WIND RIVER

Wind River® Simics®

INSTALLATION GUIDE

4.6

Revision	4081
Date	2012-11-16

Corporate Headquarters
Wind River
500 Wind River Way
Alameda, CA 94501-1153
U.S.A.

Toll free (U.S.A.): 800-545-WIND
Telephone: 510-748-4100
Facsimile: 510-749-2010

For additional contact information, see the Wind River Web site:
`www.windriver.com`
For information on how to contact Customer Support, see:
`www.windriver.com/support`

Contents

Chapter 1

Introduction

This document describes how to install and configure Simics. It identifies the hardware and software requirements for installing and running Simics and provides instructions for installation and license configuration. This document should be read by users installing Simics themselves, and all administrators for Simics.

Chapter 2

Hardware and Software Requirements

Linux (32-bit)

- Host architecture: x86 (Pentium ® II or compatible)
- OS Version: Red Hat Enterprise Linux 5 or compatible *(1)*
- Compiler (one of): GCC 4.x *(2)*

Linux (64-bit)

- Host architecture: x86-64
- OS Version: Red Hat Enterprise Linux 5 or compatible *(1)*
- Compiler (one of): GCC 4.x *(2)*

Microsoft Windows (32-bit)

- Host architecture: x86 (Pentium ® II or compatible)
- OS Version: Windows XP SP1 (desktop), Windows Server 2003 (server) or later *(3)*
- Third-Party Software: WinPcap 3.0, TAP-Win32 (OpenVPN 2.0), MinGW *(4)*
- Compiler: Simics-provided GCC 4.x *(2)* or Microsoft Visual C++ .NET 2003, 2005, 2008 or 2010 *(5)*

Microsoft Windows (64-bit)

- Host architecture: x86-64
- OS Version: Windows Vista (desktop), Windows Server 2003 (server) or later *(3)*
- Third-Party Software: WTAP-Win32 (OpenVPN 2.0), MinGW *(4)*
- Compiler: MinGW GCC 4.x *(2)* or Microsoft Visual C++ .NET 2005, 2008 or 2010 *(5)*

All Hosts

- Memory: 2GB *(6)*
- Disk: Several GB *(7)*

(1) Should work also on any compatible Linux distribution. Minimal requirements are GNU libc 2.5.x or newer and GTK 2.10 for the GUI. Note that Simics does currently **not** work with SELinux in enforcing mode.

(2) A compiler is only needed when creating user-developed modules with the Simics Model Builder product.

(3) There are currently some issues with running the Simics installer on Windows XP SP1 or Windows 2003 with no service pack (bug 17520). Please contact technical support if you need to install Simics on one of these platforms. The installer works well with all newer Windows versions, including Windows XP SP2 and Windows 2003 SP1.

Also, some older Windows installations require newer versions of Microsoft Visual C++ Run-Time which are available at `http://www.microsoft.com/download/en/details.aspx?id=5555` and `http://www.microsoft.com/download/en/details.aspx?id=10015`.

(4) MinGW is only needed when creating user-developed modules with the Simics Model Builder product.

(6) In general about 2GB and at least as much memory as the working set in the simulated machine, in order to avoid swapping.

(7) Highly dependent on the amount of storage space used in the simulated machine. Simics itself requires a few hundred MB.

(5) The Microsoft Visual C++ compilers are only supported for C++ modules.

Chapter 3

Installation Overview

Simics comes in *packages*, each one containing different parts of the functionality. The packages can be combined to a create a customized installation. The packages include:

- A base package containing *Simics Hindsight*—the Simics execution engine and the user interface, as well as a library of standard models. It also contains *Simics Analyzer* and *Simics Ethernet Networking*.
- An add-on package with the *Firststeps* tutorial machine (a PowerPC-based card).
- An add-on package for each simulated system (usually containing a CPU core and various devices).
- Some add-on packages for additional functionality, like *Simics Model Builder*, which contains the DML compiler and related files.
- Source code packages for many of the simulated systems containing source code for some of the models used in the system.

The packages are are usually encrypted, and the decryption keys are provided to Wind River customers separately.

The installation process is simple, and is described in the following chapters. This is a quick overview of the steps involved:

1. Download the base package and the add-on packages for which you have a license. If some third-party software is necessary, it will be documented later in this guide (refer to the *Third-Party Products* sections).
2. Install the packages. If the package is encrypted, the decryption key is required for this step.
3. Since Simics is a license controlled application you need to install your license file and start a license server. In many cases, this step is completely automatic though.
4. Although it is not mandatory, it is usually a good idea to create a workspace where you can keep the files you are working with, while leaving the Simics installation read-only. It will make upgrading Simics much easier later on.

Note: Add-on packages are *not* installed on top of the main Simics installation. Instead each package stays in its own directory, and Simics is simply instructed to look for add-on packages in various places using the `addon-manager` program.

When installing Simics in a shared environment, several options are provided for multi-user installations. Refer to chapter 7 for a complete description.

Chapter 4

Linux Installation

4.1 First Installation

As a Simics customer, you should have received instructions that describes which packages you should download and where to find them. If this is not the case, contact Wind River to obtain this information.

1. Start by downloading the packages you want to install. We will use package 1000 (the base package) and package 4005 (the Firststeps add-on) as an example.

   ```
   joe@computer$ cd simics-download
   joe@computer$ ls
   simics-pkg-1000-4.6.0-linux64.tar
   simics-pkg-4005-4.6.0-linux64.tar
   ```

   The packages are `.tar` files that can be unpacked with the `tar` command. Unpacking the packages will create a a `simics-4.6-install` directory with the relevant installation files:

   ```
   joe@computer$ tar xf simics-pkg-1000-4.6.0-linux64.tar
   joe@computer$ tar xf simics-pkg-4005-4.6.0-linux64.tar
   joe@computer$ ls simics-4.6-install
   install_simics_common.pm
   install-simics.pl*
   package-1000-4.6.0-linux64.packageinfo.tf
   package-1000-4.6.0-linux64.tar.gz.tf
   package-4005-4.6.0-linux64.packageinfo.tf
   package-4005-4.6.0-linux64.tar.gz.tf
   installation-guide.pdf
   SLA-1.5
   SLA-1.5.rtf
   tfdecode-linux64*
   ```

The file `installation-guide.pdf` is a copy of this manual. The `tar.gz.tf` files contains the compressed and encrypted package contents to be installed. The installation script `install-simics.pl` is used in the next step of the installation.

2. Run the installation script:

```
joe@computer$ cd simics-4.6-install
joe@computer$ ./install-simics.pl
```

For the first installation, the script will start by asking you to read and accept the Simics Software License Agreement (SLA). If you or your company already signed a valid SLA for Simics 4.6, you can say so immediately. In the example session below, we answer no:

```
Have you or your company already signed a custom Simics Software ↩
License Agreement? (y, n) [n]: n
```

If you have a valid SLA, the installation will start immediately and you can skip to the next step. If you do not have a valid SLA, you will be asked where your company is located:

```
Is your company located in one of the following areas: USA, Canada, ↩
Japan, the European Union? (y, n): y
```

If your company is not located in any of those areas, you need to contact Wind River to obtain a custom license agreement before continuing the installation.

The installation script will then allow you to read the Simics SLA. It will also point you to a file that you can open and print with almost any text editor to read the SLA at your leisure. Once you read the SLA, you will be asked to accept (by typing “accept”) or decline it. Declining will cancel the installation process. Accepting will get you to the next installation step.

```
install-simics will now show you the Simics Software License Agreement
for Simics 4.6 (revision 1.5). You can also read or print this agreement
by opening the file 'SLA-1.5' (in the current directory) in any text editor.

To continue with the installation and review the terms of the SLA, enter 'y'
To abort the installation, enter 'n'. (y, n) [y]: <ENTER>
```

...Showing the SLA...

```
Do you accept the terms of the SLA? (accept, decline) [decline]: accept
```

3. At this point, you will be asked for the keys necessary to decrypt the packages. Keys are cached automatically once they have been found to be correct, so you should need to perform this step only once (for a given package):

```
-> Looking for Simics packages in current directory...
Enter a decryption key for package-1000-4.6.0-linux32.tar.gz.tf [Abort]: ↩
0123456789ABCDEF0123456789ABCDEF
Enter a decryption key for package-4005-4.6.0-linux32.tar.gz.tf [Abort]: ↩
0123456789ABCDEF0123456789ABCDEF
```

Note: These are not the actual keys.

4. The script will present a list of packages to install. Simply type the number listed in front of each package you wish to install. In the example we install all packages (Simics and Firststeps):

```
install-simics can install the following packages from the current directory:
   Number  Name          Type    Version   Host         File
    1      Simics-Base   simics  4.6.0     linux64      package-1000-4.6.0-[...]
    2      Firststeps    addon   4.6.0     linux64      package-4005-4.6.0-[...]
    3      All packages

Please enter the numbers of the packages you want to install, as in "1 4 3"
Package numbers, or Enter to [Abort]: 3
```

5. You will be asked for a destination directory. The default is `/opt/simics/simics-4.6`. In the example session below, we will accept the default:

```
Enter a destination directory for installation [/opt/simics/simics-4.6/]: <ENTER>
The directory '/opt/simics/simics-4.6/' does not exist.
Do you want to create it? (y, n) [y]: <ENTER>
```

6. The script will show you a summary of the installation options before proceeding. If you accept, the installation will be performed:

```
The following packages will be installed in /opt/simics/simics-4.6/:
   Simics-Base 4.6.0 (package-1000-4.6.0-linux64.tar.gz.tf)
                with key 0123456789ABCDEF0123456789ABCDEF
   Firststeps 4.6.0 (package-4005-4.6.0-linux64.tar.gz.tf)
                with key 0123456789ABCDEF0123456789ABCDEF
```

```
Do you wish to perform the installation? (y, n) [y]: <ENTER>
-> Decrypting package-1000-4.6.0-linux32.tar.gz.tf
-> Testing package-1000-4.6.0-linux32.tar.gz
-> Installing package-1000-4.6.0-linux32.tar.gz
-> Decrypting package-4005-4.6.0-linux32.tar.gz.tf
-> Testing package-4005-4.6.0-linux32.tar.gz
-> Installing package-4005-4.6.0-linux32.tar.gz
```

If everything was successful, you now have two directories in `/opt/simics/simics-4.6/` called `simics-4.6.0` and `simics-firststeps-4.6.0`.

Note: The `/opt/simics/simics-4.6/simics-4.6.0` directory in the example above is the directory referred to by the `[simics]` abbreviation in examples in the rest of the Simics documentation.

7. The installer will ask you whether you want to install your license file. This file should have been provided to you by Wind River and we assume you have saved it somewhere on your computer. Enter the full path to the license file here or press Enter to skip this part and manually manage the license file later on.

```
If you wish to, Simics can automatically install a license
file in your newly installed Simics. This can also be done later.
Enter the full path to your license file or Enter to [Skip]: simics-license.lic
-> Copying simics-license.lic to /opt/simics/simics-4.6/simics-4.6.0/licenses
```

8. The installer will help you finish the configuration of the packages you installed. The default choice—to configure the Firststeps add-on with the newly installed Simics—is exactly what we want:

```
install-simics has installed the following add-on package:
   Firststeps  4.6.0  /opt/simics/simics-4.6/simics-firststeps-4.6.0

Do you wish to make these add-on packages available in ↩
Simics-Base 4.6.0? (y, n) [y]: <ENTER>
-> Making add-on package available in Simics-Base 4.6.0
```

9. The installer should now summarize the installation and report that everything was successful.

10. If you skipped the installation of the license file above you can install it manually. Copy the license file to *[simics]*`/licenses`:

```
joe@computer$ cp my_license.lic [simics]/licenses/
```

where my_license.lic is the license file you received from Wind River.

11. You can run Simics directly with the following command:

```
joe@computer$ /opt/simics/simics-4.6/simics-4.6.0/bin/simics-gui
```

You can also create a workspace first, where you can keep your own modified scripts and the files you are working with:

```
joe@computer$ /opt/simics/simics-4.6/simics-4.6.0/bin/workspace-setup↵
 simics-workspace
joe@computer$ cd simics-workspace
joe@computer$ ./simics-gui
```

If you are new to Simics, it is probably time to open the *Getting Started* document for some reading and tutorial. You will find it in the `doc/` directory of your Simics installation or your workspace, as well as in the Simics on-line help.

4.2 Academic Users: First Installation

As an academic user, you have access to a number of Simics packages in your download area.

1. Start by downloading the packages you want to install. We will use package 1000 (the base package) and package 4005 (the Firststeps add-on) as an example.

```
joe@computer$ cd simics-download
joe@computer$ ls
simics-pkg-firststeps-4.6.0-linux64.tar
simics-pkg-simics-base-4.6.0-linux64.tar
```

The packages are `.tar` files that can be unpacked with the `tar` command. Unpacking the packages will create a `simics-4.6-install` directory with the relevant installation files:

```
joe@computer$ tar xf simics-pkg-simics-base-4.6.0-linux64.tar
joe@computer$ tar xf simics-pkg-firststeps-4.6.0-linux64.tar
joe@computer$ ls simics-4.6-install
firststeps-4.6.0-linux64.packageinfo
```

```
firststeps-4.6.0-linux64.tar.gz
install_simics_common.pm
install-simics.pl*
simics-base-4.6.0-linux64.packageinfo
simics-base-4.6.0-linux64.tar.gz
installation-guide.pdf
SLA-academic-1.5
SLA-academic-1.5.rtf
```

The file `installation-guide.pdf` is a copy of this manual. The `tar.gz.tf` files contains the compressed package contents to be installed. The installation script `install-simics.pl` is used in the next step of the installation.

2. Run the installation script:

```
joe@computer$ cd simics-4.6-install
joe@computer$ ./install-simics.pl
```

For the first installation, the script will start by asking you to read and accept the Simics Software License Agreement (SLA). Besides showing you the SLA online, it will point you to a file that you can open and print with almost any text editor to read the SLA at your leisure. Once you read the SLA, you will be asked to accept (by typing "accept") or decline it. Declining will cancel the installation process. Accepting will get you to the next installation step.

```
install-simics will now show you the Simics Software License Agreement
for Simics 4.6 (revision 1.5). You can also read or print this agreement
by opening the file 'SLA-academic-1.5' (in the current directory) in any text editor.

To continue with the installation and review the terms of the SLA, enter 'y'
To abort the installation, enter 'n'. (y, n) [y]: <ENTER>
```

...Showing the SLA...

```
Do you accept the terms of the SLA? (accept, decline) [decline]: accept
```

3. The script will present a list of packages to install. Simply type the number listed in front of each package you wish to install. In the example we install all packages (Simics and Firststeps):

```
install-simics can install the following packages from the current directory:
   Number  Name          Type    Version Host         File
    1      Simics-Base   simics  4.6.0   linux64      simics-base-4.6.0-[...]
    2      Firststeps    addon   4.6.0   linux64      firststeps-4.6.0-[...]
    3      All packages

Please enter the numbers of the packages you want to install, as in "1 4 3"
Package numbers, or Enter to [Abort]: 3
```

4. You will be asked for a destination directory. The default is `/opt/simics/simics/4.6`. In the example session below, we will accept the default:

```
Enter a destination directory for installation [/opt/simics/simics-4.6/]: <ENTER>
The directory '/opt/simics/simics-4.6/' does not exist.
Do you want to create it? (y, n) [y]: <ENTER>
```

5. The script will show you a summary of the installation options before proceeding. If you accept, the installation will be performed:

```
The following packages will be installed in /opt/simics/simics-4.6/:
   Simics-Base 4.6.0 (simics-base-4.6.0-linux64.tar.gz)
   Firststeps 4.6.0 (firststeps-4.6.0-linux64.tar.gz)

Do you wish to perform the installation? (y, n) [y]: <ENTER>
-> Testing simics-base-4.6.0-linux64.tar.gz
-> Installing simics-base-4.6.0-linux64.tar.gz
-> Testing firststeps-4.6.0-linux64.tar.gz
-> Installing firststeps-4.6.0-linux64.tar.gz
```

If everything was successful, you now have two directories in `/opt/simics/simics-4.6/` called `simics-4.6.0` and `simics-firststeps-4.6.0`.

Note: The `/opt/simics/simics-4.6/simics-4.6.0` directory in the example above is the directory referred to by the `[simics]` abbreviation in examples in the rest of the Simics documentation.

6. The installer will ask you whether you want to install your license file. This file should have been provided to you by Wind River and we assume you have saved it somewhere on your computer. Enter the full path to the license file here or press Enter to skip this part and manually manage the license file later on.

```
If you wish to, Simics can automatically install a license
file in your newly installed Simics. This can also be done later.
Enter the full path to your license file or Enter to [Skip]: simics-license.lic
-> Copying simics-license.lic to /opt/simics/simics-4.6/simics-4.6.0/licenses
```

7. The installer will help you finish the configuration of the packages you installed. The default choice—to configure the Firststeps add-on with the newly installed Simics—is exactly what we want:

   ```
   install-simics has installed the following add-on package:
      Firststeps  4.6.0  /opt/simics/simics-4.6/simics-firststeps-4.6.0

   Do you wish to make these add-on packages available in ↩
   Simics-Base 4.6.0? (y, n) [y]: <ENTER>
   -> Making add-on package available in Simics-Base 4.6.0
   ```

8. The installer should now summarize the installation and report that everything was successful.

9. If you skipped the installation of the license file above you can install it manually. Copy the license file to *[simics]*`/licenses/`:

   ```
   joe@computer$ cp my_license.lic [simics]/licenses/
   ```

   where my_license.lic is the license file you received from Wind River.

10. You can run Simics directly with the following command:

    ```
    joe@computer$ /opt/simics/simics-4.6/simics-4.6.0/bin/simics-gui
    ```

    You can also create a workspace first, where you can keep your own modified scripts and the files you are working with:

    ```
    joe@computer$ /opt/simics/simics-4.6/simics-4.6.0/bin/workspace-setup↩
     simics-workspace
    joe@computer$ cd simics-workspace
    joe@computer$ ./simics-gui
    ```

    If you are new to Simics, it is probably time to open the *Getting Started* document for some reading and tutorial. You will find it in the `doc/` directory of your Simics installation or your workspace, as well as in the Simics on-line help.

4.3 Installing More Packages

You can install more packages at any time using the same steps as above. The easiest way is to install all packages in the same directory (like `/opt/simics/simics-4.6` above).

The installer will propose a default configuration that should always work with the latest installed version of Simics.

If you want to handle the installation differently, you should read the next sections about add-on package management, the installer command line switches and the `addon-manager` options.

4.4 Uninstalling Simics

Uninstalling Simics is very simple: you just need to remove the directories that were created for each package during installation. Note the following:

- If you delete an add-on package that is still referenced in the add-on packages list of a Simics installation, the add-on package will simply be considered invalid and ignored. The `workspace-setup` program will however suggest that you update workspaces that were using that add-on package.
- If you delete a Simics installation, you will need to reconfigure the workspaces that depended on it by running the `workspace-setup` program from another Simics installation.
- See chapter 9 for information on how to shut down the license server.

4.5 Installing VMP

VMP uses direct execution to simulate x86 systems at near native speed. Kernel modules are needed to communicate directly with the host hardware, and installing those kernel modules requires a separate step.

Change directory to the Simics Base package and run:

```
joe@computer$ ./scripts/vmp-kernel-install.sh
```

The script will compile and install the kernel modules that are used by VMP. The script needs to be run after every reboot (unless you setup the system to load the kernel modules automatically).

See the *VMP* section in the *Simulation Performance* chapter of the *Hindsight User's Guide* for information about enabling and using VMP.

4.6 Installing the Network Interface Helper

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

- Set up **openif** as setuid root. 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, `linux64`:

  ```
  # 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, you can move **openif** to some location which allows setuid binaries and create a symbolic link which points to the moved file.

  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. As above, use a symbolic link to tell Simics where to find it.

- 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 a symbolic link 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 process connected to a real network 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 5

Windows Installation

5.1 First Installation

As a Simics customer, you received a mail (or instructions) that describes which packages you should download and where to find them. If this is not the case, contact Wind River to obtain this information.

1. Start by downloading the packages you want to install. We will use package 1000 (Simics itself) and package 4005 (the Firststeps add-on) as an example. These packages are `.exe` installers.

2. Run the installer for package 1000. You will first need to provide a decryption key for the package. Keys are cached automatically once they have been found to be correct, so you should need to perform this step only once (for a given package).

   If you provide an incorrect key, or if the package itself was damaged during the download process, the installer will warn you at the end of the decryption phase that something went wrong. Check carefully the key you entered and ensure that the download process was finished correctly (check the file size for example). If this does not solve the problem, contact Wind River on the support forum.

3. Once the installer was decrypted successfully, it will simply run through a number of standard steps before installing the product itself:

   - Indicate whether you already signed a custom SLA for Simics 4.6 or not. If you have, the installation will directly go two steps further.
   - If you have not signed any custom agreement, the installer will ask you where your company is located, in order to check if it is covered by our standard Software License Agreement (SLA). If the standard agreement does not apply, you will need to contact Wind River to obtain a custom SLA.
   - Read and accept the SLA if you agree with the terms.
   - Choose an installation directory. The default directory is `C:\Program Files (x86)\Simics\Simics 4.6\Simics 4.6.x` where x is the minor version.

- Tell the installer where it can find the license file. This file should have been provided to you by Wind River and we assume you have saved it somewhere on your computer. You can skip this part and do it manually later, see below.

4. If you skipped the installation of the license file above you can install it manually. Copy the license file to `C:\Program Files (x86)\Simics\Simics 4.6\Simics 4.6.x\licenses\` where x is the minor version.

5. Once package 1000 is installed and the license server is running, Simics is ready to run on your computer. However, no simulated target system is installed, so it will be of limited use. Package 4005 (Firststeps) is an example of a target system and its installation is fairly similar to the base package:

   (a) Enter the decryption key.
   (b) The Software License Agreement was already accepted when installing the base package, so there is no need to accept it again.
   (c) Choose an installation directory.
   (d) Locate a Simics installation for registering this add-on package. As you just installed Simics before, it should be suggested automatically and you can simply press Next to continue.

6. Simics and the Firststeps packages are now installed on your computer. You can start Simics by simply going to the Start Menu and clicking on Simics.

   When Simics starts, it will suggest that you create a workspace, to keep your own modified scripts and the files you are working with. This is a good idea unless you have special needs and know how to work without a workspace.

   If you are new to Simics, it is a good time to open the *Getting Started* document for some reading and tutorial. You will find it in the Start Menu under the Documentation folder, or in the Simics on-line help.

Note: If the list views in the GUI beeps on Windows Vista, then the following workaround can be applied: `http://support.microsoft.com/kb/944150`

Note: If the installation, or if the `addon-manager`, fails to complete it may be that some older Windows installations require newer versions of Microsoft Visual C++ Run-Time which are available at `http://www.microsoft.com/download/en/details.aspx?id=5555` and `http://www.microsoft.com/download/en/details.aspx?id=10015`.

5.2 Academic Users: First Installation

As an academic user, you have access to a number of Simics packages in your download area.

1. Start by downloading the packages you want to install. We will use the Simics-Base package (Simics itself) and the Firststeps add-on as an example. These packages are normal `.msi` installers.

2. Double-click on the installer for the Simics-Base package. It will simply run through a number of standard steps before installing the product itself:

   - If you have not done so already, read and accept the Simics Software License Agreement.
   - Choose an installation directory. The default directory is `C:\Program Files (x86)\Simics\Simics 4.6\Simics 4.6.x` where x is the minor version.
   - Tell the installer where it can find the license file. This file should have been provided to you by Wind River and we assume you have saved it somewhere on your computer. You can skip this part and do it manually later, see below.

3. If you skipped the installation of the license file above you can install it manually. Copy the license file to `C:\Program Files (x86)\Simics\Simics 4.6\Simics 4.6.x\licenses\` where x is the minor version.

4. Once the Simics-Base package is installed, Simics is ready to run on your computer. However, no simulated target system is installed, so it will be of limited use. The Firststeps package is an example of a target system and its installation is fairly similar to the base package:

   - The Software License Agreement was already accepted when installing the base package, so there is no need to accept it again.
   - Choose an installation directory.
   - Locate a Simics installation for registering this add-on package. As you just installed Simics before, it should be suggested automatically and you can simply press Next to continue.

5. Simics and the Firststeps packages are now installed on your computer. You can start Simics by simply going to the Start Menu and clicking on Simics.

   When Simics starts, it will suggest that you create a workspace, to keep your own modified scripts and the files you are working with. This is a good idea unless you have special needs and know how to work without a workspace. you know exactly what you are doing.

   If you are new to Simics, it is a good time to open the *Getting Started* document for some reading and tutorial. You will find it in the Start Menu under the Documentation folder, or in the Simics on-line help.

5.3 Installing More Packages

You can install more packages at any time using the same steps as above. The installer will suggest a default configuration that should always work with the latest installed version of Simics.

If you want to handle the installation differently, you should read the next sections about add-on package management and the `addon-manager` options.

5.4 Uninstalling Simics

To uninstall Simics, use Add or Remove Programs in the Control Panel. Note the following:

- If you delete an add-on package that is still referenced in the add-on packages list of a Simics installation, the add-on package will simply be considered invalid and ignored. The `workspace-setup` program will however suggest that you update workspaces that were using that add-on.
- If you remove a Simics installation, you will need to reconfigure the workspaces that depended on it the next time you want to use them. You can also do that by running the `workspace-setup` program from another Simics installation.
- See chapter 9 for information on how to shut down the license server.

5.5 Third-Party Products and Other Add-ons

Some functionality in Simics requires third-party products. These are: TAP-Win32, WinPcap 3.0, FabulaTech Virtual Serial Port Kit, and MinGW.

The port-forwarding kind of real network connection, included in Simics Hindsight, does not require any third party software. But the real network connections provided by Simics Ethernet Networking use TAP-Win32 or WinPcap to access the host network. TAP-Win32 is only needed when using TAP interface connections and WinPcap when using raw Ethernet connections. When a port-forwarding connection cannot be used, it is recommended to use a TAP interface to access the host network. Raw access should be avoided if possible.

To learn more about the different real network connection types provided by Simics Ethernet Networking read the *Ethernet Networking User's Guide*.

Simics requires the FabulaTech Virtual Serial Port Kit program to create virtual COM ports on Windows hosts.

If you plan to create new models in Simics, you will need the MinGW tools to compile you models.

TAP-Win32 (from OpenVPN 2.0) Installation

TAP-Win32 is distributed as a part of OpenVPN and needs to be installed separately from Simics. You do not need to install any other components of OpenVPN except TAP-Win32. You can download OpenVPN 2.0 at `http://openvpn.net/`.

The simplest way to install it is to download and run the windows installer, and in the installation options stage of the installer, uncheck all parts except the TAP device. Users will need administrative privileges on the simulation host to connect to the real network using TAP-Win32. Multiple TAP devices can be installed if needed. For details on configuring and adding additional TAP devices see the *TAP Access* section of the *Ethernet Networking User's Guide*.

Note: Simics uses the MAC addresses of the simulation host's network interfaces to identify the host for licensing purposes. The licensing system does not differentiate TAP-Win32 interfaces from real Ethernet interfaces. Creating a TAP-Win32 interface (which is done automatically when installing TAP-Win32) may therefore cause your license to stop working if you are using a node-locked or evaluation license, see the *TAP Access* section of the *Ethernet Networking User's Guide* for details on configuring your TAP device to minimize this chance.

WinPcap 3.0 Installation

WinPcap 3.0 is not included in the Simics installer, so you need to download and install it separately. You will find it at `http://www.winpcap.org/`.

WinPcap relies on a Windows service called NPF (Netgroup Packet Filter) to capture and send traffic on the real network. By default, the NPF service will be started automatically the first time someone tries to use it. However, starting a service requires administrative privileges, so users will need administrative privileges the first time they connect to the real network after the simulation host has booted. To start the NPF service automatically when the simulation host boots, so that users always can connect to the real network without administrative privileges, change the registry key `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NPF\Start` from `0x3` (`SERVICE_DEMAND_START`) to `0x2` (`SERVICE_AUTO_START`).

Fabulatech Virtual Serial Port Kit

To connect **host-serial-console** to a virtual serial port it must have been created in advance. Simics includes a program from FabulaTech to create such a port. The installer is located in the Simics Base package in *[simics]\[hosttype]*`\bin\setup-vspk.msi`. Install it and see the installed documentation for use of the program.

MinGW Installation

MinGW provides the GCC compiler and the make program that is needed to compile Simics modules on Windows. The MinGW tools can be downloaded from `https://www.simics.net/pub/`.

A package for with MinGW tools for 32-bit Windows known to work with Simics, is available at `https://www.simics.net/pub/MinGW-Simics-4.4.1.msi`.

For 64-bit windows there is no package available, but it is possible to download a Zip-file from `https://www.simics.net/pub/mingw64-i686-20110207.zip` containing the required files. If you extract this under `C:\` everything will work out of the box. You can also install the required tools directly from the MinGW-64 project. If you do this, or install in a different location you will need to update the file `workspace.py` in the `scripts` directory in the Simics installation directory. Look for the string "C:\mingw64-i686-20110207" and replace it with the path to your own installation.

Chapter 6

Upgrading Simics

6.1 Upgrading from Simics 3.2, 4.0, 4.2 and 4.4

Simics 4.6 is not directly compatible with Simics 3.2, 4.0, 4.2 and 4.4, in that add-on packages for those versions will not work with Simics 4.6 and vice versa. However, Simics 4.6 supports the same kind of installation and multi-user configuration, so upgrading should be very simple.

Simics 4.6 is API but not ABI compatible with Simics 3.2, 4.0, 4.2 and 4.4. This means that user-written modules in workspaces for those version have to be recompiled before they can be used with Simics 4.6. When recompiling, the 3.2, 4.0, 4.2 or 4.4 API has to be selected to get access to deprecated parts of the API. This is described in the *Simics Migration Guide*.

6.2 Upgrading between Minor 4.6.x Releases

Simics 4.6.x releases are binary compatible, so nothing special needs to be done when upgrading. Unless specifically mentioned in the *Release Notes*, add-on packages should work with any Simics 4.6 installation. A more detailed description of compatibility issues for user-created modules is available in the *Simics Migration Guide*.

Chapter 7

Advanced Installation Options

7.1 Unix: Add-on Packages Management

In Simics, all packages are installed separately in different directories. This makes it easy to upgrade or remove a package without touching the rest of the installation. Each Simics installation keeps its own list of paths to where it can find add-on packages, called the add-on package list. It makes them available to the Simics engine at run-time. This list can be printed and configured using the `addon-manager` program.

Here are a few examples of what the `addon-manager` can do:

Listing the add-on packages configured

To print out the add-on packages configured in a given installation of Simics, do the following:

```
joe@computer$ [simics]/bin/addon-manager

Simics 4.6 Add-on Package Manager
=================================

This script will configure this Simics installation to use optional
Simics add-on packages.
Default alternatives are enclosed in square brackets ([ ]).

=== Using the package list in Simics installation ↩
(/opt/simics/simics-4.6/simics-4.6.0) ===

Configured add-on packages:
   Firststeps  4.6.0  (linux64)  ../simics-firststeps-4.6.0
```

If there are any add-on packages that are invalid (the directory does not exist anymore, or the necessary information files are invalid), `addon-manager` will ask you if they can be removed. Refer to the paragraph below describing how to remove add-on packages that have been uninstalled.

Add an add-on package to the list

To add an installed add-on package to the list, do the following:

```
joe@computer$ [simics]/bin/addon-manager -s /opt/[...]/simics-x86-440bx-4.6.0

[...]
Configured add-on packages:
   Firststeps  4.6.0  (linux64)  ../simics-firststeps-4.6.0

The following operations will be performed:
   -> Add  X86-440BX  4.6.0  (linux64)  ../simics-x86-440bx-4.6.0

New package list:
   Firststeps  4.6.0  (linux64)  ../simics-firststeps-4.6.0
   X86-440BX   4.6.0  (linux64)  ../simics-x86-440bx-4.6.0

Do you want to update the package list? (y/n) [y] <ENTER>
```

Whenever you install a new add-on package, the installer will suggest to register it to the most likely Simics installation found, so this step will often be unnecessary.

Remove an add-on package from the list

To remove an add-on package from the list, do the following:

```
joe@computer$ [simics]/bin/addon-manager -d /opt/[...]/simics-x86-440bx-4.6.0/

[...]

Configured add-on packages:
   Firststeps  4.6.0  (linux64)  ../simics-firststeps-4.6.0
   X86-440BX   4.6.0  (linux64)  ../simics-x86-440bx-4.6.0

The following operations will be performed:
   -> Remove  X86-440BX  4.6.0  (linux64)  ../simics-x86-440bx-4.6.0

New package list:
   Firststeps  4.6.0  (linux64)  ../simics-firststeps-4.6.0

Do you want to update the package list? (y/n) [y] <ENTER>
```

Remove an uninstalled add-on package from the list

`addon-manager` will automatically propose to remove add-on packages that are not valid (the path does not exist, or the necessary information files are not present). In that case, you just need to accept to update the configuration.

```
joe@computer$ [simics]/bin/addon-manager

[...]

Configured add-on packages:
   Firststeps    4.6.0  (linux64)  ../simics-firststeps-4.6.0
   (invalid: missing)                  ../simics-x86-440bx-4.6.0

The following operations will be performed:
   -> Remove  (invalid: missing)       ../simics-x86-440bx-4.6.0

New package list:
   Firststeps  4.6.0  (linux64)  ../simics-firststeps-4.6.0

Do you want to update the package list? (y/n) [y] <ENTER>
```

Re-use the add-on packages configuration from another Simics

When installing a newer version of Simics, you may want to re-use the same add-on packages you used to have in the previous version. This is done by *upgrading* from the previous installation:

```
joe@computer$ /opt/[...]/simics-4.6.1/bin/addon-manager ↩
 -u /opt/[...]/simics-4.6.0/

[...]

Package list successfully updated from '/opt/[...]/simics-4.6.0/'.
The previous package list has been saved as ↩
'/opt/[...]/simics-4.6.1/.package-list.backup'.

Configured add-on packages:
   X86-440BX  4.6.0  linux64  ../simics-x86-440bx-4.6.0

joe@computer$
```

If the installer finds a previous Simics version installed, it will suggest an automatic upgrade to the newer version.

7.2 Windows: Add-on Packages Management

In Simics 4.6, all packages are installed separately, in different directories. This makes it easy to upgrade or remove a package without touching the rest of the installation. Each

Simics installation keeps its own list of paths to where it can find add-on packages, called the add-on package list. It makes them available to the Simics engine at run-time. This list can be printed and configured using the `addon-manager` program. Unfortunately, this program is only available from the command line at this moment, so you will have to run it from a `cmd.exe` window.

Note: When running `addon-manager.bat` in a Cygwin terminal, you may have problems with paths including spaces, even when quoting them properly. This is a Cygwin-specific problem. You can work-around the problem by using `cmd.exe` for that specific command, or by installing Simics in a directory that does not contain spaces.

Here are a few examples of what the `addon-manager` can do:

Listing the add-on packages available

To print out the add-on packages configured in a given installation of Simics, do the following:

```
C:\>"c:\Program Files\Simics\Simics 4.6\Simics 4.6.0\bin\addon-manager.bat"

Simics 4.6 Add-on Package Manager
=================================

This script will configure this Simics installation to use optional
Simics add-on packages.
Default alternatives are enclosed in square brackets ([ ]).


Configured add-on packages:
   Firststeps  4.6.0  win32  c:\Program Files\[...]\Firststeps 4.6.0
```

If there are any add-on packages that are invalid (the directory does not exist anymore, or the necessary information files are invalid), `addon-manager` will ask if they can be removed. Refer to the paragraph below describing how to remove add-on packages that have been uninstalled.

Add an add-on package to the list

To add an installed add-on package to the list, do the following:

```
C:\>"c:\Program Files\[...]\Simics 4.6.0\bin\addon-manager.bat" ↩
-s "c:\Program Files\[...]\X86-440BX 4.6.0"

[...]
Configured add-on packages:
   Firststeps  4.6.0  win32  c:\Program Files\[...]\Firststeps 4.6.0
```

```
The following operations will be performed:
   -> Add  X86-440BX  4.6.0  win32  c:\Program Files\[...]\X86-440BX 4.6.0

New package list:
   X86-440BX   4.6.0  win32  c:\Program Files\[...]\X86-440BX 4.6.0
   Firststeps  4.6.0  win32  c:\Program Files\[...]\Firststeps 4.6.0

Do you want to update the configuration? (y, n) [y]: <ENTER>
```

Whenever you install a new add-on package, the installer will propose to register it to the most likely Simics installation found, so this step will often be unnecessary.

Remove an add-on package from the list

To remove an add-on package from the list, do the following:

```
C:\>"c:\Program Files\[...]\Simics 4.6.0\bin\addon-manager.bat" ↩
-d "c:\Program Files\[...]\X86-440BX 4.6.0"

[...]

Configured add-on packages:
   X86-440BX   4.6.0  win32  c:\Program Files\[...]\X86-440BX 4.6.0
   Firststeps  4.6.0  win32  c:\Program Files\[...]\Firststeps 4.6.0

The following operations will be performed:
   -> Remove  X86-440BX  4.6.0  win32  c:\Program Files\[...]\X86-440BX 4.6.0

New package list:
   Firststeps  4.6.0  win32  c:\Program Files\[...]\Firststeps 4.6.0

Do you want to update the configuration? (y, n) [y]: <ENTER>
```

Remove an uninstalled add-on package from the list

`addon-manager` will automatically propose to remove add-on packages that are not valid (the path does not exist, or the necessary information files are not present). In that case, you just need to accept to update the configuration.

```
C:\>"c:\Program Files\[...]\Simics 4.6.0\bin\addon-manager.bat"

[...]

Configured add-on packages:
   Firststeps  4.6.0  win32  c:\Program Files\[...]\Firststeps 4.6.0
   (invalid: missing)        c:\Program Files\[...]\X86-440BX 4.6.0
```

```
The following operations will be performed:
   -> Remove  (invalid: missing) c:\Program Files\[...]\X86-440BX 4.6.0

New package list:
   Firststeps  4.6.0  win32  c:\Program Files\[...]\Firststeps 4.6.0

Do you want to update the configuration? (y, n) [y]: <ENTER>
```

Re-use the add-on packages configuration from another Simics

When installing a newer version of Simics, you may want to re-use the same add-on packages you used to have in the previous version. This is done by *upgrading* from the previous installation:

```
C:\>"c:\Program Files\[...]\Simics 4.6.1\bin\addon-manager.bat" ↩
-u "c:\Program Files\[...]\Simics 4.6.0"

[...]

Package list successfully updated from 'c:\Program Files\[...].0'.
The previous package list has been saved as ↩
'c:\Program Files\[...].1\.package-list.backup'.

Configured add-on packages:
   X86-440BX  4.6.0  win32  c:\Program Files\[...]\X86-440BX 4.6.0
```

If the installer finds a previous Simics version installed, it will propose you to upgrade automatically when installing a new version.

7.3 Add-on Packages: Module Lookup

Simics automatically recognizes modules that are located in the current workspace, in any of the configured add-on packages, and in the base installation. When confronted with several versions of the same modules, it applies the following selection rules:

User Modules

This includes the modules compiled in the current workspace as well as the modules found in paths specified by the *-L* command-line option, the `SIMICS_EXTRA_LIB` environment variable, or the **add-module-directory** command.

Packaged Modules

This include all modules provided by Simics itself or its add-on packages.

When the same module is found in several locations, user modules are always chosen in preference to packaged modules: if you recompile a module provided as source code in your workspace, the workspace version will always be chosen first, as long as it is possible to load it into Simics.

Within the same group of modules, priority is given to the newest module, according to the build date that is compiled in the module at linking time. For example, when two add-on packages provide the same module, Simics will always load the newest one.

7.4 Multi-user Installations

7.4.1 Shared Installations

By default, Simics is organized so that the installed packages can be left read-only; users work in a workspace, as described in our manuals, or in an appropriate work area. This makes it easy to share a Simics installation since the packages can be simply installed in a common location accessible to all users. In this case, all of them will share the same combination of add-on packages, essentially the exact same Simics installation.

7.4.2 Per-user Configuration

Simics can be installed so that add-on package configuration is done on a per-user basis. Each user will configure the specific add-on packages it wishes to use. This configuration can be stored in a workspace (the easiest alternative) or in a separate file. Per-user configuration lets you install Simics in a read-only location while giving users full freedom to configure and run with the add-on packages of their choice.

By default a workspace inherits the package associations of the main installation: When a package is installed in the main installation, this affects Simics when invoked from the workspace. When per-user configuration i used, a workspace instead manages add-on packages independently of the main installation, via the `addon-manager` script located in the workspace. The workspace will not be affected by the package associations of the main installation. In order to enable per-user configuration in a workspace, run the `addon-manager` script located in that workspace, with the *-c* or *-C* option. With *-c*, the initial workspace-local configuration will have no package associations; with -C, it will be based on the set of packages configured in the main installation at that time. Local configuration is done with further invocations of the `addon-manager` script in the workspace.

The add-on package configuration can also be stored in a specific file, by using the *-f* option of the `addon-manager` script. Configuration is identical to the other cases, but Simics must be informed explicitly of the existence of the file via its own command line argument *-package-file*.

7.4.3 Installation under Configuration Management Systems

If the Simics packages are imported in a configuration management system such as ClearCase or CVS, different options are available:

- If each user can check-out and modify its own copy of the package installation, they might use a common configuration or customize it. Relative paths in the add-on configuration will be useful to let users check-out the Simics installation wherever they wish.
- If the installation must be kept read-only, configurations can be performed either globally, when importing the Simics packages, or on a per-user basis, as described in the previous section.

7.4.4 Multi-user Installation on Windows

At this point, Simics packages are meant to be installed per machine. User installation is not supported. Simics packages can be installed side by side without risks, as each release of a given package is independent from all others.

Installed packages are read-only, and all the instructions above can be followed to create global or per-user configurations.

Except for academic users, the current packages are encrypted and cannot be installed in silent mode.

7.5 Unix: Installer Options

The Simics installer script provides command line options to directly specify the packages to install and the way it should be done. When command line options are provided, the corresponding questions in the interactive installation are skipped automatically.

It is possible to run the installer in batch mode, by passing the *-b* switch. The batch mode requires that you have accepted a SLA, and will refuse to run otherwise.

The installation script creates a log file called `install.log` in the current directory. In the case something goes wrong with the installation, this log file can be used to examine what happened to help understand what went wrong.

The rest of this section describes the command line arguments that can be given to `install-simics.pl`.

Usage

```
     install-simics.pl
(or) install-simics.pl [OPTIONS] <package1> [<key1>] <package2> [<key2>] ...
```

Arguments

<packageN>
Filename of a package to install.

<keyN>
Key corresponding to <packageN>. The key can be omitted if it is already cached or if the package is not encrypted.

Options

-a, --autoselect
If Simics and Simics add-on packages are installed at the same time, the script will automatically configure the new Simics installation to use these add-on packages.

-b, --batch
The script will act according to the command-line arguments without asking any questions.

-h, --help
Display this help text.

-p <path>, --prefix <path>
Specify the directory in which to install the packages (defaults to the last directory used, or `/opt/simics/simics-4.6/`).

-s <path>, --select-in <path>
Configure automatically the Simics installation in <path> to use the Simics add-on packages being installed.

-u <path>, --upgrade-from <path>
Re-use the configuration of an existing Simics installation located in <path>.

-v, --version
Print the version of the script and the major version of Simics that it can be used with.

--leave-tmp-files
Do not delete the temporary files created during installation.

7.6 Add-on Manager Options

The Simics Add-on Manager is provided with your Simics installation. You will find it at *[simics]*`/bin/addon-manager`, or at *[workspace]*`/bin/addon-manager`. The addon-manager script found in the workspace will associate packages with the workspace, while the script in the base package will associate packages with the main Simics installation.

By default, a workspace inherits the package associations of the main installation. If you want to make package associations locally in a workspace, you must first run addon-manager once with the -C or *-c* switch, in order to override this default.

In all cases, the `addon-manager` script creates a log file called `addon-manager.log` in the current directory. This log file can, if necessary, be used to debug and understand any errors that occurred.

It is possible to run the addon-manager in batch mode, by passing the *-b* switch.

The rest of this section describes the command line arguments that can be given to `addon-manager`.

Usage

```
addon-manager [options] [-u <path>] [-s <path>] [-d <path>]
```

Options

-b, --batch
The script will act according to the command-line arguments without asking any questions.

-c, --create-empty
Create an empty package list in the workspace. This will cause the workspace to ignore any package associations of the main Simics installation. Only valid when operating on a workspace or together with the -f parameter.

-C, --copy-from-simics-base
Create a new package list in a workspace, by copying the package list from the main Simics installation. Only valid when operating on a workspace or together with the -f parameter. Roughly equivalent to -u <simics_dir>

-d, --deselect <path>
Remove the package in <path> from the add-on package list. Note that listed packages that do not exist anymore will be removed automatically whenever this script runs.

-f, --package-list
Specify the file containing the package list. This is an alternative to the default (storing the list in the main Simics installation) or running addon-manager in a workspace.

-s, --select <path>
Add the package in <path> to the add-on package list.

-u, --upgrade <path>
Re-use the add-on package list found as <file> or present in <path>. The current list is backed-up as with a .backup extension.

-v, --version
Print the version of the script and the major version of Simics that it can be used with.

-w, --workspace <path>
Handle the package associations of the workspace <path> instead of the base package. This can also be done by running the addon-manager script located in <path>/bin.

Chapter 8

Installing the Simics Eclipse Tools

The Simics Eclipse Tools can be added to any Eclipse installation conforming to the requirements in section 8.1. If you already have Eclipse installed, e.g. from a Linux distribution, you can also use that version, assuming it meets the mentioned requirements. The recommended way of using the Simics Eclipse Tools is to download Eclipse from `http://www.eclipse.org/`. It is also possible to install the plugins using Wind River Workbench.

Some debugging features in Eclipse with Simics require Eclipse 3.7 (Indigo).

Once either of these environments for the Simics Eclipse Plugin Tools has been installed, you can proceed to install the Simics Eclipse Tools as described in section 8.3.

8.1 Requirements

Simics Eclipse plugins exist for the Eclipse Platform version 3.7 and 3.6. The following table lists the Eclipse features and versions which are required by the Simics Eclipse Tools.

Product	**Indigo**	**Helios**
Eclipse Platform	3.7.x	3.6.1 or newer
C/C++ Development Tools (CDT)	7.0.2 or newer	7.0.2 or newer

The Simics Eclipse Tools also requires Eclipse to be launched with Java Runtime version 5 or 6.

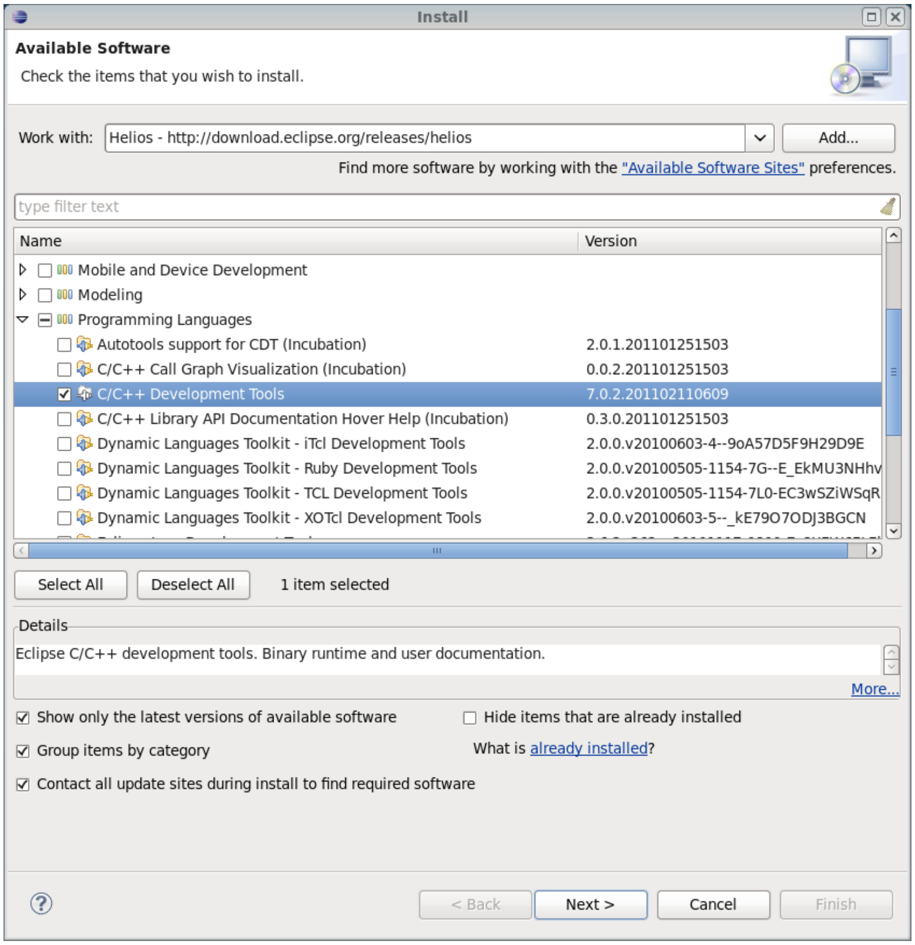
Note: If you want to use Java 7 on Windows you need to work around a bug in Java (`http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=7179799`). You need to pass `-vmargs -Djava.net.preferIPv4Stack=true` as command line arguments to the `eclipse` binary.

8.2 Install the Eclipse Platform from Eclipse.org

Go to `http://www.eclipse.org/downloads/packages/release/indigo/r` and download the package called “Eclipse IDE for C/C++ Developers” (107 MB). To install it, just extract the archive. This package includes the Eclipse Platform and CDT.

If you rather installed Eclipse Classic you must install the C/C++ Development Tools. Launch Eclipse, and select **Help** → **Install New Software. . .** . At the following dialog, select the Eclipse releases site as is shown in picture 8.1.

Wait while Eclipse synchronizes with the site and populates the list of available items. Find and select "Eclipse C/C++ Development Tools", as shown in the picture 8.1, and click Next and Next. Accept the license agreement and Finish. Restart Eclipse when prompted.



Work with: Helios - http://download.eclipse.org/releases/helios

Find more software by working with the "Available Software Sites" preferences.

type filter text

Name	Version
▷ □ Mobile and Device Development	
▷ □ Modeling	
▽ ⊟ Programming Languages	
□ Autotools support for CDT (Incubation)	2.0.1.201101251503
□ C/C++ Call Graph Visualization (Incubation)	0.0.2.201101251503
☑ C/C++ Development Tools	7.0.2.201102110609
□ C/C++ Library API Documentation Hover Help (Incubation)	0.3.0.201101251503
□ Dynamic Languages Toolkit - iTcl Development Tools	2.0.0.v20100603-4--9oA57D5F9H29D9E
□ Dynamic Languages Toolkit - Ruby Development Tools	2.0.0.v20100505-1154-7G--E_EkMU3NHhv
□ Dynamic Languages Toolkit - TCL Development Tools	2.0.0.v20100505-1154-7L0-EC3wSZiWSqR
□ Dynamic Languages Toolkit - XOTcl Development Tools	2.0.0.v20100603-5--_kE79O7ODJ3BGCN

1 item selected

Details

Eclipse C/C++ development tools. Binary runtime and user documentation.

☑ Show only the latest versions of available software

☐ Hide items that are already installed

☑ Group items by category

What is already installed?

☑ Contact all update sites during install to find required software

Figure 8.1: Select the Eclipse C/C++ Development Tools

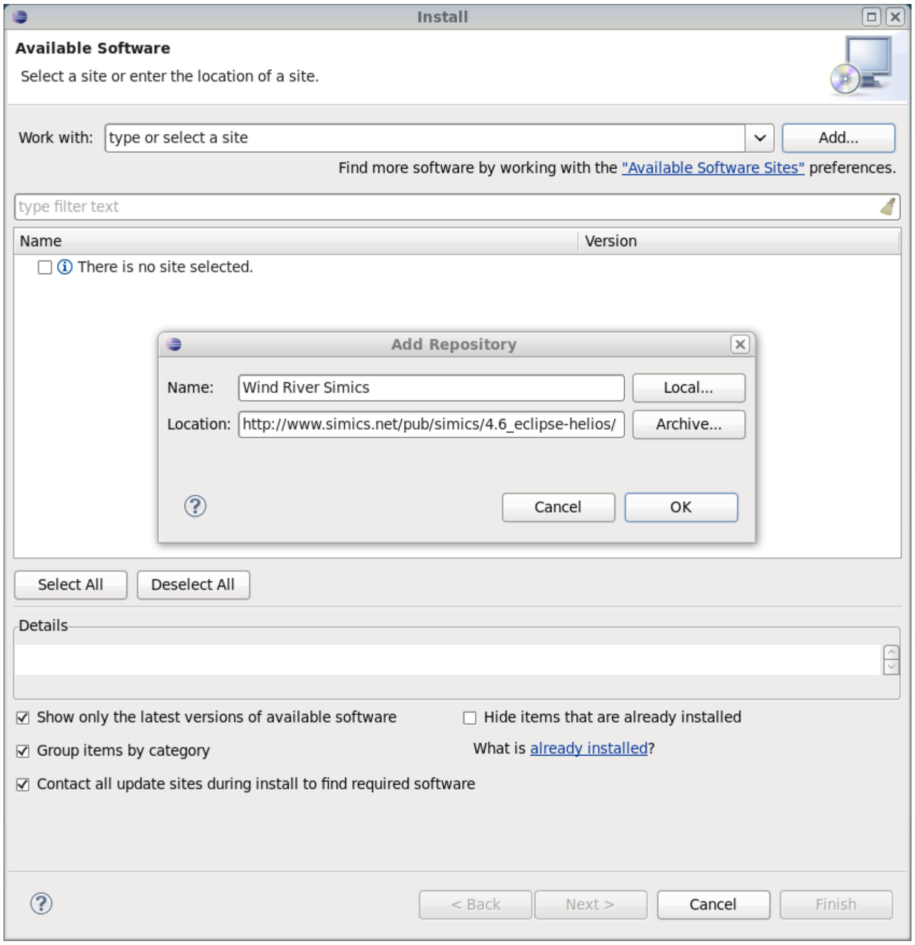
You are now ready to install the Simics Eclipse Tools, see section 8.3.

8.3 Install the Simics Eclipse Tools

Once you have installed Eclipse using any of the alternatives described in the previous sections you can install the Simics Eclipse Tools. The following instructions describe Eclipse 3.6 (Helios) but the procedure is similar for any Eclipse version.

Launch Eclipse and select **Help** → **Install New Software...**. In the following dialog, press the button **Add...** to add a new site. Enter the following information, as is shown in picture 8.2:

Name:	Wind River Simics
URL (3.7, Indigo):	`http://www.simics.net/pub/simics/4.6_eclipse-indigo/`
URL (3.6, Helios):	`http://www.simics.net/pub/simics/4.6_eclipse-helios/`



Available Software

Select a site or enter the location of a site.

Work with: type or select a site

Find more software by working with the "Available Software Sites" preferences.

type filter text

Name	Version
□ There is no site selected.	

Name: Wind River Simics

Location: http://www.simics.net/pub/simics/4.6_eclipse-helios/

☑ Show only the latest versions of available software

□ Hide items that are already installed

☑ Group items by category

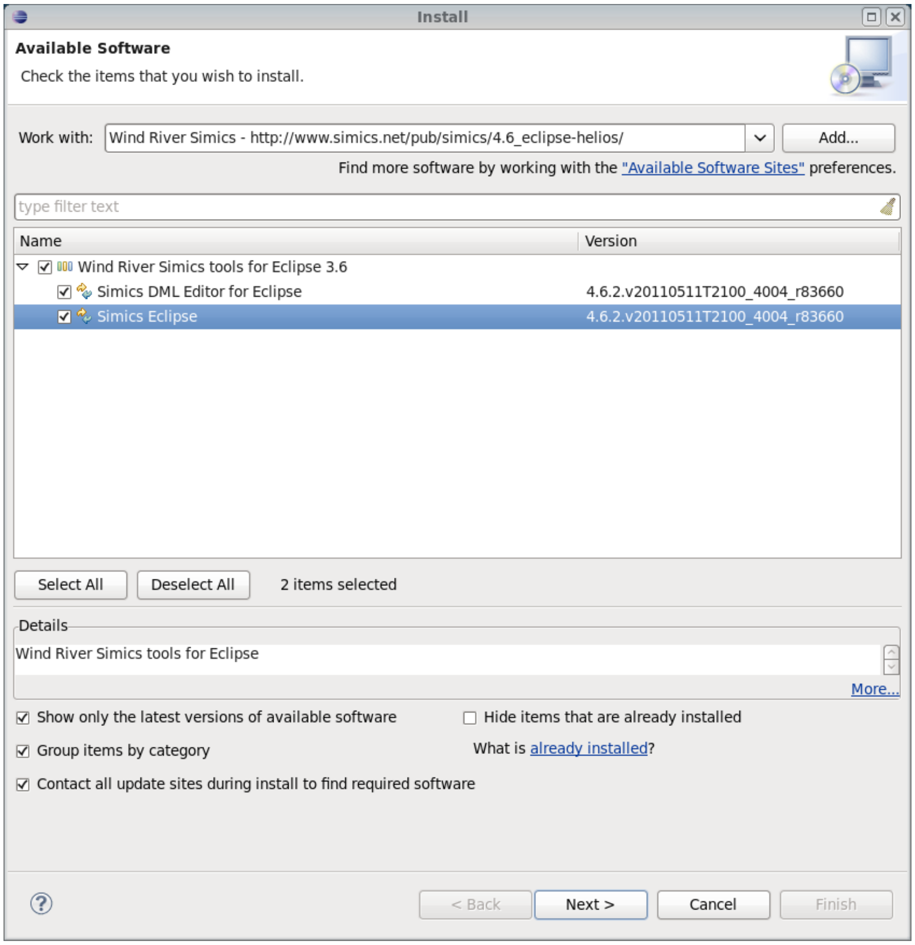
What is already installed?

☑ Contact all update sites during install to find required software

Figure 8.2: Add the Wind River Simics Update site

Once loaded, the Wind River Simics update site should look something like picture 8.3. The Simics Eclipse Tools are separated into two different Eclipse features.

- **Simics Eclipse:** This feature contains the base functionality for communicating with Simics, launch and configuration tools, as well as advanced features for analyzing, debugging, profiling, and more, and also the Simics command line view. This feature is a prerequisite for all the other features.



Available Software

Check the items that you wish to install.

Work with: Wind River Simics - http://www.simics.net/pub/simics/4.6_eclipse-helios/

Find more software by working with the "Available Software Sites" preferences.

type filter text

Name	Version
☑ Wind River Simics tools for Eclipse 3.6	
☑ Simics DML Editor for Eclipse	4.6.2.v20110511T2100_4004_r83660
☑ Simics Eclipse	4.6.2.v20110511T2100_4004_r83660

2 items selected

Details

Wind River Simics tools for Eclipse

☑ Show only the latest versions of available software

☐ Hide items that are already installed

☑ Group items by category

What is already installed?

☑ Contact all update sites during install to find required software

Figure 8.3: Select the Simics features to install

- **Wind River DML Editor for Eclipse:** This feature provides an editor for the DML modeling language. It provides a built in parser and is able to detect syntax errors while typing, without having to run the DML compiler. It also integrates with CDT/Make in order to build the DML devices. The DML Editor requires the *Model Builder* Simics package and license in order to be used.

Select all the features you need and click "Next" to proceed, and Next. Agree to the license agreement, click Finish, and wait for Eclipse to download and install the Simics features. Finally, once the installation is complete, restart Eclipse.

Note: Midway you may be prompted by a security warning and asked to agree to install the feature even though it has not been digitally signed by Eclipse.org. Just click OK to indicate that you want to proceed with the installation. The reason the features are not signed is for compatibility with OpenJDK.

For more information on how to use the Simics Eclipse Tools, please refer to the *Simics Eclipse User's Guide*, which is available from within Eclipse, via the menu **Help** → **Help Contents**.

8.4 Associate Eclipse with a Simics Base installation

It is required to associate an Eclipse-Simics project with an installation of Simics Base for the Simics Eclipse Tools to function properly. This is also required when you install a newer version of the Simics Base package, or if you want to switch between different versions of Simics.

The Simics Eclipse Tools tries to find a Simics installation, in Windows by querying the registry for Simics and at Linux by querying the /opt/ file path, and it suggests a version. Otherwise, to specify the location of the Simics Base package, open the Preferences dialog via **Window** → **Preferences**. Select the Simics section and add or change the path to refer to the folder for the preferred Simics Base installation.

It is possible to have several Simics projects, each associated with a different Simics installation. It is also possible to re-associate a project with another Simics version, which you probably want to do after installing a new Simics version. Selecting a Simics installation is done via the properties for the project, either menu **Project** → **Properties** → **Simics**, or right-click the project and select **Properties** → **Simics**, find and select the wanted Simics version.

Eclipse will use Simics add-on packages that are associated with the chosen Simics installation. This *is not* handled by Eclipse but with the Simics **addon-manager**, see the "Installing More Packages" section, or the "Advanced Installation Options" chapter.

Note: If this is a new installation of Eclipse you are presented with the screen *Welcome to Eclipse*. To find the Simics Eclipse Tools, close this screen and open a Simics perspective at **Window** → **Open Perspective** → **Other...** → **Simics**.

Chapter 9

License Installation

Simics requires a license server and an installed license. The very first time Simics is started, it will search for license files in the directory *[simics]*`/licenses/`. If you specified a license file when Simics was installed, the file was copied to the *[simics]*`/licenses/` directory, Simics will find it and no further action is required. If you skipped the installation step where a license file is specified, you will have to specify a license file now. Specifying license file is done in GUI mode.

Once a valid license file has been specified, it will be saved in your Simics preferences for future use. If a local license server is needed, one will be started as needed.

9.1 FLEXnet Node Locked Licenses

Simics node-locked licenses may require a license server to run on the same computer as you are running Simics on. There is no need to manually configure the license server as this is handled by Simics. When Simics prompts you for a license file when started the first time, do NOT specify a license server, instead specify that you will use a license file and browse to the node-locked license file. Simics will detect if the license file requires a license server to run and start it automatically.

9.2 FLEXnet Floating Licenses

The license server used is *FLEXnet Publisher* (formerly called *FLEXlm*) from Flexera. Simics is built for FLEXnet version 11.7 but newer versions of the license server will also work. This chapter briefly describe how to set up the server. For further information consult *FLEXnet Licensing — End User Guide* which can be found at `http://www.globes.com/support/fnp_utilities_download.htm` or at `http://www.simics.net` by clicking *download* and then *FLEXnet*. More information may also be available from `http://www.flexerasoftware.com`.

Note: FLEXnet is a 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.

In this chapter the '/' character is used to separate directories and files like Linux do. If you are using Windows you should read the character as '\' instead.

To manage the license server the following programs are needed:

- `lmgrd` — The License Manager Daemon (license server)
- `lmutil` — FLEXnet utility program(s)
- `simics` — The vendor daemon for Wind River Simics

The files are located in the Simics distribution under `[simics]/flexnet/`*host*`/bin` (e.g., `[simics]/flexnet/linux64/bin`) but can be moved to any location if desired.

On Windows host there is also the `lmtools` utility included that provides a graphical interface to `lmutils`.

Note: For some Linux distributions, if you face an error such as `[simics]/flexnet/<host>/bin/lmutil: Command not found` you need to install the `lsb-core` package.

Now you need your license file (e.g., `simics.lic`) from Wind River (usually sent via e-mail), that specifies your license. Copy the license file to the `licenses` directory in the Simics distribution (see section 9.3 if you want to keep the license file somewhere else). Open the file in a text editor and locate the line that starts with `SERVER`:

```
SERVER host hostid
```

This line specifies on which host you can run the license server. If "`hostid`" is `ANY` you can run the license server on any host you like. Change the "host" to the name of that computer. This can be a global name as in `server.company.com`, or just the host name. Skip this step if `host` is already preset and set to one of your computers.

Now locate the `VENDOR` line:

```
VENDOR simics /path_to_daemon/simics
```

Change the path to the location of the `simics` vendor daemon.

You can now start the license server. If the name of the license file is `simics.lic` and lmgrd is in your path, run:

```
user@host:~$ lmgrd -c simics.lic
```

Remember to run `lmgrd` on the `host` specified in the license file. Since `lmgrd` does not need root privileges it is recommended to run it as a non-privileged user.

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

If you have several license files from different vendors you can list them all when you start the server. Only one server is necessary.

Once the license server is started, 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 ask the license server for licenses and if available licenses exist, Simics will *check-out* the corresponding features and run. On exit, Simics will give back the features to the server (*check-in*). Certain features may linger some time before they are returned to the server. This depends on your type of license.

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

To shut down the license server use the `lmutil` program:

```
user@host:~$ lmutil lmdown
```

On Windows host the graphical frontend `lmtools` can be used instead.

Note: Simics will not update the `.flexlmrc` file in the user's home directory after a successful checkout, as some earlier versions did.

9.2.1 Serving licenses for both Simics 4.6/4.4/4.2 and 4.0

There are two ways to serve licenses for both Simics 4.6/4.4/4.2 and Simics 4.0.

- Running two server processes, one that serves 4.6, 4.4 and 4.2 licenses and one that serves 4.0 licenses.
- Running one server process that serves both 4.6/4.4/4.2 and 4.0 licenses.

In the first case, running two server processes, all you need to do is start one license server process after the other:

```
user@host:~$ lmgrd -c /path/to/4.0-license.lic
user@host:~$ lmgrd -c /path/to/4.6-license.lic
```

The license server processes will find an unused port in the default range 27000 - 27009. The first server process started will usually pick port 27000 and the second port 27001. If you want to control exactly which port a particular server uses, you can select it by editing the license file for that server. Look for a line similar to:

```
SERVER myserver.mydomain 123456789abc
```

and add the port you want to use to the end of that line, in this example 4711:

```
SERVER myserver.mydomain 123456789abc 4711
```

In the second case, running one server process, no special action needs to be taken, as the name of the vendor daemon differs between Simics 4.0 (vtech) and Simics 4.6, 4.4 and 4.2 (simics). Both 4.6/4.4/4.2 and 4.0 licenses can be served by the same license server process. All that needs to be done is to start the license server with both license files as arguments:

```
user@host:~$ lmgrd -c /path/to/4.0-license.lic:/path/to/4.6-license.lic
```

9.3 Overriding License File Location

When Simics is started Simics searches for a license file in the following order:

1. If the Simics command line option `-license-file` *file* is given this file is used.
2. If Simics has a saved preferences file and the option `license_file` is set to a file, this file is used. The preferences file, `prefs`, is located in the `~/.simics/4.6/` directory on Linux and in the `C:\Documents and Settings\`*User*`\Application Data\Simics\4.6` directory on Windows.
3. If the `SIMICS_LICENSE_FILE` environment variable is set to a license file, this file is used (the FLEXnet `LM_LICENSE_FILE` environment variable is always ignored by Simics).
4. If a file with the file extension `.lic` exists in the `license` directory in the Simics installation tree, this file is used.

If a license file is chosen using the `-license-file` option or by setting the `SIMICS_LICENSE_FILE` variable it will only be used for the current session. It is possible to tell Simics to use it for subsequent sessions as well by saving the current setting with the preferences:

```
simics> prefs->license_file = sim->license_file
simics> save-preferences
```

A directory can be specified instead of a license file. In this case Simics will search that directory for a valid license file. You can also use the notation `@host` to point to the host that is running the license server.

If you want to override the location of the license file for all Simics users that use an installation, you can place a pointer to the license server in the license directory by creating a file there with the `.lic` extension (e.g., `server.lic`) that contains the following lines:

```
SERVER host ANY
USE_SERVER
```

where "host" is the name of the license server.

9.4 Borrowing Licenses

Simics supports license borrowing which means that you can check out licenses from the license server and save them temporarily on a computer that will be disconnected from the network and thus lose its connection to the license server. You may then still be able to work with Simics while you are off line. Your license file need to support borrowing for this to work.

To borrow a license you use the start-up option `-E` when you start Simics. A date description should follow that tells how long you intend to borrow the license. This description has one of the formats `DD-MMM-YYYY[:hh:mm]`, `+Nh`, or `+Nd`. The first format specifies the time when the borrowed license will expire. For example,

```
$ simics -E 24-dec-2008:15:00
```

where the time `:15:00` is optional. The default time is `00:00`. The second and third format specifics how many hours or days from present the license is borrowed, e.g.,

```
$ simics -E +7d
```

will borrow a license a week from now and

```
$ simics -E +24h
```

will borrow a license 24 hours from now.

When you borrow a license that license will not be available to other users until the borrow time expires. There is currently no way to return a borrowed license earlier.

Note: To be able to use all features of Simics off line you must make sure to access them when you run Simics with the `-E` option since Simics caches your license locally for each Simics Product used. A good way to assure this is to do everything you wish to do off line when you have specified the `-E` option to Simics. Most notable is to run at least one instruction since this will make sure you cache the licenses of the different CPU modules used.

9.5 Troubleshooting License Problems

This section contains the steps you need to take if you still have license problems after having configured Simics to use your license file.

9.5.1 Are you using the correct license file?

The most common licensing problem is that the wrong license file is used. Section 9.3 contains the details about where Simics looks for a license file. Use the `-license-file` command line option to be absolutely certain about which license file you are using. If you know the name of the computer that is supposed to be the license server, you can use this option to specify it with a command line like the following:

```
joe@computer$ simics -license-file [port]@[servername]
```

[port] can be omitted if the license server is using a default port:

```
joe@computer$ simics -license-file @[servername]
```

The license file contains a line that identifies the license server to use. The line starts with the text SERVER and looks something like this:

```
SERVER mylicense.server 001831f6c749
```

You are allowed to edit the server name (mylicense.server in this example) in the license file to suit the actual name of the license server.

9.5.2 Does the license file match the license server computer?

The license file contains a line that identifies the license server to use. The line starts with the text SERVER and looks something like this:

```
SERVER mylicense.server 001831f6c749
```

The 12 hexadecimal digits after the host name is the *host id* of the license server. The host id of a computer can be obtained by running the following command on that computer:

```
joe@computer$ [simics]/flexnet/[hosttype]/bin/lmutil lmhostid
[...]
The FLEXnet host ID of this machine is "001831f6c749"
```

[simics] is the path to your Simics-Base installation and *[hosttype]* is the host type of the computer you are using, for example "linux32". As can be seen in the output, the host id matches the SERVER line in the license file. That means that is is possible to start a license server using this license file on this computer.

Editing the host id field in the license file makes the entire license file invalid. If you have a license file where the host id on the SERVER line does not match the host id of the computer that you want to use as license server, you have to contact Wind River support and ask for a new license file with the correct host id.

If you have a node locked or evaluation license, an incorrect host id in the license file will inhibit the automatic start of a local license server. Simics will instead try to contact the computer whose name is in the SERVER line to retrieve a license. You will most likely be faced with an error message saying that the license server cannot be contacted. In that case, you will need to contact Wind River support to get a license that is appropriate for the computer you want to run Simics on.

9.5.3 The license server cannot be reached.

If you have verified that you are using the correct license file or license server host, but get errors saying that the license server cannot be contacted, the most probable explanation is that the license server is not running on that host.

To establish that you are able to communicate with the host for the license server, you should start by pinging it:

```
joe@computer$ ping [mylicense.server]
```

```
PING mylicense.server (192.168.47.11) 56(84) bytes of data.
64 bytes from 192.168.47.11: icmp_req=1 ttl=64 time=306 ms
[...]
```

If ping fails, you will have to contact your system administrator. You probably have a networking problem, something that Wind River cannot fix.

If all is well and the host for the license server responds to ping, the next step is to establish that a license server is really running on that host. The relevant lines to look for is given in the example:

```
joe@computer$ [simics]/flexnet/[hosttype]/bin/lmutil lmstat -c [port]@[mylicense.server]
[...]
mylicense.server: license server UP (MASTER) v11.7
[...]
Vendor daemon status (on mylicense.server):

    simics: UP v11.7
[...]
```

As in the previous example, *[port]* can be omitted if the license server runs on a standard port. Instead of specifying the server name directly on the command line, you can use the *-c* option to specify your license file as well.

If you get an error message instead of the above output, the license server is not running on the license server host. If you have a node-locked or evaluation license and run a local license server, the license server probably did not start correctly, and you should have received an error message saying what the problem was. If you did not get an error message, it is likely that the license file you are using does not match your computer. Take a look at the above trouble shooting step. If a local license server was indeed started, you can always look at the log file generated by it. It can be found in `~/.simics/`*version*`/license-server.log` on Linux and `C:\Documents and Settings\`*User*`\Application Data\Simics\4.6\license-server.log` on Windows.

If you do not have a node-locked or evaluation license, but a normal floating license, you need contact your system administrator to get the license server running on the license server host.

If the required output is missing, the license server is running, but it is not configured to serve Simics licenses. You need contact your system administrator to configure the license server to serve Simics licenses. This cannot happen if you have a node-locked or evaluation license and run the license server on your own computer.

9.5.4 Feature cannot be checked out.

Simics require *license features* for different tasks. The above example shows the feature “simics”, which is required for Simics to even start, and “simics_model_builder”, required to compile DML modules. Other features are “simics_hindsight”, required to use the Simics command line, and “ppce600”, required to run simulated machines with the PowerPC e600 processor. The feature that Simics is trying to acquire from the license server is printed in the error message that is displayed if the acquisition fails.

If the license server indeed can be contacted but you are still not able to run Simics, the problem is likely to be that the someone else is using all of the available license features that you need, or that the license server does not support the feature you need at all. You can check both cases with the `lmutil lmstat` command like in the previous example, only this time the *-a* option is added:

```
joe@computer$ [simics]/flexnet/[hosttype]/bin/lmutil lmstat -a -c [port]@[mylicense.server]
[...]
Users of simics:  (Total of 15 licenses issued;  Total of 1 license in use)

  "simics" v4.6, vendor: simics
[...]
Users of simics_model_builder:  (Total of 1000 licenses issued;  Total of 0 licenses in use)
[...]
```

The output shows that there are 15 simics license features in total, of which 1 is currently used.

Chapter 10

Workspace Management

A Simics installation is meant to be used by multiple users at the same time. It is also designed to make it easy to keep several versions of the same package (whether Simics or an add-on product) on the same machine, to handle different projects at a different stage of maturity.

To achieve these goals, each user should work in *workspaces* rather than in the installation directory itself. A workspace is simply a directory in which Simics is allowed to store a number of book-keeping files, while a user can create and modify scripts and setups to his leisure.

Workspaces are also used for creating new device models. This is described in more details in the *Model Builder User's Guide* provided in the *Model Builder* add-on product.

Creating a Workspace

The first time you start the Simics GUI, it will suggest that you create a workspace. It is possible to create a new workspace or change the current workspace with the **Create Workspace...** and **Change Workspace...** menu items under the **File** menu.

You can also create a workspace on the command line using the `workspace-setup` program (`workspace-setup.bat` on Windows) that is controlling all workspace management:

Unix

```
[simics]/bin/workspace-setup ~/my-workspace
```

Windows

```
[simics]\bin\workspace-setup.bat ↩
"C:\Documents And Settings\Joe\Simics-Workspace"
```

In both cases, a workspace will be created, containing a number of files described later in this section.

Updating a Workspace

Every time your start the GUI, or whenever you run `workspace-setup` in your workspace, it will check for changes in your Simics installation (for example, new or updated add-on packages). If it finds any changes, it will update the corresponding files. Note that `workspace-setup` will *never* overwrite one of your files or a file you modified.When you have installed a new Simics base package, you need to explicitly tell the workspace to switch to the new version. To do this, run the new base package's workspace-setup script in your workspace directory, just as when creating a new workspace.

The files created in a workspace depends on which add-on packages are installed, but the most important are listed here. Files related to modeling are described in the *Model Builder User's Guide*.

`simics`
Starts Simics in command line mode.

`simics-gui`
Starts Simics in GUI mode.

`bin` directory
Contains scripts to start various other tools, such as `addon-manager` and `workspace-setup`.

`doc` directory
On Linux, contains links to documentation. On Windows the documentation can be found in the Start Menu instead.

`targets` directory
Contains some pre-configured machines for the system add-on packages installed.

The following sections describe how to run workspace-setup. Options related to modeling are described in the *Model Builder User's Guide*.

Usage

`workspace-setup` [*options*] [*workspace*]

Options

`-h, --help`
Show help message and exit.

`-v, --version`
Prints information about Simics (version, installation directory).

`-q, --quiet`
Cancel the verbose flag.

`-V, --verbose`
Print more information about the actions taken by the script.

`-n, --dry-run`
Execute normally, but do not change or create any files.

`--ignore-existing-files`
Prevent the script from warning that the future workspace directory is not empty.

`--ignore-cygwin-warning`
(Windows-only) Prevent the script from checking whether the workspace directory looks like a Cygwin path or not.

`--force`
Force the workspace-setup script to create or update a workspace. Files that would be overwritten will be backed up automatically.

`--check-workspace-version`
Check the version of the workspace, and return 1 if it needs to be created or updated, 0 otherwise.

Chapter 11

Contact Technical Support

Wind River's support website for Simics is located at `http://www.simics.net/support`. The *Simics Support* application note provides guidelines regarding technical support.

For each commercial customer, a unique and private support forum is established that the customer can use to ask for product advice, post questions, and report problems. Wind River's engineering staff, product specialists, and field application engineers monitor this forum on a daily basis to provide prompt replies to customer inquiries. All commercial forums are username and password protected. The field application engineer assigned to your company will provide further details on how to log in and use the support forum.

For license related issues contact `license@windriver.com`.

For academic users, Wind River maintains a public, shared forum that is available for all academic users of Simics. This forum is located at `http://www.simics.net`.

Chapter 12

Overview of the Simics Directory Structure

The Simics packages, regardless of what components are included (source code or not, etc), are intended to have the following structure. Note that some of these files or directories may not be included in all packages: **[base]** marks a file that will only be present in the Simics base package; **[builder]** marks a file present in the *Model Builder* package.

Directory/File	Description
`Version`	**[base]** version and timestamp
`DOCUMENTATION(.html)`	**[base]** documentation available.
`RELEASENOTES(.html)`	**[base]** description of all the changes.
`README.previous`	README files for previous Simics versions
`flexnet/`*host*`/bin`	**[base]** FLEXnet binaries
`licenses`	**[base]** FLEXnet license files goes here
`bin`	Various utilities
`scripts`	Various tools and scripts needed for compilation
`src`	Main source code directory
`src/devices`	Main device module directory
`src/devices/...`	One directory for each device module
`src/extensions`	Main extension module directory
`src/extensions/...`	One directory for each extension or component module
`src/include`	**[builder]** include files for Simics API
`src/misc`	**[builder]** source code for external programs
`config`	**[builder]** generic configuration files for make
`config/workspace`	**[builder]** workspace makefiles
`doc`	Simics documentation
`doc/*.pdf`	PDF documentation
`doc/*.htb`	Online documentation
host	Directory with files for named *host* type

host/bin	Simics binaries
host/lib	Simics loadable modules
host/lib/python	Simics Python files
host/lib/python-*x.y*	**[base]** Standard Python files
host/sys/lib	External libraries used by Simics
targets/*	Virtual machine configurations

A Simics workspace is intended to have the following structure.

Directory/File	**Description**
simics	Shortcut to start Simics in command line mode
simics-gui	Shortcut to start Simics in GUI mode
simics-eclipse	Shortcut to start Simics with the Eclipse frontend
GNUmakefile	Makefile to build all modules in this workspace
config.mk	Default definitions of make variables
compiler.mk	Default definitions for the compiler
bin	Scripts to call other tools provided with Simics
doc	Links to documentation (on Unix only).
modules	Source code of user-developed modules
targets/*	Ready-to-run target machine configurations
host	Directory with user files for named *host* type
host/lib	Simics loadable user-developed modules
host/lib/python	User-developed Simics Python files.
host/obj	scratch compilation area
.workspace-properties	For internal bookkeeping