name" and contains the text "waves". At the bottom right of the dialog box are five buttons: "OK" (highlighted in red), "Cancel", "Defaults", "Apply", and "Help".

Simulation Run Directory is a read-only field showing the path to the current simulation run directory. You cannot change the directory by editing this form. To change the directory, click *Cancel* and select the *Initialize* command again.

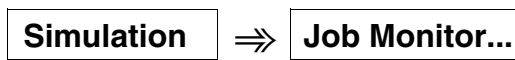
Waveform File Name specifies the name of the waveform file to display. The system expects this file to be in the *raw* directory in the simulation run directory. The default waveform file is *waves*. Most of the standard interfaces do not require that you change this field. If you are using a tool that produces multiple waveform files, such as the Cadence HSPICE interface, you need to change this field to look at the waveforms for each analysis run.

Using Show Registers

1. Select *Show Registers*.
2. Fill out the form that appears and click *OK*.

The system opens the register display window. Refer to the *Waveform Display User Guide* for information on the commands available in this window.

Job Monitor

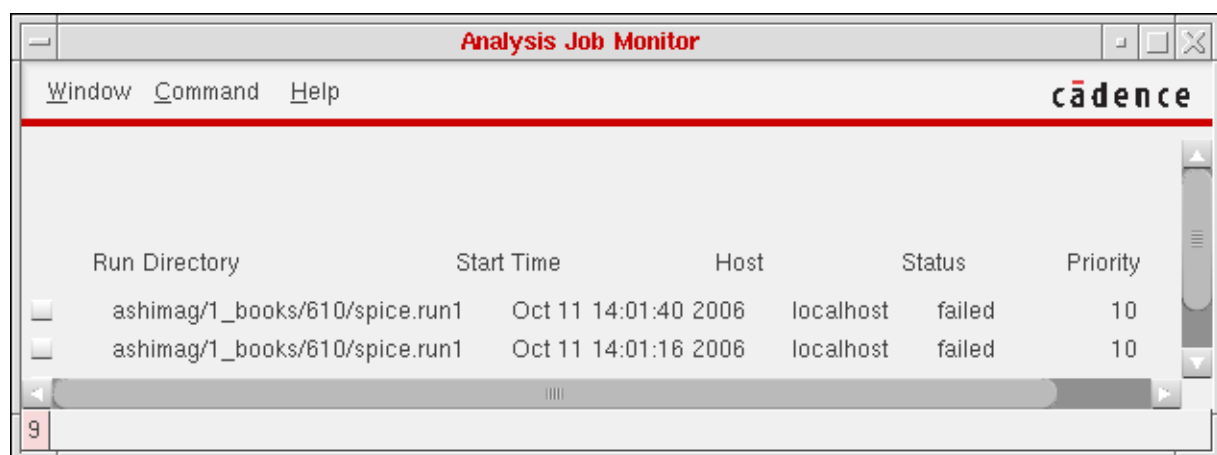


Lets you check the status of a background analysis job, monitor its output, suspend it, change its execution priority, or stop it. SE background analysis jobs appear in the *Job Monitor* form whether or not you started them using the Simulation menu.

Prerequisites

You must run *Initialize* and a background simulation before you run *Job Monitor*.

Job Monitor Form



Show Run Log brings up a window showing the run log (*si.log*) for the selected jobs.

Set Priority brings up a form in which you can change the priority of the selected jobs. Unless you have root access privilege, you cannot lower the priority of a job once it is started.

Kill terminates the selected jobs.

Suspend suspends the selected jobs.

Continue resumes the jobs you suspended with *Suspend*.

Remove Entry deletes the selected jobs from the form.

Using Job Monitor

1. Select *Job Monitor*.
2. Click the jobs on which you want to run a *Job Monitor* command.
3. Select *Command – Show Run Log* to bring up a window showing the run log (*si.log*) for the selected jobs.
4. Select *Command – Set Priority* to bring up a subform in which you can change the priority of the selected jobs.

You can only decrease the priority of a job. For example, to decrease the priority of a job set at 10, change it to 12. This slows down the background job and increases the speed of the Cadence graphics shell.

5. Select *Command – Kill* to terminate the selected jobs.

A dialog box appears asking you to confirm that you want to kill the jobs. If you kill a job, you cannot continue it with the *Continue* command.

6. Select *Command – Suspend* to suspend the selected jobs.
7. Select *Command – Continue* to resume the jobs you suspended with the *Suspend* command.
8. Select *Command – Remove Entry* to delete the selected jobs from the form.

The simulations they represent will not be terminated. Because these jobs will not appear on the form, you cannot change their priority or terminate them. You can still look at the output from these simulations (with the *Shown Run Log* command) by initializing the environment and specifying their simulation run directory.

9. Select *Window – Close* on the *Job Monitor* form to close the form.

Top-Level SKILL Command

`simJobMonitor()`

Simulation Environment Help

SE Functions Reference

Sample Files

In this chapter you will find information about

- [Sample control File](#) on page 71
- [Sample si.inp File](#) on page 72
- [Sample si.inp File Generated for Cadence SILOS II](#) on page 73

For information on SE SKILL APIs, see the [Digital Design Netlisting and Simulation SKILL Reference](#).

Sample control File

Following this table is a sample `control` file for Cadence SILOS II. This table explains the function of each individual line.

Line numbers	Line function
1	Tells Cadence SILOS II to read the netlist file using the Cadence SILOS II file inclusion command
2	Tells Cadence SILOS II to read network data from the terminal input that is the <code>si.inp</code> file created from this <code>control</code> file
3	Comment line
4, 5, 6	Name translations
7, 8, 9, 10, 11	Stimulus patterns used to drive the simulator
13, 14, 15, 16, 17, 18	Cadence SILOS II commands (The exclamation point (!) is an escape character required for commands that follow the <code>input.term</code> line.)

Simulation Environment Help

Sample Files

Line	#
1	input netlist
2	input .term
3	\$This is test #1 of the 74LS169A counter.
4	.TABfsLE [#QD] [#QC] [#QB] [#QA] [#RCO]
5	[#CLOCK] .CLK 0 S0 20 S1 30 S0 .REP 0
6	.PATTERN [#UP] [#LOAD*] [#P*] [#T*] [#D] [#C] [#B] [#A]
7	0 0 0 0 0 0000
8	60 0 0 0 0 0000
9	120 0 0 0 0 0001
10	180 0 0 0 0 0010
11	240 0 0 0 0 0011
12	.EOP
13	!type errors
14	!simul 0 to 6960
15	!save
16	!type errors
17	!type outputs on change
18	!type network
19	.end
20	!exit

Sample si.inp File

When you run simulation, the system automatically translates the `control` file and creates the `si.inp` file. The system uses the `si.inp` file as input for the simulator. This file has the same text as the `control` file, with commands in square brackets (described in [“Using Substitution Functions”](#)) replaced by their interpreted value.

Simulation Environment Help

Sample Files

The `si.inp` file shown below was created from the previous `control` file. In the `si.inp` file, the net names have become “N” followed by the net number on lines 4, 5, and 6 of the `control` file. These names were generated by the flat netlister.

Sample `si.inp` File Generated for Cadence SILOS II

```
input netlist
input .term
$This is test #1 of the 74LS169A counter.
.TABLE N4 N3 N2 N1 N23
N14 .CLK 0 S0 20 S1 30 S0 .REP 0
.PATTERN N15 N22 N16 N17 N21 N20 N19 N18
0 0 0 0 0 0000
60 0 0 0 0 0 0000
120 0 0 0 0 0 0001
180 0 0 0 0 0 0010
240 0 0 0 0 0 0011
.EOP
!type errors
!simul 0 to 6960
!save
!type errors
!type outputs on change
!type network
.end
!exit
```