virtutech

Simics Installation Guide for Linux/Solaris

Personal Academic License (PAL)

Simics Version	3.0
Revision	1406
Date	2008-02-19

Contents

Chapter 1

About Simics Documentation

1.1 Conventions

Let us take a quick look at the conventions used throughout the Simics documentation. Scripts, screen dumps and code fragments are presented in a `monospace` font. In screen dumps, user input is always presented in bold font, as in:

```
Welcome to the Simics prompt
simics> this is something that you should type
```

Sometimes, artificial line breaks may be introduced to prevent the text from being too wide. When such a break occurs, it is indicated by a small arrow pointing down, showing that the interrupted text continues on the next line:

```
This is an artificial ↵
line break that shouldn't be there.
```

The directory where Simics is installed is referred to as `[simics]`, for example when mentioning the `[simics]/README` file. In the same way, the shortcut `[workspace]` is used to point at the user's workspace directory.

1.2 Simics Guides and Manuals

Simics comes with several guides and manuals, which will be briefly described here. All documentation can be found in `[simics]/doc` as Windows Help files (on Windows), HTML files (on Unix) and PDF files (on both platforms). The new Eclipse-based interface also includes Simics documentation in its own help system.

Simics Installation Guide for Unix and for Windows

These guides describe how to install Simics and provide a short description of an installed Simics package. They also cover the additional steps needed for certain features of Simics to work (connection to real network, building new Simics modules, . . .).

Simics User Guide for Unix and for Windows

These guides focus on getting a new user up to speed with Simics, providing information on Simics features such as debugging, profiling, networks, machine configuration and scripting.

Simics Eclipse User Guide

This is an alternative User Guide describing Simics and its new Eclipse-based graphical user interface.

Simics Target Guides

These guides provide more specific information on the different architectures simulated by Simics and the example machines that are provided. They explain how the machine configurations are built and how they can be changed, as well as how to install new operating systems. They also list potential limitations of the models.

Simics Programming Guide

This guide explains how to extend Simics by creating new devices and new commands. It gives a broad overview of how to work with modules and how to develop new classes and objects that fit in the Simics environment. It is only available when the DML add-on package has been installed.

DML Tutorial

This tutorial will give you a gentle and practical introduction to the Device Modeling Language (DML), guiding you through the creation of a simple device. It is only available when the DML add-on package has been installed.

DML Reference Manual

This manual provides a complete reference of DML used for developing new devices with Simics. It is only available when the DML add-on package has been installed.

Simics Reference Manual

This manual provides complete information on all commands, modules, classes and haps implemented by Simics as well as the functions and data types defined in the Simics API.

Simics Micro-Architectural Interface

This guide describes the cycle-accurate extensions of Simics (Micro-Architecture Interface or MAI) and provides information on how to write your own processor timing models. It is only available when the DML add-on package has been installed.

RELEASENOTES and LIMITATIONS files

These files are located in Simics's main directory (i.e., `[simics]`). They list limitations, changes and improvements on a per-version basis. They are the best source of information on new functionalities and specific bug fixes.

Simics Technical FAQ

This document is available on the Virtutech website at `http://www.simics.net/support`. It answers many questions that come up regularly on the support forums.

Simics Support Forum

The Simics Support Forum is the main support tool for Simics. You can access it at `http://www.simics.net`.

Other Interesting Documents

Simics uses Python as its main script language. A Python tutorial is available at `http://www.python.org/doc/2.4/tut/tut.html`. The complete Python documentation is located at `http://www.python.org/doc/2.4/`.

Chapter 2

System Requirements

- **Linux/x86**
  Simics is built for Red Hat Linux 7.3. Simics also runs on later Red Hat versions and many other Linux distributions. Minimum processor requirement is a Pentium II or equivalent.

  Linux/AMD64
  Built for SuSE 9.0. Simics also runs on many other Linux distributions.

  Solaris/UltraSPARC 64-bit
  Built for Solaris 8. Simics also runs on Solaris 9 and 10.

- Requirements on memory and disk sizes depend on the workload, but at least 512MB RAM and several GB of free disk space is recommended.

- The Eclipse based user-interface in Simics requires version 2.2.1 of the GTK+ widget toolkit and associated libraries (GLib, Pango). To read the online documentation in Eclipse, Mozilla 1.4GTK2 or later is also required.

Chapter 3

Installing Simics

Simics is provided as a `tar` archive file called `simics-pal-3.0.0-linux.tar` where 3.0.0 is the Simics version number. It is recommended that Simics is installed by `root` in a read-only location, such as `/opt/virtutech/`. Once Simics is installed, a Simics workspace can be setup, as described in section 4.

Although Simics for PAL (Personal Academic License) users is a single user product, it is recommended that the workspace with user files is kept separate from the base Simics installation.

Installing a Simics Package

1. Download the Simics package from the download site.
2. Unpack the Simics package by running `tar`.

   ```
   $ tar xf simics-pal-3.0.0-linux.tar
   ```

   This will create a temporary installation directory called `simics-3.0-install`.
3. Change to the temporary installation directory and run the install script.

   ```
   $ su
   Password:
   # cd simics-3.0-install
   # sh install-simics.sh
   ```

4. The install script will search the current directory for packages to install and ask for an alternative directory to install Simics in instead of the default one.
5. When the install script finishes, Simics has been installed in `/opt/virtutech/simics-<version>/`. This path is referred to as `[simics]` in all Simics documentation. The `/opt/virtutech/` path may differ if an alternative installation directory was specified when the install script was run.

6. The temporary installation directory may be removed when Simics has been successfully installed.

Installing the License

Copy the license file, that you should have received by email, to `[simics]/licenses`:

```
# cp simics-license-hostid.lic /opt/virtutech/simics-<version>/licenses/
```

Note: Once FLEXlm has successfully checked out a license it will add the path to the license search path (if it is not already present) in the `.flexlmrc` file in the user's home directory. If it takes a long time when Simics checks out the license, check this file for old incorrect paths to remove.

Installing Disk Images

A disk image, also called *disk dump* represents the contents of a hard disk, usually in a compressed form. Since images can be quite big, they are not provided in the standard Simics packages, but have to be downloaded separately. Publicly available image files can be downloaded from the Simics web site, `https://www.simics.net/`.

3.0.1 Single User Environment

In single user environments it is usually easiest to keep image files in the corresponding `[workspace]/targets/<architecture>/` directory. Simics will automatically look for files in this directory, and images do not have to be moved when Simics is upgraded.

3.0.2 Multi User Environment

To allow several users to share the same image files, it is recommended that images are placed in a shared directory in the file-system, for example in `/opt/virtutech/images/`. After installing Simics, a reference to this directory should be added in the installation wide startup file `[simics]/config/startup-commands`:

```
add-directory /opt/virtutech/images/
```

Chapter 4

User Workspace Setup

A Simics User Workspace is a directory containing Simics files such as user developed modules and customized setup scripts. The `[simics]/bin/workspace-setup` utility simplifies the management of workspaces, for example upgrading to a new Simics version.

4.1 Creating a User Workspace

To create a workspace, run the `workspace-setup` with the name of a workspace directory to create. It is also possible to run the script with no arguments from an empty directory.

```
$ [simics]/bin/workspace-setup workspace
```

This will create a workspace with the name `workspace`. The workspace will contain Simics start scripts, machine scripts and makefiles for building user-developed Simics modules.

To start Simics, without a machine configuration, run the `simics` script in the workspace directory:

```
$ cd workspace
$ ./simics
```

4.2 Upgrading to a New Version

To upgrade to a new Simics version, run the `workspace-setup` for the new version from an existing workspace directory. For example, if `workspace` was created for Simics 3.0.0, the following commands shows how it can be upgraded to Simics 3.0.1.

```
$ cd workspace
$ /opt/virtutech/simics-3.0.1/bin/workspace-setup
```

The new version does not have to be a more recent one. It is possible to upgrade a workspace to an older version of Simics, if it also supports user workspaces.

Chapter 5

Installing the Network Interface Helper

Simics provides many ways to connect the simulated network to the real network. They are listed in the network chapter of *Simics User Guide*. For some of them a small helper program, **openif**, needs to be installed. **openif** must run with super-user privileges. There are several ways of accomplishing this:

- Install **openif** as setuid root. This usually the easiest way. In the simplest case, it is just a matter of running these shell commands, replacing *host* with the host type you are running Simics on, for example, x86-linux:

  ```
  # cd [simics]
  # chown root host/bin/openif
  # chmod u+s host/bin/openif
  ```

  If the file system where Simics is installed does not allow setuid binaries, then it can be moved to some other location and either a symbolic link to the new location be made, or the Simics command **network-helper** be used to point out the new location.

  To verify that **openif** has been installed properly, try executing it directly from the shell as an ordinary user. If it complains about root privileges, it does not have the correct permission flags set and you may need to move it, perhaps to a directory on a local (not networked) file system.

  Since the **openif** program allows an unprivileged program to read and send raw packets from the network interface, you may want to restrict access to it. In this case, it can be put in a directory to which access is restricted using the standard Unix mechanisms. Its location then has to be indicated as above.

- If you have the common Unix utility **sudo** installed, it can be used to run **openif** from Simics by the means of a small wrapper script. This script would look like:

  ```
  #!/bin/sh
  ```

```
exec sudo /path/to/openif $*
```

Then use either a symbolic link or the **network-helper** command to tell Simics where to find the script. Note that **sudo** must be configured to allow the user to run **openif** as root, without giving a password.

- You can also run the entire Simics Central process as root. This alternative is not recommended, and may cause problems depending on the license model chosen.

The source code of **openif** is available as part of the Simics distribution, open to inspection and modification as needed.

Note that allowing unprivileged users to access the raw network device may cause security problems. Therefore it is not recommended to use the real-network connection in an untrusted network environment. A safer solution is to use the port forwarding feature in Simics instead.

Chapter 6

Simics Directory Structure

The Virtutech Simics environment, regardless of what components are included (source code or not, etc), is currently intended to have the following structure. Note that some of these files or directories may not be included in your particular package.

Directory/File	**Description**
`simics/Version`	version and timestamp for this distribution
`simics/RELEASENOTES`	description of the current Simics subversion changes
`simics/LIMITATIONS`	known problems and limitations
`simics/README.previous/`	README files for previous Simics version
`simics/LICENSE`	Simics license
`simics/licenses/`	FLEXlm license files goes here
`simics/bin/`	various utilities
`simics/scripts/`	various tools and scripts needed for compilation
`simics/src/`	main source code directory
`simics/src/devices/`	main device module directory
`simics/src/devices/FAS366U/`	a typical SCSI device . . .
`simics/src/devices/..`	.. etc
`simics/src/extensions/`	main extension module directory
`simics/src/extensions/trace/`	generic instruction trace tool
`simics/src/extensions/..`	.. etc
`simics/src/include/`	include files for Simics API
`simics/src/misc/`	source code for external programs
`simics/config/`	generic configuration files for make
`simics/config/masters/`	main makefiles
`simics/doc/`	Simics documentation
`simics/doc/*-one/`	single-document HTML documentation (if any)
`simics/doc/`*directories*`/`	multi-document HTML documentation
`simics/`*host*`/`	directory with files for named *host* type
`simics/`*host*`/bin/`	Simics binaries files

simics/*host*/lib/	Simics loadable modules
simics/*host*/lib/python/	Simics Python files
simics/*host*/lib/python-*x.y*/	Standard Python files
simics/*host*/sys/flexlm/bin/	FLEXlm binaries
simics/*host*/sys/lib/	external libraries used by Simics
simics/targets/*	virtual machine configurations

A Simics workspace is intended to have the following structure.

Directory/File	**Description**
workspace/simics	Shortcut to start Simics in command-line mode
workspace/simics-eclipse	Shortcut to start Simics with the Eclipse frontend
workspace/GNUmakefile	Makefile to build all modules in this workspace
workspace/Workspace.mk	Inner makefile
workspace/Config.mk	Default definitions of make variables
workspace/Config-user.mk	Optional file with overridden make variables
workspace/modules/	Source code of user-developed modules
workspace/targets/*	Example target machine configurations
workspace/*host*/	Directory with user files for named *host* type
workspace/*host*/lib/	Simics loadable user-developed modules
workspace/*host*/lib/python/	User-developed Simics Python files.
workspace/*host*/obj/	scratch compilation area
workspace/.workspace-properties/	For internal bookkeeping

Virtutech, Inc.

1740 Technology Dr., suite 460
San Jose, CA 95110
USA

Phone +1 408-392-9150
Fax +1 408-608-0430

http://www.virtutech.com