



virtutech

Simics Installation Guide for Linux/Solaris

Simics Version 3.0

Revision 1376

Date 2007-01-24

© 1998–2006 Virtutech AB
Norrtullsgatan 15, SE-113 27 STOCKHOLM, Sweden

Trademarks

Virtutech, the Virtutech logo, Simics, and Hindsight are trademarks or registered trademarks of Virtutech AB or Virtutech, Inc. in the United States and/or other countries.

The contents herein are Documentation which are a subset of Licensed Software pursuant to the terms of the Virtutech Simics Software License Agreement (the “Agreement”), and are being distributed under the Agreement, and use of this Documentation is subject to the terms the Agreement.

This Publication is provided “as is” without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

This Publication could include technical inaccuracies or typographical errors. Changes are periodically added to the information herein; these changes will be incorporated in new editions of the Publication. Virtutech may make improvements and/or changes in the product(s) and/or the program(s) described in this Publication at any time.

Contents

1	About Simics Documentation	5
1.1	Conventions	5
1.2	Simics Guides and Manuals	5
	Simics Installation Guide for Unix and for Windows	5
	Simics User Guide for Unix and for Windows	6
	Simics Eclipse User Guide	6
	Simics Target Guides	6
	Simics Programming Guide	6
	DML Tutorial	6
	DML Reference Manual	6
	Simics Reference Manual	6
	Simics Micro-Architectural Interface	6
	RELEASENOTES and LIMITATIONS files	7
	Simics Technical FAQ	7
	Simics Support Forum	7
	Other Interesting Documents	7
2	System Requirements	8
3	Installing Simics	9
	Installing a Simics Package	9
	Installing the License	10
	Installing Disk Images	10
	3.0.1 Single User Environment	10
	3.0.2 Multi User Environment	10
4	User Workspace Setup	12
4.1	Creating a User Workspace	12
4.2	Upgrading to a New Version	12
5	FLEXlm Floating Licenses	13
6	.tf Encoded Files	16
7	Installing the Network Interface Helper	17

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 one or several `tar` archive files, for example `simics-pkg-00-3.0.0-linux.tar` where 00 is a customer specific package number, and 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, each user can create a Simics workspace as described in section 4.

Installing a Simics Package

1. Download the appropriate Simics packages from the download site. (The download URL and package numbers are distributed separately).
2. Unpack all package files by running `tar`.

```
$ tar xf simics-pkg-00-3.0.0-linux.tar
```

This will create a temporary installation directory called `simics-3.0-install`. One Simics base package and several add-on packages can be installed at the same time, by unpacking them with `tar` before continuing with the next step, running the install script. It is possible to install multiple base packages of the same package type, where only the host type differs, at the same time.

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. It will also ask for the decryption key for each package. If packages have been unpacked earlier with

the same package number and the same Simics version number, then the decode key has been cached in `$HOME/.simics-tfkeys`.

Note: It is not possible to install several Simics base packages in the same directory.

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

If you have a node-locked license file, simply copy it to the `[simics]/licenses/` directory. To install floating licenses, refer to section 5.

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

FLEXlm Floating Licenses

Simics needs a license server for *floating licenses*. The license server used is *FLEXlm v9.2* from Macrovision. Below is a brief description on how to set up the license server. For further information on how to manage the server please consult the *FLEXlm End Users Guide* which can be found at <http://www.macrovision.com/services/support/index.shtml>.

Note: FLEXlm is a very common license technology, and if you are a corporate user then it is likely that you already have a centrally managed corporate server. Ask your system administrator.

All files needed to setup the FLEXlm license server are included in the Simics distribution in `[simics]/host/sys/flexlm/bin`, e.g., `[simics]/x86-linux/sys/flexlm/bin`:

- `lmgrd` — The License Manager Daemon (server)
- `lmutil` — FLEXlm utility program(s)
- `vtech` — The vendor daemon from Virtutech

A license file from Virtutech is also needed (usually sent by e-mail), that specifies your licenses. The license file is typically called `simics.lic`.

`lmgrd`, `lmutil` and `vtech` can be installed anywhere in the system, for example in `/usr/local/flexlm/bin/`. The license file, e.g., `simics.lic`, should then be placed in the `[simics]/licenses/` directory. This is the default location for a license file, meaning that Simics will find it automatically. To override this, the environment variable `VTECH_LICENSE_FILE` can be set to the name of the license file or to the directory where it resides. It can also be set to `@license-host` (i.e., the machine running the license server) if the license file cannot be reached through the file system. In this case the license server will know the location of the license file. A typical license file looks like this:

```
SERVER server ANY
USE_SERVER
# change the path to where the license daemon (vtech) is located
```

```

VENDOR vtech /usr/local/flexlm/bin/vtech
#
FEATURE p_developer vtech 3.0 30-sep-2005 10 03DBBBBC9607 \
    VENDOR_STRING=Tc;Gcm HOSTID=ANY SIGN=30C8A736AB4A
#
FEATURE eclipse vtech 3.0 30-sep-2005 10 5E347B88E389 HOSTID=ANY \
    SIGN=421DDD8207A2
#
FEATURE host_rh6-x86 vtech 3.0 30-sep-2005 uncounted 3A06239E1993 \
    HOSTID=ANY SIGN=BC67891604C4
#

```

where `server` is the name of the host that will run the license server. This can be changed by the user if the line ends with `ANY`, otherwise the license server need to be run on a special host. The `VENDOR` line specifies the vendor and where to find the vendor daemon. Edit this line to point to the `vtech` daemon file. `FEATURE` specifies the type of license and for what it is used. In this case it is 10 commercial floating licenses for the *developer* product in Simics 3.0 on x86 Linux host with expire date 30-sep-2005. The last hexadecimal codes are checksums which assert that the license file is not altered in an unauthorized way by the user.

To be able to build user modules that can be loaded into Simics a *Simics model builder license* is needed. If you have such a license, the *p_modelbuilder* feature and *builder* feature will be present in the license file as well.

The license server can be started in the following way:

```
joe@server:~$ lmgrd -c simics.lic
```

Remember to run `lmgrd` on the `server` host specified in the license file; in the example it is “`server`”. Since `lmgrd` does not need root privileges it is recommended to run it as an ordinary user.

The server will log all actions to `stdout`, unless a log file is specified by the `-l log-file` option.

`lmgrd` will find the vendor daemon and start it. All communication between the licensed software and the vendor daemon goes via the license server (`lmgrd`). It is the vendor daemon that keeps track of products and number of licenses from a particular vendor, thus only one license server is needed to run software from several vendors.

Note: When upgrading the license on an already running license server, it is recommended that the server is stopped and then restarted, instead of using the “`reread`” feature in the FLEXlm utilities. The reason is that the FLEXlm server can’t handle some license changes properly. *Stopping and restarting is required if you have a server with still valid licenses for Simics 2.x.*

When the license server is running, Simics can be run like any program and the communication with the license server should be transparent to the user. When Simics is started,

it will contact the server and, if the user is authorized and available licenses exist, Simics will *check-out* a license and run. On exit, Simics will give back the license to the daemon (*check-in*).

Note: A model builder license is only checked out the first time a new module (compiled by a user) is loaded into Simics. Further use of the same module does not require a model builder license. Model builder licenses will be kept for 24 hours before they are returned to the license pool, but can be reused an unlimited number of times by the same user.

If Simics ever crashes or exits abnormally, a license may be lost for a while but will return to the license pool after about 2 hours.

To shut down the license server and the vendor daemon type:

```
joe@enterprise:~$ lmutil lmdown
```

and the server will go down.

Chapter 6

.tf Encoded Files

Some of the files proposed for download are encrypted in the `.tf` format. To decrypt them, the `tfdecode` utility is needed. It is included in the Simics distribution, in the `[simics]/bin/` directory.

```
$ tfdecode key < file.tf > file
```

where *key* should be replaced with the decryption key corresponding to *file.tf*.

Chapter 7

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 8

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

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

A Simics workspace is intended to have the following structure.

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