



# Simics/Serengeti Target Guide

*Simics Version* 3.0

*Revision* 1406  
*Date* 2008-02-20

*VIRTUTECH CONFIDENTIAL*

© 1998–2006 Virtutech AB  
Drottningholmsv. 14, SE-112 42 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.

The proprietary information contained within this Publication must not be disclosed to others without the written consent of Virtutech.

# Contents

<b>1</b>	<b>About Simics Documentation</b>	<b>6</b>
1.1	Conventions . . . . .	6
1.2	Simics Guides and Manuals . . . . .	6
	Simics Installation Guide for Unix and for Windows . . . . .	6
	Simics User Guide for Unix and for Windows . . . . .	7
	Simics Eclipse User Guide . . . . .	7
	Simics Target Guides . . . . .	7
	Simics Programming Guide . . . . .	7
	DML Tutorial . . . . .	7
	DML Reference Manual . . . . .	7
	Simics Reference Manual . . . . .	7
	Simics Micro-Architectural Interface . . . . .	7
	RELEASENOTES and LIMITATIONS files . . . . .	8
	Simics Technical FAQ . . . . .	8
	Simics Support Forum . . . . .	8
	Other Interesting Documents . . . . .	8
<b>2</b>	<b>Simics/Serengeti Overview</b>	<b>9</b>
2.1	Introduction . . . . .	9
2.2	Supported Hardware . . . . .	9
<b>3</b>	<b>Simulated Machines</b>	<b>11</b>
3.1	Abisko . . . . .	11
	3.1.1 Abisko Scripts . . . . .	11
3.2	Sarek . . . . .	12
	3.2.1 Sarek Scripts . . . . .	12
3.3	Parameters for Machine Scripts . . . . .	13
	3.3.1 abisko-common and sarek-common . . . . .	13
<b>4</b>	<b>Supported Components</b>	<b>15</b>
4.1	Serengeti Components . . . . .	15
	4.1.1 serengeti-3800-chassis . . . . .	15
	4.1.2 serengeti-4800-chassis . . . . .	17
	4.1.3 serengeti-4810-chassis . . . . .	18
	4.1.4 serengeti-6800-chassis . . . . .	20

4.1.5	serengeti-cluster-chassis	21
4.1.6	serengeti-us-iii-cpu-board	24
4.1.7	serengeti-us-iii-plus-cpu-board	25
4.1.8	serengeti-us-iv-cpu-board	26
4.1.9	serengeti-us-iv-plus-cpu-board	27
4.1.10	serengeti-pci8-board	28
4.1.11	serengeti-cpci4-board	28
4.1.12	serengeti-sp-cpci6-board	29
4.2	PCI Device Components	30
4.2.1	sun-cpci-hme-isp	30
4.2.2	sun-cpci-isp-isp	31
4.2.3	sun-cpci-qlc-qlc	32
4.2.4	sun-pci-ce	32
4.2.5	sun-pci-hme	33
4.2.6	sun-pci-hme-isp	34
4.2.7	sun-pci-pgx64	35
4.2.8	sun-pci-qlc	35
4.2.9	sun-pci-qlc-qlc	36
4.2.10	pci-bcm5703c	37
4.2.11	pci-bcm5704c	38
4.2.12	pci-sym53c875	39
4.2.13	pci-sym53c876	39
4.3	Standard Components	40
4.3.1	std-ethernet-link	40
4.3.2	std-service-node	41
4.3.3	std-scsi-bus	42
4.3.4	std-scsi-disk	42
4.3.5	std-scsi-cdrom	43
4.3.6	simple-fc-disk	44
4.3.7	std-text-console	45
4.3.8	std-server-console	46
4.3.9	std-graphics-console	47
4.3.10	std-text-graphics-console	48
4.4	Timing Components	49
4.4.1	sample-gcache	49
4.4.2	sample-ma-model	49
4.4.3	sample-ooo-model	50
4.5	Base Components	51
4.5.1	component	51
4.5.2	top-component	52
<b>5</b>	<b>Examples</b>	<b>53</b>
5.1	Adding a new Disk to a Serengeti Machine	53

<b>6</b>	<b>Installing an OS on Simics</b>	<b>56</b>
6.1	Installing Solaris on Simics . . . . .	56
6.1.1	Installation, step by step . . . . .	56
<b>7</b>	<b>Miscellaneous Notes</b>	<b>58</b>
7.1	Notes on Solaris for Serengeti . . . . .	58
7.2	Multiple Network Devices . . . . .	58
7.3	Changing the Processor Clock Frequency . . . . .	59
<b>8</b>	<b>Limitations</b>	<b>60</b>
8.1	Limitations of the Simulated Model . . . . .	60
8.2	Other Limitations . . . . .	60
	<b>Index</b>	<b>61</b>

## 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

# Simics/Serengeti Overview

### 2.1 Introduction

Simics/Serengeti models the *Sun Fire 3800 - 6800* class of servers. A Serengeti server can be configured with up to 24 UltraSPARC III, UltraSPARC III Cu (III+), UltraSPARC IV, or UltraSPARC IV+ processors and 192GB of memory. A variety of PCI-bus based devices are supported. Only Solaris is supported as target Operating System.

Virtutech does not provide any disk images with Solaris for Serengeti, due to licensing issues. However, scripts are included for installing Solaris 8, 9 or 10 on the virtual machine. Installing Solaris is described in chapter [6.1](#).

### 2.2 Supported Hardware

Serengeti servers can have up to 10 board-slots depending on the model. Simics models the largest machine, Sun Fire 6800, by default, but this is configurable. The first 6 slots can hold CPU boards, and the last 4 can hold I/O boards. There are three kinds of PCI boards, the kind that can be used depends on the server model. For example, the 3800 server only supports the (SP) cPCI board.

#### Chassis

Sun Fire 3800	SP	2 CPU boards + 2 I/O boards
Sun Fire 4800	MD	3 CPU boards + 2 I/O boards
Sun Fire 4810	ME	3 CPU boards + 2 I/O boards
Sun Fire 6800	DS	6 CPU boards + 4 I/O boards

#### CPU Boards

0 - 4 UltraSPARC-III	max 8 GB of memory per CPU, 32GB total
0 - 4 UltraSPARC-III Cu	max 8 GB of memory per CPU, 32GB total
0 - 4 UltraSPARC-IV	max 8 GB of memory per CPU, 32GB total
0 - 4 UltraSPARC-IV+	max 8 GB of memory per CPU, 32GB total

**I/O Boards**

8 slot PCI board

4-slot Compact PCI board

6-slot Compact PCI board    SP Chassis only

**Supported Devices**

'ce'	Gb Ethernet controller	(Cassini+)
'bge'	Gb Ethernet controller	(BCM5703C)
'bge'	Dual Gb Ethernet controller	(BCM5704C)
'hme'	Ethernet controller	(Cheerio)
'glm'	SCSI controller	(SYM53C875)
'isp'	SCSI controller	(ISP1040)
'pgx64'	24-Bit Frame Buffer	(pgx64)
'qlc'	Fibre-Channel controller	(ISP2200)
	PCI-to-PCI bridge	(i21152)
	PCI-to-PCI bridge	(i21554)

A good guide to the Sun Fire servers and what boards and devices that are supported can be found in the "Sun System Handbook", available online at: [http://sunsolve.sun.com/handbook\\_pub/](http://sunsolve.sun.com/handbook_pub/)

**Note:**

The Simics/Serengeti model does not currently support a keyboard or mouse, and thus the pgx64 card cannot be used to create an interactive X session. It can be used as a passive display by starting an X server in no mouse and no keyboard mode.

## Chapter 3

# Simulated Machines

Simics scripts for starting Serengeti machines are located in the `[workspace]/targets/serengeti/` directory, while the actual configuration scripts can be found in `[simics]/targets/serengeti/`.

### 3.1 Abisko

Abisko is a Sun Fire 6800 server with a single UltraSPARC-III Cu processor running at 75 MHz, and 256 MB of memory. It has one Ethernet adapter, one SCSI disk and one SCSI CD-ROM. The default configuration can be modified as described in section 3.3. An operating system must be installed on abisko before it can be used.

#### 3.1.1 Abisko Scripts

**`abisko-common.simics`**

Starts the Abisko machine with the default configuration.

**`abisko-gcache-common.simics`**

Default Abisko machine with a *g-cache* cache model connected.

**`abisko-ma-common.simics`**

Default Abisko machine with a simple processor timing model connected. Simics must be started in MAI mode (-ma) to run this script.

**`abisko-ooo-common.simics`**

Default Abisko machine with a simple out-of-order timing model connected. Simics must be started in MAI mode (-ma) to run this script.

**`abisko-multi.simics`**

Example script with two Abisko machines in the same session, connected by an Ethernet link.

**`abisko-sol<version>-cd-install11.simics`**

Script for installing Solaris on the simulated machine, phase 1. <version> is one of 8, 9 and 10.

**abisko-sol<version>-cd-install2.simics**

Script for installing Solaris on the simulated machine, phase 2. <version> is one of 8, 9 and 10.

**abisko-sol<version>-cd-install3.simics**

Script for installing Solaris on the simulated machine, phase 3. <version> is one of 8, 9 and 10.

## 3.2 Sarek

Sarek is a Sun Fire 6800 server with a single UltraSPARC-III Cu processor running at 75 MHz, and 256 MB of memory. It has one Ethernet adapter, one SCSI disk and one SCSI CD-ROM. The default configuration can be modified as described in section 3.3.

The Sarek machine is configured for existing Solaris 8, 9 or 10 disk dumps. The disk dumps are only available for commercial customer with a special license agreement with Sun. Some common GNU utilities are installed on the disk images, such as `bash`, `gcc`, `gmake` and `emacs`. The *SimicsFS* file-system is also included.

Additional information:

- Solaris 8 (7/01) and Solaris 9 (5/02) installed as “Developer System” directly on Simics.
- SimicsFS support.
- Login `root`, no password.
- Configured with static IP address 10.10.0.11, gw 10.10.0.1, when DHCP not used.

### 3.2.1 Sarek Scripts

**sarek-common.simics**

Starts the Sarek machine with the default configuration.

**sarek-dhcp-common.simics**

Similar to `sarek-common.simics`, but gets the host name and IP address from the DHCP server.

**sarek-gcache-common.simics**

Default Sarek machine with a *g-cache* cache model connected.

**sarek-ma-common.simics**

Default Sarek machine with a simple processor timing model connected. Simics must be started in MAI mode (`-ma`) to run this script.

**sarek-ooo-common.simics**

Default Sarek machine with a simple out-of-order timing model connected. Simics must be started in MAI mode (`-ma`) to run this script.

**sarek-multi.simics**

Example script with two Sarek machines in the same session, connected by an Ethernet link.

### 3.3 Parameters for Machine Scripts

The following parameters can be set before running the `abisko-common.simics`, or `sarek-common.simics` scripts. Other `.simics` scripts may set some of the parameters unconditionally, and do not allow the user to override them. For example, the `sarek-dhcp-common.simics` script will always set the `$create_network` variable to `yes`.

#### 3.3.1 abisko-common and sarek-common

**\$create\_network**

Set to `yes` if the script should create an Ethernet link and connect the primary Ethernet adapter to it.

**\$cpu\_class**

The type of processor to create. Should be one of `ultrasparc-iii`, `ultrasparc-iii-plus`, `ultrasparc-iv` and `ultrasparc-iv-plus`.

**\$disk\_size**

Size of the primary hard disk. This parameter must match any disk images that are added to the primary disk.

**\$do\_boot**

Set to `no` to stop at OBP prompt, without booting the OS.

**\$do\_login**

Set to `no` to prevent the script from logging in as root automatically when the operating system has reached the login prompt.

**\$eth\_link**

The Ethernet link to connect the primary Ethernet adapter to. This parameter should be set when a link already exist and the `$create_network` parameter is `no`.

**\$hostid**

The *hostid* for the simulated machine.

**\$freq\_mhz**

The clock frequency in MHz for all processors.

**\$host\_name**

The host name used by the DHCP and DNS servers for this machine This variable will not change the host name set for the machine on the disk dumps.

**\$ip\_address**

The IP address used by the DHCP and DNS servers for this machine This variable will not change any IP address set for the machine on the disk dumps.

**\$mac\_address**

MAC address of the primary Ethernet adapter.

**\$megs\_per\_cpu**

The amount of system memory, in MB, for each processor.

**\$num\_cpus**

The number of processors in the machine.

**\$os**

The operating system to boot, one of `solaris10`, `solaris9`, and `solaris8`. Requires that a matching disk dump exists.

**\$rtc\_time**

Date and time of the real-time clock at boot.

**\$service\_node**

The *service node* to use for DHCP and DNS. This parameter should be set when a service node already exist and the `$create_network` parameter is `no`.

## Chapter 4

# Supported Components

The following sections list components that are supported for the Serengeti architecture. There also exist other components in Simics, such as various PCI devices, that may work for Serengeti but that have not been tested.

The default machines are constructed from components in the `-system.include` files in `[simics]/targets/serengeti/`. See the Configuration and Checkpointing chapter in the Simics User Guide for information on how to define your own machine, or make modifications to an existing machine.

### 4.1 Serengeti Components

#### 4.1.1 serengeti-3800-chassis

##### Description

The “sunfire-3800-chassis” component represents the chassis, backplane and system-console of a Sun Fire 3800 server, with slots for up to two processor boards and two I/O boards. The system is sometimes called SP for Service Provider server.

##### Attributes

###### *hostid*

**Required** attribute; **read/write** access; type: **Integer**.

The hostid of the machine.

###### *mac\_address*

**Required** attribute; **read/write** access; type: **String**.

The main MAC address is the machine.

###### *rtc\_time*

**Required** attribute; **read/write** access; type: **String**.

The date and time of the Real-Time clock.

##### Commands

**create-serengeti-3800-chassis** [*name*] *hostid* "*mac\_address*" "*rtc\_time*"

Creates a non-instantiated component of the class "serengeti-3800-chassis". If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**new-serengeti-3800-chassis** [*name*] *hostid* "*mac\_address*" "*rtc\_time*"

Creates an instantiated component of the class "serengeti-3800-chassis". If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**<serengeti-3800-chassis>.get-nvram-hostid**

Reads the Sun hostid from the NVRAM.

**<serengeti-3800-chassis>.get-nvram-mac**

Reads the default MAC address from the NVRAM.

**<serengeti-3800-chassis>.get-prom-env** [*variable*]

Prints an OBP variable with its value, or all variables if no argument is specified. Only variables with string, integer, boolean and enumeration types are supported.

**<serengeti-3800-chassis>.info**

Print detailed information about the configuration of the device.

**<serengeti-3800-chassis>.set-nvram-hostid** *hostid*

Writes the Sun hostid into the NVRAM.

**<serengeti-3800-chassis>.set-nvram-mac** "*mac*"

Writes the default MAC address into the NVRAM.

**<serengeti-3800-chassis>.set-prom-env** "*variable*" (*int*|"string")

Sets the value of an OBP variable. Only variables with string, integer, boolean and enumeration types are supported.

**<serengeti-3800-chassis>.status**

Print detailed information about the current status of the device.

**Connectors**

Name	Type	Direction
console	serial	down
cpu-slot0	serengeti-cpu-board	down
cpu-slot2	serengeti-cpu-board	down
io-slot6	serengeti-sp-io-board	down
io-slot8	serengeti-sp-io-board	down



### 4.1.2 serengeti-4800-chassis

#### Description

The “sunfire-4800-chassis” component represents the chassis, backplane and system-console of a Sun Fire 4800 server, with slots for up to three processor boards and two I/O boards. The system is sometimes called MD for Midrange Desk-side server.

#### Attributes

##### *hostid*

**Required** attribute; **read/write** access; type: **Integer**.  
The hostid of the machine.

##### *mac\_address*

**Required** attribute; **read/write** access; type: **String**.  
The main MAC address is the machine.

##### *rtc\_time*

**Required** attribute; **read/write** access; type: **String**.  
The date and time of the Real-Time clock.

#### Commands

##### **create-serengeti-4800-chassis** [*“name”*] *hostid* *“mac\_address”* *“rtc\_time”*

Creates a non-instantiated component of the class “serengeti-4800-chassis”. If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

##### **new-serengeti-4800-chassis** [*“name”*] *hostid* *“mac\_address”* *“rtc\_time”*

Creates an instantiated component of the class “serengeti-4800-chassis”. If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

##### **<serengeti-4800-chassis>.get-nvram-hostid**

Reads the Sun hostid from the NVRAM.

##### **<serengeti-4800-chassis>.get-nvram-mac**

Reads the default MAC address from the NVRAM.

##### **<serengeti-4800-chassis>.get-prom-env** [*“variable”*]

Prints an OBP variable with its value, or all variables if no argument is specified. Only variables with string, integer, boolean and enumeration types are supported.

**<serengeti-4800-chassis>.info**

Print detailed information about the configuration of the device.

**<serengeti-4800-chassis>.set-nvram-hostid *hostid***

Writes the Sun hostid into the NVRAM.

**<serengeti-4800-chassis>.set-nvram-mac "*mac*"**

Writes the default MAC address into the NVRAM.

**<serengeti-4800-chassis>.set-prom-env "*variable*" (*int*|"string")**

Sets the value of an OBP variable. Only variables with string, integer, boolean and enumeration types are supported.

**<serengeti-4800-chassis>.status**

Print detailed information about the current status of the device.

**Connectors**

<b>Name</b>	<b>Type</b>	<b>Direction</b>
console	serial	down
cpu-slot0	serengeti-cpu-board	down
cpu-slot2	serengeti-cpu-board	down
cpu-slot4	serengeti-cpu-board	down
io-slot6	serengeti-io-board	down
io-slot8	serengeti-io-board	down

**4.1.3 serengeti-4810-chassis****Description**

The "sunfire-4810-chassis" component represents the chassis, backplane and system-console of a Sun Fire 4810 server, with slots for up to three processor boards and two I/O boards. The system is sometimes called ME for Midrange Enterprise server.

**Attributes*****hostid***

**Required** attribute; **read/write** access; type: **Integer**.

The hostid of the machine.

***mac\_address***

**Required** attribute; **read/write** access; type: **String**.

The main MAC address is the machine.

***rtc\_time***

**Required** attribute; **read/write** access; type: **String**.

The date and time of the Real-Time clock.

**Commands****create-serengeti-4810-chassis** [*“name”*] *hostid* *“mac\_address”* *“rtc\_time”*

Creates a non-instantiated component of the class “serengeti-4810-chassis”. If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**new-serengeti-4810-chassis** [*“name”*] *hostid* *“mac\_address”* *“rtc\_time”*

Creates an instantiated component of the class “serengeti-4810-chassis”. If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**<serengeti-4810-chassis>.get-nvram-hostid**

Reads the Sun hostid from the NVRAM.

**<serengeti-4810-chassis>.get-nvram-mac**

Reads the default MAC address from the NVRAM.

**<serengeti-4810-chassis>.get-prom-env** [*“variable”*]

Prints an OBP variable with its value, or all variables if no argument is specified. Only variables with string, integer, boolean and enumeration types are supported.

**<serengeti-4810-chassis>.info**

Print detailed information about the configuration of the device.

**<serengeti-4810-chassis>.set-nvram-hostid** *hostid*

Writes the Sun hostid into the NVRAM.

**<serengeti-4810-chassis>.set-nvram-mac** *“mac”*

Writes the default MAC address into the NVRAM.

**<serengeti-4810-chassis>.set-prom-env** *“variable”* (*int*|*“string”*)

Sets the value of an OBP variable. Only variables with string, integer, boolean and enumeration types are supported.

**<serengeti-4810-chassis>.status**

Print detailed information about the current status of the device.

**Connectors**

Name	Type	Direction
console	serial	down
cpu-slot0	serengeti-cpu-board	down
cpu-slot2	serengeti-cpu-board	down
cpu-slot4	serengeti-cpu-board	down
io-slot6	serengeti-io-board	down
io-slot8	serengeti-io-board	down

**4.1.4 serengeti-6800-chassis****Description**

The “sunfire-6800-chassis” component represents the chassis, backplane and system-console of a Sun Fire 6800 server, with slots for up to six processor boards and four I/O boards. The system is sometimes called DS for Datacenter Server.

**Attributes***hostid*

**Required** attribute; **read/write** access; type: **Integer**.

The hostid of the machine.

*mac\_address*

**Required** attribute; **read/write** access; type: **String**.

The main MAC address is the machine.

*rtc\_time*

**Required** attribute; **read/write** access; type: **String**.

The date and time of the Real-Time clock.

**Commands****create-serengeti-6800-chassis** [*“name”*] *hostid* *“mac\_address”* *“rtc\_time”*

Creates a non-instantiated component of the class “serengeti-6800-chassis”. If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**new-serengeti-6800-chassis** [*“name”*] *hostid* *“mac\_address”* *“rtc\_time”*

Creates an instantiated component of the class “serengeti-6800-chassis”. If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**<serengeti-6800-chassis>.get-nvram-hostid**

Reads the Sun hostid from the NVRAM.

**<serengeti-6800-chassis>.get-nvram-mac**

Reads the default MAC address from the NVRAM.

**<serengeti-6800-chassis>.get-prom-env ["*variable*"]**

Prints an OBP variable with its value, or all variables if no argument is specified. Only variables with string, integer, boolean and enumeration types are supported.

**<serengeti-6800-chassis>.info**

Print detailed information about the configuration of the device.

**<serengeti-6800-chassis>.set-nvram-hostid *hostid***

Writes the Sun hostid into the NVRAM.

**<serengeti-6800-chassis>.set-nvram-mac "*mac*"**

Writes the default MAC address into the NVRAM.

**<serengeti-6800-chassis>.set-prom-env "*variable*" (*int*|"*string*")**

Sets the value of an OBP variable. Only variables with string, integer, boolean and enumeration types are supported.

**<serengeti-6800-chassis>.status**

Print detailed information about the current status of the device.

## Connectors

Name	Type	Direction
console	serial	down
cpu-slot[0-5]	serengeti-cpu-board	down
io-slot[6-9]	serengeti-io-board	down

### 4.1.5 serengeti-cluster-chassis

#### Description

The "sunfire-cluster-chassis" component represents the chassis, backplane and system-console of up to sixteen Sun Fire 6800 servers connected in a cluster. This setup is to be considered experimental, and Solaris does not support all possible configurations of this component.

**Attributes***hostid*

**Required** attribute; **read/write** access; type: **Integer**.

The hostid of the machine.

*mac\_address*

**Required** attribute; **read/write** access; type: **String**.

The main MAC address is the machine.

*rtc\_time*

**Required** attribute; **read/write** access; type: **String**.

The date and time of the Real-Time clock.

**Commands****create-serengeti-cluster-chassis** [*name*] *hostid* *mac\_address* *rtc\_time*

Creates a non-instantiated component of the class "serengeti-cluster-chassis". If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**new-serengeti-cluster-chassis** [*name*] *hostid* *mac\_address* *rtc\_time*

Creates an instantiated component of the class "serengeti-cluster-chassis". If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**<serengeti-cluster-chassis>.get-nvram-hostid**

Reads the Sun hostid from the NVRAM.

**<serengeti-cluster-chassis>.get-nvram-mac**

Reads the default MAC address from the NVRAM.

**<serengeti-cluster-chassis>.get-prom-env** [*variable*]

Prints an OBP variable with its value, or all variables if no argument is specified. Only variables with string, integer, boolean and enumeration types are supported.

**<serengeti-cluster-chassis>.info**

Print detailed information about the configuration of the device.

**<serengeti-cluster-chassis>.set-nvram-hostid** *hostid*

Writes the Sun hostid into the NVRAM.

**<serengeti-cluster-chassis>.set-nvram-mac "mac"**

Writes the default MAC address into the NVRAM.

**<serengeti-cluster-chassis>.set-prom-env "variable" (int|"string")**

Sets the value of an OBP variable. Only variables with string, integer, boolean and enumeration types are supported.

**<serengeti-cluster-chassis>.status**

Print detailed information about the current status of the device.

## Connectors

Name	Type	Direction
console	serial	down
cpu-slot[0-5]	serengeti-cpu-board	down
cpu-slot[10-15]	serengeti-cpu-board	down
cpu-slot[100-105]	serengeti-cpu-board	down
cpu-slot[110-115]	serengeti-cpu-board	down
cpu-slot[120-125]	serengeti-cpu-board	down
cpu-slot[130-135]	serengeti-cpu-board	down
cpu-slot[140-145]	serengeti-cpu-board	down
cpu-slot[150-155]	serengeti-cpu-board	down
cpu-slot[20-25]	serengeti-cpu-board	down
cpu-slot[30-35]	serengeti-cpu-board	down
cpu-slot[40-45]	serengeti-cpu-board	down
cpu-slot[50-55]	serengeti-cpu-board	down
cpu-slot[60-65]	serengeti-cpu-board	down
cpu-slot[70-75]	serengeti-cpu-board	down
cpu-slot[80-85]	serengeti-cpu-board	down
cpu-slot[90-95]	serengeti-cpu-board	down
io-slot[106-109]	serengeti-io-board	down
io-slot[116-119]	serengeti-io-board	down
io-slot[126-129]	serengeti-io-board	down
io-slot[136-139]	serengeti-io-board	down
io-slot[146-149]	serengeti-io-board	down
io-slot[156-159]	serengeti-io-board	down
io-slot[16-19]	serengeti-io-board	down
io-slot[26-29]	serengeti-io-board	down
io-slot[36-39]	serengeti-io-board	down
io-slot[46-49]	serengeti-io-board	down
io-slot[56-59]	serengeti-io-board	down
io-slot[6-9]	serengeti-io-board	down
io-slot[66-69]	serengeti-io-board	down

io-slot[76-79]	serengeti-io-board	down
io-slot[86-89]	serengeti-io-board	down
io-slot[96-99]	serengeti-io-board	down

#### 4.1.6 serengeti-us-iii-cpu-board

##### Description

The “serengeti-us-iii-cpu-board” component represents a processor board with up to four UltraSPARC III processors and 32GB of memory, for use in Sun Fire 3800-6800 servers.

##### Attributes

###### *cpu\_frequency*

**Required** attribute; **read/write** access; type: **Integer**.  
Processor frequency in MHz.

###### *memory\_megs*

**Required** attribute; **read/write** access; type: **Integer**.  
The amount of RAM in megabytes on the processor board.

###### *num\_cpus*

**Required** attribute; **read/write** access; type: **Integer**.  
Number of processors on the board (0 to 4).

##### Commands

**create-serengeti-us-iii-cpu-board** [*“name”*] *num\_cpus memory\_megs cpu\_frequency*

Creates a non-instantiated component of the class “serengeti-us-iii-cpu-board”. If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**<serengeti-us-iii-cpu-board>.info**

Print detailed information about the configuration of the device.

**<serengeti-us-iii-cpu-board>.status**

Print detailed information about the current status of the device.

##### Connectors

Name	Type	Direction
chassis	serengeti-cpu-board	up
cache-cpu[0-3]	timing-model	down



### 4.1.7 serengeti-us-iii-plus-cpu-board

#### Description

The “serengeti-us-iii-plus-cpu-board” component represents a processor board with up to four UltraSPARC III Cu processors and 32GB of memory, for use in Sun Fire 3800-6800 servers.

#### Attributes

##### *cpu\_frequency*

**Required** attribute; **read/write** access; type: **Integer**.

Processor frequency in MHz.

##### *memory\_megs*

**Required** attribute; **read/write** access; type: **Integer**.

The amount of RAM in megabytes on the processor board.

##### *num\_cpus*

**Required** attribute; **read/write** access; type: **Integer**.

Number of processors on the board (0 to 4).

#### Commands

**create-serengeti-us-iii-plus-cpu-board** [*“name”*] *num\_cpus memory\_megs cpu\_frequency*

Creates a non-instantiated component of the class “serengeti-us-iii-plus-cpu-board”. If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**<serengeti-us-iii-plus-cpu-board>.info**

Print detailed information about the configuration of the device.

**<serengeti-us-iii-plus-cpu-board>.status**

Print detailed information about the current status of the device.

#### Connectors

Name	Type	Direction
chassis	serengeti-cpu-board	up
cache-cpu[0-3]	timing-model	down

### 4.1.8 serengeti-us-iv-cpu-board

#### Description

The “serengeti-us-iv-cpu-board” component represents a processor board with up to four dual-core UltraSPARC IV processors and 32GB of memory, for use in Sun Fire 3800-6800 servers.

#### Attributes

##### *cpu\_frequency*

**Required** attribute; **read/write** access; type: **Integer**.

Processor frequency in MHz.

##### *memory\_megs*

**Required** attribute; **read/write** access; type: **Integer**.

The amount of RAM in megabytes on the processor board.

##### *num\_cpus*

**Required** attribute; **read/write** access; type: **Integer**.

Number of processors on the board (0 to 4).

#### Commands

**create-serengeti-us-iv-cpu-board** [*“name”*] *num\_cpus memory\_megs cpu\_frequency*

Creates a non-instantiated component of the class “serengeti-us-iv-cpu-board”. If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**<serengeti-us-iv-cpu-board>.info**

Print detailed information about the configuration of the device.

**<serengeti-us-iv-cpu-board>.status**

Print detailed information about the current status of the device.

#### Connectors

Name	Type	Direction
chassis	serengeti-cpu-board	up
cache-cpu[0-3]	timing-model	down

### 4.1.9 serengeti-us-iv-plus-cpu-board

#### Description

The “serengeti-us-iv-plus-cpu-board” component represents a processor board with up to four dual-core UltraSPARC IV+ processors and 32GB of memory, for use in Sun Fire 3800-6800 servers.

#### Attributes

##### *cpu\_frequency*

**Required** attribute; **read/write** access; type: **Integer**.

Processor frequency in MHz.

##### *memory\_megs*

**Required** attribute; **read/write** access; type: **Integer**.

The amount of RAM in megabytes on the processor board.

##### *num\_cpus*

**Required** attribute; **read/write** access; type: **Integer**.

Number of processors on the board (0 to 4).

#### Commands

**create-serengeti-us-iv-plus-cpu-board** [*“name”*] *num\_cpus memory\_megs cpu\_frequency*

Creates a non-instantiated component of the class “serengeti-us-iv-plus-cpu-board”. If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**<serengeti-us-iv-plus-cpu-board>.info**

Print detailed information about the configuration of the device.

**<serengeti-us-iv-plus-cpu-board>.status**

Print detailed information about the current status of the device.

#### Connectors

Name	Type	Direction
chassis	serengeti-cpu-board	up
cache-cpu[0-3]	timing-model	down

#### 4.1.10 serengeti-pci8-board

##### Description

The “serengeti-pci8-board” component represents an I/O board with slots for up to eight PCI cards, for use in Sun Fire 4800-6800 servers, i.e. not the SP model.

##### PCI Slot Mappings:

Simics slot	AID	PCI Bus	PCI Slot	Bus address
0	0	B	1	0,700000
1	0	B	2	0,700000
2	0	B	3	0,700000
3	0	A	1	0,600000
4	1	B	1	1,700000
5	1	B	2	1,700000
6	1	B	3	1,700000
7	1	A	1	1,600000

##### Commands

##### **create-serengeti-pci8-board** [*“name”*]

Creates a non-instantiated component of the class “serengeti-pci8-board”. If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

##### **<serengeti-pci8-board>.info**

Print detailed information about the configuration of the device.

##### **<serengeti-pci8-board>.status**

Print detailed information about the current status of the device.

##### Connectors

Name	Type	Direction
chassis	serengeti-io-board	up
pci-slot[0-7]	pci-bus	down

#### 4.1.11 serengeti-cpci4-board

##### Description

The “serengeti-cpci4-board” component represents an I/O board with slots for up to four CompactPCI cards, for use in Sun Fire 4800-6800 servers, i.e. not the SP model.

##### PCI Slot Mappings:

Simics slot	AID	PCI Bus	PCI Slot	Bus address
0	0	B	1	0,700000
1	0	A	1	0,600000
2	1	B	1	1,700000
3	1	A	1	1,600000

## Commands

### **create-serengeti-cpci4-board** [*“name”*]

Creates a non-instantiated component of the class “serengeti-cpci4-board”. If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

### **<serengeti-cpci4-board>.info**

Print detailed information about the configuration of the device.

### **<serengeti-cpci4-board>.status**

Print detailed information about the current status of the device.

## Connectors

Name	Type	Direction
chassis	serengeti-io-board	up
pci-slot[0-3]	cpci-bus	down

### 4.1.12 serengeti-sp-cpci6-board

#### Description

The “serengeti-sp-cpci6-board” component represents an I/O board with slots for up to six CompactPCI cards, for use in a Sun Fire 3800 server, i.e. only the SP model.

#### PCI Slot Mappings:

Simics slot	AID	PCI Bus	PCI Slot	Bus address
0	0	B	1	0,700000
1	0	B	2	0,700000
2	0	A	1	0,600000
3	1	B	1	1,700000
4	1	B	2	1,700000
5	1	A	1	1,600000

## Commands

**create-serengeti-sp-cpci6-board ["name"]**

Creates a non-instantiated component of the class "serengeti-sp-cpci6-board". If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**<serengeti-sp-cpci6-board>.info**

Print detailed information about the configuration of the device.

**<serengeti-sp-cpci6-board>.status**

Print detailed information about the current status of the device.

**Connectors**

Name	Type	Direction
chassis	serengeti-sp-io-board	up
pci-slot[0-5]	cpci-bus	down

## 4.2 PCI Device Components

### 4.2.1 sun-cpci-hme-isp

**Description**

The "sun-cpci-hme-isp" component represents an CompactPCI card with one HME ethernet controller and one ISP SCSI controller for use in Sun systems.

**Attributes***mac\_address*

**Required** attribute; **read/write** access; type: **String**.

The MAC address of the Ethernet adapter.

*scsi\_id*

**Optional** attribute; **read/write** access; type: **Integer**.

The ID on the SCSI bus.

**Commands****create-sun-cpci-hme-isp ["name"] "mac\_address" [scsi\_id]**

Creates a non-instantiated component of the class "sun-cpci-hme-isp". If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**<sun-cpci-hme-isp>.info**

Print detailed information about the configuration of the device.

**<sun-cpci-hme-isp>.status**

Print detailed information about the current status of the device.

**Connectors**

Name	Type	Direction
pci-bus	cpci-bus	up
ethernet	ethernet-link	down
scsi-bus	scsi-bus	down

**4.2.2 sun-cpci-isp-isp****Description**

The “sun-cpci-isp-isp” component represents an CompactPCI card with two ISP SCSI controllers for use in Sun systems.

**Attributes***scsi\_id0*

**Optional** attribute; **read/write** access; type: **Integer**.

The ID on the SCSI bus for the first ISP.

*scsi\_id1*

**Optional** attribute; **read/write** access; type: **Integer**.

The ID on the SCSI bus for the second ISP.

**Commands****create-sun-cpci-isp-isp** [*name*] [*scsi\_id0*] [*scsi\_id1*]

Creates a non-instantiated component of the class “sun-cpci-isp-isp”. If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**<sun-cpci-isp-isp>.info**

Print detailed information about the configuration of the device.

**<sun-cpci-isp-isp>.status**

Print detailed information about the current status of the device.

**Connectors**

Name	Type	Direction
pci-bus	cpci-bus	up
scsi-bus[0-1]	scsi-bus	down

### 4.2.3 sun-cpci-qlc-qlc

#### Description

The “sun-cpci-qlc-qlc” component represents an CompactPCI card with two QLC Fibre-Channel SCSI controllers for use in Sun systems.

#### Attributes

##### *loop\_id0*

**Required** attribute; **read/write** access; type: **Integer**.

The FC loop ID of the first QLC controller.

##### *loop\_id1*

**Required** attribute; **read/write** access; type: **Integer**.

The FC loop ID of the second QLC controller.

#### Commands

##### **create-sun-cpci-qlc-qlc** [*“name”*] *loop\_id0 loop\_id1*

Creates a non-instantiated component of the class “sun-cpci-qlc-qlc”. If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

##### **<sun-cpci-qlc-qlc>.info**

Print detailed information about the configuration of the device.

##### **<sun-cpci-qlc-qlc>.status**

Print detailed information about the current status of the device.

#### Connectors

Name	Type	Direction
pci-bus	cpci-bus	up
fc-loop[0-1]	simple-fc-loop	down

### 4.2.4 sun-pci-ce

#### Description

The “sun-pci-ce” component represents a PCI card with a Cassini gigabit Ethernet adapter, for use in Sun systems.

#### Attributes

##### *mac\_address*

**Required** attribute; **read/write** access; type: **String**.

The MAC address of the Ethernet adapter.



**Commands****create-sun-pci-ce** [*name*] *mac\_address*

Creates a non-instantiated component of the class “sun-pci-ce”. If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**<sun-pci-ce>.info**

Print detailed information about the configuration of the device.

**<sun-pci-ce>.status**

Print detailed information about the current status of the device.

**Connectors**

Name	Type	Direction
pci-bus	pci-bus	up
ethernet	ethernet-link	down

**4.2.5 sun-pci-hme****Description**

The “sun-pci-hme” component represents a PCI card with a HME Ethernet adapter, for use in Sun systems.

**Attributes***mac\_address*

**Required** attribute; **read/write** access; type: **String**.

The MAC address of the Ethernet adapter.

**Commands****create-sun-pci-hme** [*name*] *mac\_address*

Creates a non-instantiated component of the class “sun-pci-hme”. If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**<sun-pci-hme>.info**

Print detailed information about the configuration of the device.

**<sun-pci-hme>.status**

Print detailed information about the current status of the device.

**Connectors**

Name	Type	Direction
pci-bus	pci-bus	up
ethernet	ethernet-link	down

**4.2.6 sun-pci-hme-isp****Description**

The “sun-pci-hme-isp” component represents a PCI card with one HME Ethernet adapter and one ISP SCSI controller for use in Sun systems.

**Attributes***mac\_address*

**Required** attribute; **read/write** access; type: **String**.

The MAC address of the Ethernet adapter.

*scsi\_id*

**Optional** attribute; **read/write** access; type: **Integer**.

The ID on the SCSI bus.

**Commands****create-sun-pci-hme-isp** [*“name”*] [*“mac\_address”*] [*scsi\_id*]

Creates a non-instantiated component of the class “sun-pci-hme-isp”. If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

<**sun-pci-hme-isp**>.info

Print detailed information about the configuration of the device.

<**sun-pci-hme-isp**>.status

Print detailed information about the current status of the device.

**Connectors**

Name	Type	Direction
pci-bus	pci-bus	up
ethernet	ethernet-link	down
scsi-bus	scsi-bus	down

### 4.2.7 sun-pci-pgx64

#### Description

The “sun-pci-pgx64” component represents a PCI card with a PGX64 (Rage XL) graphics adapter, for use in Sun systems.

#### Commands

##### **create-sun-pci-pgx64** [*“name”*]

Creates a non-instantiated component of the class “sun-pci-pgx64”. If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

##### **<sun-pci-pgx64>.info**

Print detailed information about the configuration of the device.

##### **<sun-pci-pgx64>.status**

Print detailed information about the current status of the device.

#### Connectors

Name	Type	Direction
pci-bus	pci-bus	up
console	graphics-console	down

### 4.2.8 sun-pci-qlc

#### Description

The “sun-pci-qlc” component represents a PCI card with a QLC Fibre-Channel SCSI controller for use in Sun systems.

#### Attributes

##### *loop\_id*

**Required** attribute; **read/write** access; type: **Integer**.

The FC loop ID of the QLC controller.

#### Commands

##### **create-sun-pci-qlc** [*“name”*] *loop\_id*

Creates a non-instantiated component of the class “sun-pci-qlc”. If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**<sun-pci-qlc>.info**

Print detailed information about the configuration of the device.

**<sun-pci-qlc>.status**

Print detailed information about the current status of the device.

**Connectors**

Name	Type	Direction
pci-bus	pci-bus	up
fc-loop	simple-fc-loop	down

**4.2.9 sun-pci-qlc-qlc****Description**

The “sun-pci-qlc-qlc” component represents a PCI card with two QLC Fibre-Channel SCSI controller for use in Sun systems.

**Attributes*****loop\_id0***

**Required** attribute; **read/write** access; type: **Integer**.

The FC loop ID of the first QLC controller.

***loop\_id1***

**Required** attribute; **read/write** access; type: **Integer**.

The FC loop ID of the second QLC controller.

**Commands****create-sun-pci-qlc-qlc [*“name”*] *loop\_id0 loop\_id1***

Creates a non-instantiated component of the class “sun-pci-qlc-qlc”. If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**<sun-pci-qlc-qlc>.info**

Print detailed information about the configuration of the device.

**<sun-pci-qlc-qlc>.status**

Print detailed information about the current status of the device.

**Connectors**

Name	Type	Direction
pci-bus	pci-bus	up
fc-loop[0-1]	simple-fc-loop	down

**4.2.10 pci-bcm5703c****Description**

The “pci-bcm5703c” component represents a Broadcom 5703C PCI based gigabit Ethernet adapter.

**Attributes***bios*

**Optional** attribute; **read/write** access; type: **String**.  
The x86 BIOS file to use.

*mac\_address*

**Required** attribute; **read/write** access; type: **String**.  
The MAC address of the Ethernet adapter.

**Commands****create-pci-bcm5703c** [*“name”*] [*“mac\_address”*] [*“bios”*]

Creates a non-instantiated component of the class “pci-bcm5703c”. If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

## &lt;pci-bcm5703c&gt;.info

Print detailed information about the configuration of the device.

## &lt;pci-bcm5703c&gt;.status

Print detailed information about the current status of the device.

**Connectors**

Name	Type	Direction
pci-bus	pci-bus	up
ethernet	ethernet-link	down

### 4.2.11 pci-bcm5704c

#### Description

The “pci-bcm5704c” component represents a Broadcom 5704C PCI based dual-port gigabit Ethernet adapter.

#### Attributes

##### *bios*

**Optional** attribute; **read/write** access; type: **String**.

The x86 BIOS file to use.

##### *mac\_address0*

**Required** attribute; **read/write** access; type: **String**.

The MAC address of the first Ethernet adapter.

##### *mac\_address1*

**Required** attribute; **read/write** access; type: **String**.

The MAC address of the second Ethernet adapter.

#### Commands

**create-pci-bcm5704c** [*“name”*] [*“mac\_address0”*] [*“mac\_address1”*] [*“bios”*]

Creates a non-instantiated component of the class “pci-bcm5704c”. If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**<pci-bcm5704c>.info**

Print detailed information about the configuration of the device.

**<pci-bcm5704c>.status**

Print detailed information about the current status of the device.

#### Connectors

Name	Type	Direction
pci-bus	pci-bus	up
ethernet[0-1]	ethernet-link	down

### 4.2.12 pci-sym53c875

#### Description

The “pci-sym53C875” component represents a SYM53C875PCI based SCSI controller.

#### Attributes

##### *bios*

**Optional** attribute; **read/write** access; type: **String**.

The x86 SCSI BIOS file to use.

#### Commands

##### **create-pci-sym53c875** [*“name”*] [*“bios”*]

Creates a non-instantiated component of the class “pci-sym53c875”. If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

##### **<pci-sym53c875>.info**

Print detailed information about the configuration of the device.

##### **<pci-sym53c875>.status**

Print detailed information about the current status of the device.

#### Connectors

Name	Type	Direction
pci-bus	pci-bus	up
scsi-bus	scsi-bus	down

### 4.2.13 pci-sym53c876

#### Description

The “pci-sym53C876” component represents a SYM53C876PCI based dual-port SCSI controller.

#### Commands

##### **create-pci-sym53c876** [*“name”*]

Creates a non-instantiated component of the class “pci-sym53c876”. If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**<pci-sym53c876>.info**

Print detailed information about the configuration of the device.

**<pci-sym53c876>.status**

Print detailed information about the current status of the device.

**Connectors**

Name	Type	Direction
pci-bus	pci-bus	up
scsi-bus[0-1]	scsi-bus	down

## 4.3 Standard Components

### 4.3.1 std-ethernet-link

**Description**

The “std-ethernet-link” component represents a standard Ethernet link.

**Attributes***frame\_echo*

**Optional** attribute; **read/write** access; type: **Integer**.

Set this attribute to echo frames back to the sender. Default is not to echo frames.

*link\_name*

**Optional** attribute; **read/write** access; type: **String**.

The name to use for the **ethernet-link** object. An error will be raised at instantiation time if the link cannot be given this name.

**Commands****create-std-ethernet-link** [*“name”*] [*“link\_name”*] [*frame\_echo*]

Creates a non-instantiated component of the class “std-ethernet-link”. If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**new-std-ethernet-link** [*“name”*] [*“link\_name”*] [*frame\_echo*]

Creates an instantiated component of the class “std-ethernet-link”. If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**<std-ethernet-link>.info**

Print detailed information about the configuration of the device.



**<std-ethernet-link>.status**

Print detailed information about the current status of the device.

**Connectors**

Name	Type	Direction
device	ethernet-link	any

**4.3.2 std-service-node****Description**

The “std-service-node” component represents a network service node that can be connected to Ethernet links to provide services such as DNS, DHCP/BOOTP, RARP and TFTP. A service node component does not have any connectors by default. Instead, connectors have to be added using the **<std-service-node>.add-connector** command.

**Attributes***dynamic\_connectors*

**Optional** attribute; **read/write** access; type: **[[iss]\*]**.  
List of user added connectors

*next\_connector\_id*

**Optional** attribute; **read/write** access; type: **Integer**.  
Next service-node device ID.

**Commands****create-std-service-node** [*“name”*]

Creates a non-instantiated component of the class “std-service-node”. If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**new-std-service-node** [*“name”*]

Creates an instantiated component of the class “std-service-node”. If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**<std-service-node>.add-connector** *“ip”* [*“netmask”*]

Adds a connector to the service-node with specified IP address and netmask. A connector must be created for the service-node before an Ethernet link can be connected to it. The *ip* argument is the IP address that the service node will use on the link. The *netmask* argument is optional, and defaults to 255.255.255.0. The name of the new connector is returned.

**<std-service-node>.info**

Print detailed information about the configuration of the device.

**<std-service-node>.status**

Print detailed information about the current status of the device.

**4.3.3 std-scsi-bus****Description**

The “std-scsi-bus” component represents a 16 slot SCSI bus.

**Commands****create-std-scsi-bus [“name”]**

Creates a non-instantiated component of the class “std-scsi-bus”. If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**<std-scsi-bus>.info**

Print detailed information about the configuration of the device.

**<std-scsi-bus>.status**

Print detailed information about the current status of the device.

**Connectors**

Name	Type	Direction
scsi-bus	scsi-bus	any

**4.3.4 std-scsi-disk****Description**

The “std-scsi-disk” component represents a SCSI-2 disk.

**Attributes***file*

**Optional** attribute; **read/write** access; type: **String**.

File with disk contents for the full disk Either a raw file or a CRAFF file.

*scsi\_id*

**Required** attribute; **read/write** access; type: **Integer**.

The ID on the SCSI bus.

*size*

**Required** attribute; **read/write** access; type: **Integer**.

The size of the SCSI disk in bytes.

## Commands

**create-std-scsi-disk** [*name*] *scsi\_id* *size* [*file*]

Creates a non-instantiated component of the class "std-scsi-disk". If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**<std-scsi-disk>.info**

Print detailed information about the configuration of the device.

**<std-scsi-disk>.status**

Print detailed information about the current status of the device.

## Connectors

Name	Type	Direction
scsi-bus	scsi-bus	up

### 4.3.5 std-scsi-cdrom

#### Description

The "std-scsi-cdrom" component represents a SCSI-2 CD-ROM.

#### Attributes

*scsi\_id*

**Required** attribute; **read/write** access; type: **Integer**.

The ID on the SCSI bus.

## Commands

**create-std-scsi-cdrom** [*name*] *scsi\_id*

Creates a non-instantiated component of the class "std-scsi-cdrom". If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**<std-scsi-cdrom>.info**

Print detailed information about the configuration of the device.

**<std-scsi-cdrom>.status**

Print detailed information about the current status of the device.

**Connectors**

Name	Type	Direction
scsi-bus	scsi-bus	up

**4.3.6 simple-fc-disk****Description**

The “simple-fc-disk” component represents a SCSI-2 disk for use with Fibre-Channel SCSI controllers using the simplified FC protocol in Simics.

**Attributes***file*

**Optional** attribute; **read/write** access; type: **String**.

File with disk contents for the full disk Either a raw file or a CRAFF file.

*loop\_id*

**Required** attribute; **read/write** access; type: **Integer**.

The loop ID for the FC disk.

*node\_name*

**Required** attribute; **read/write** access; type: **Integer**.

The node name for the FC disk.

*port\_name*

**Required** attribute; **read/write** access; type: **Integer**.

The port name for the FC disk.

*size*

**Required** attribute; **read/write** access; type: **Integer**.

The size of the FC disk in bytes.

**Commands****create-simple-fc-disk [“name”] size [“file”] loop\_id node\_name port\_name**

Creates a non-instantiated component of the class “simple-fc-disk”. If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**<simple-fc-disk>.info**

Print detailed information about the configuration of the device.

**<simple-fc-disk>.status**

Print detailed information about the current status of the device.

**Connectors**

Name	Type	Direction
fc-loop	simple-fc-loop	up

**4.3.7 std-text-console****Description**

The “std-text-console” component represents a serial text console.

**Attributes***bg\_color*

**Optional** attribute; **read/write** access; type: **String**.

The background color.

*fg\_color*

**Optional** attribute; **read/write** access; type: **String**.

The foreground color.

*height*

**Optional** attribute; **read/write** access; type: **Integer**.

The height of the console window.

*title*

**Optional** attribute; **read/write** access; type: **String**.

The Window title.

*width*

**Optional** attribute; **read/write** access; type: **Integer**.

The width of the console window.

*win32\_font*

**Optional** attribute; **read/write** access; type: **String**.

Font to use in the console on Windows host.

*x11\_font*

**Optional** attribute; **read/write** access; type: **String**.

Font to use in the console when using X11 (Linux/Solaris host).

**Commands**

**create-std-text-console** ["name"] ["title"] ["bg\_color"] ["fg\_color"] ["x11\_font"] ["win32\_font"] [w]

Creates a non-instantiated component of the class "std-text-console". If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**new-std-text-console** ["name"] ["title"] ["bg\_color"] ["fg\_color"] ["x11\_font"] ["win32\_font"] [w]

Creates an instantiated component of the class "std-text-console". If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**<std-text-console>.info**

Print detailed information about the configuration of the device.

**<std-text-console>.status**

Print detailed information about the current status of the device.

## Connectors

Name	Type	Direction
serial	serial	up

### 4.3.8 std-server-console

#### Description

The "std-server-console" component represents a serial console accessible from the host using telnet.

#### Attributes

*telnet\_port*

**Required** attribute; **read/write** access; type: **Integer**.

TCP/IP port to connect the telnet service of the console to.

#### Commands

**create-std-server-console** ["name"] *telnet\_port*

Creates a non-instantiated component of the class "std-server-console". If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**new-std-server-console ["name"] telnet\_port**

Creates an instantiated component of the class "std-server-console". If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**<std-server-console>.info**

Print detailed information about the configuration of the device.

**<std-server-console>.status**

Print detailed information about the current status of the device.

**Connectors**

Name	Type	Direction
serial	serial	up

**4.3.9 std-graphics-console****Description**

The "std-graphics-console" component represents a graphical console for displaying output from a simulated graphics adapters and getting input for mouse and keyboard devices.

**Attributes***window*

**Optional** attribute; **read/write** access; type: **b**.

Try to open window if TRUE (default). FALSE disabled the window.

**Commands****create-std-graphics-console ["name"] [window]**

Creates a non-instantiated component of the class "std-graphics-console". If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**new-std-graphics-console ["name"] [window]**

Creates an instantiated component of the class "std-graphics-console". If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**<std-graphics-console>.info**

Print detailed information about the configuration of the device.

**<std-graphics-console>.status**

Print detailed information about the current status of the device.

**Connectors**

Name	Type	Direction
device	graphics-console	up
keyboard	keyboard	up
mouse	mouse	up

**4.3.10 std-text-graphics-console****Description**

The “std-text-graphics-console” component represents a text console for use with VGA instead of a graphics console.

**Commands****create-std-text-graphics-console [“name”]**

Creates a non-instantiated component of the class “std-text-graphics-console”. If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**new-std-text-graphics-console [“name”]**

Creates an instantiated component of the class “std-text-graphics-console”. If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**<std-text-graphics-console>.info**

Print detailed information about the configuration of the device.

**<std-text-graphics-console>.status**

Print detailed information about the current status of the device.

**Connectors**

Name	Type	Direction
device	graphics-console	up
keyboard	keyboard	up



## 4.4 Timing Components

### 4.4.1 sample-gcache

#### Description

A pre-configured combined L1 instruction and data cache

#### Commands

##### **create-sample-gcache** [*name*]

Creates a non-instantiated component of the class "sample-gcache". If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

##### **new-sample-gcache** [*name*]

Creates an instantiated component of the class "sample-gcache". If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

##### **<sample-gcache>.info**

Print detailed information about the configuration of the device.

##### **<sample-gcache>.status**

Print detailed information about the current status of the device.

#### Connectors

Name	Type	Direction
cpu-space	timing-model	up

### 4.4.2 sample-ma-model

#### Description

A sample SPARC MAI model with a simple cache

#### Commands

##### **create-sample-ma-model** [*name*]

Creates a non-instantiated component of the class "sample-ma-model". If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**new-sample-ma-model ["name"]**

Creates an instantiated component of the class "sample-ma-model". If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**<sample-ma-model>.info**

Print detailed information about the configuration of the device.

**<sample-ma-model>.status**

Print detailed information about the current status of the device.

**Connectors**

Name	Type	Direction
cpu-space	timing-model	up

**4.4.3 sample-ooo-model****Description**

A sample SPARC MAI model based on **ooo\_micro\_arch** and a simple cache.

**Commands****create-sample-ooo-model ["name"]**

Creates a non-instantiated component of the class "sample-ooo-model". If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**new-sample-ooo-model ["name"]**

Creates an instantiated component of the class "sample-ooo-model". If *name* is not specified, the component will get a class-specific default name. The other arguments correspond to class attributes.

**<sample-ooo-model>.info**

Print detailed information about the configuration of the device.

**<sample-ooo-model>.status**

Print detailed information about the current status of the device.

**Connectors**

Name	Type	Direction
cpu-space	timing-model	up

## 4.5 Base Components

The base components are abstract classes that contain generic component attributes and commands available for all components.

### 4.5.1 component

**Description**

Base component class, should not be instantiated.

**Attributes***connections*

**Optional** attribute; **read/write** access; type: **[[sos]\*]**.

List of connections for the component. The format is a list of lists, each containing the name of the connector, the connected component, and the name of the connector on the other component.

*connectors*

**Pseudo class** attribute; **read-only** access; type: **D**.

Dictionary of dictionaries with connectors defined by this component class, indexed by name. Each connector contains the name of the connector “type”, a “direction” (“up”, “down” or “any”), a flag indicating if the connector can be “empty”, another flag that is set if the connector is “hotplug” capable, and finally a flag that is TRUE if multiple connections to this connector is allowed.

*instantiated*

**Optional** attribute; **read/write** access; type: **b**.

Set to TRUE if the component has been instantiated.

*object\_list*

**Optional** attribute; **read/write** access; type: **D**.

Dictionary with objects that the component consists of.

*object\_prefix*

**Optional** attribute; **read/write** access; type: **String**.

Object prefix string used by the component. The prefix is typically set by the **set-component-prefix** command before the component is created.

*top\_component*

**Optional** attribute; **read/write** access; type: **Object**.

The top level component. Attribute is not valid until the component has been instantiated.

*top\_level*

**Optional** attribute; **read/write** access; type: **b**.

Set to TRUE for top-level components, i.e. the root of a hierarchy.

## 4.5.2 top-component

### Description

Base top-level component class, should not be instantiated.

### Attributes

*components*

**Optional** attribute; **read/write** access; type: **[o\*]**.

List of components below the the top-level component. This attribute is not valid until the object has been instantiated.

*cpu\_list*

**Optional** attribute; **read/write** access; type: **[o\*]**.

List of all processors below the the top-level component. This attribute is not valid until the object has been instantiated.

## Chapter 5

# Examples

### 5.1 Adding a new Disk to a Serengeti Machine

#### 1. Add a SCSI Disk Component

First create a file in `[workspace]/targets/serengeti/` called `abisko-disk.simics`. In this file add the following contents:

```
script-branch {
    wait-for-variable machine_defined
    local $disk = (create-std-scsi-disk size = 4256972800)
    connect-components $scsi_bus scsi-slot-2 $disk scsi-bus
}

run-command-file abisko-common.simics
```

This will run a *script branch* that first waits for the machine to be defined by the machine configuration script (included from `abisko-common.simics`). Once the `$machine_defined` variable has triggered, a SCSI disk component, representing a 4GB disk, will be created that on the following line is connected to the SCSI bus component on connector `scsi-slot-2`.

#### 2. Prepare the Boot

Start Simics, but do not start the simulation. Before booting, the disk needs an empty partition table for Solaris to recognize the disk. The partition table must contain a geometry that matches the size of the disk. Also add a large partition that covers the full disk.

```
simics> sd1.create-sun-vtoc-header 5470 19 80
simics> sd1.create-sun-vtoc-partition number = 0 start-block = 0 num-blocks = 4256972800
```

---

**Note:** Configuring partitions can also be done using the Solaris **format** command once the simulated machine has booted.

---

The partition table should now look something like:

Partition Table:

Number	Tag	Flag	Start	End	Size
0	2 (root)	0 (RW)	0	8311359	8311360
2	5 (backup)	1 (unmountable)	0	8311359	8311360

Before booting, tell Solaris that new hardware has been added, by adding the `-r` argument to the OBP boot command variable:

```
simics> system_cmp0.set-prom-env boot-command "boot disk -rv"
```

### 3. Configure Solaris

Start the simulation and wait for the simulated machine to reach the prompt. A file system has to be created on the new disk, this is done using the Solaris **newfs** command. At the same time, also add a mount point, and an entry in the file-system table. This way Solaris will automatically mount the disk on the next boot.

```
# newfs /dev/dsk/c0t2d0s0
newfs: construct a new file system /dev/rdisk/c0t2d0s0: (y/n)? y
```

<output from newfs here>

```
# mkdir /disk
# cat >> /etc/vfstab
/dev/dsk/c0t2d0s0    /dev/rdisk/c0t2d0s0    /disk ufs 2    yes    -
<control-D>
# mount /disk
```

The disk can now be accessed as `/disk/` in the file-system.

### 4. Save the Changes

To save the changes to the new disk, shut down the simulated machine and save the modifications. Issue:

```
# init 0
```

then wait for Solaris to shut down, stop the simulation and save the all modifications using the **save-persistent-state** command. But first remove `-r` from the boot command, or it will be for the next boot as well, making the boot slower.

```
simics> system_cmp0.set-prom-env boot-command "boot disk -v"  
simics> save-persistent-state new-disk1.state
```

Now exit Simics, and restart the `abisko-disk.simics` script. Before running, load the disk modifications saved earlier:

```
simics> load-persistent-state new-disk1.state
```

Now boot the machine again. The new disk will be mounted as `/disk/`.

## Chapter 6

# Installing an OS on Simics

### 6.1 Installing Solaris on Simics

Solaris can be installed directly on the simulated machine in Simics. Solaris can be obtained from Sun's web-site at <http://www.sun.com/software/solaris/binaries/get.html> in the form of ISO images.

To simplify the installation process, some scripts are supplied with the Simics distribution for the abisko machine: `abisko-sol<version>-cd-install11.simics`, `abisko-sol<version>-cd-install12.simics` and `abisko-sol<version>-cd-install13.simics`, where `<version>` is 8, 9 or 10. The scripts will answer all questions automatically to create a standard workstation install.

#### 6.1.1 Installation, step by step

This section describes how to install Solaris using the command-line version Simics.

1. Select the install script to use, depending on Solaris version to install, either `abisko-sol10-cd-install11.simics` for Solaris 10, `abisko-sol9-cd-install11.simics` for Solaris 9, or `abisko-sol8-cd-install11.simics` for Solaris 8.
2. Set the path to the CD image in the simics script. The line

```
$cdrom_path = "sol-10-u2-ga-sparc-v1.iso"
```

should be changed to reflect the location and name of the CD image for stage one of the installation. It can either be an ISO image file, or a CD-ROM device file (Linux and Solaris host only).

3. Start the first installation script, for example:

```
$ ./simics targets/serengeti/abisko-sol10-cd-install11.simics
```

and wait for it to complete. This may take several hours, depending on the performance of the host machine.



4. When the script stops, installation from the first CD is finished, and Solaris has tried to reboot the system. Since Simics does not support system reboot for this architecture, exit Simics at this point.

If the installation is performed from a real CD, it is now time to change disc in the drive. Also make sure that the path to the CD is correct in the second install script.

5. Now run the second script in the same way as the first, this script may also take a few hours to complete.
6. When the second script has stopped, run the third and last one. This script only takes a few minutes to finish.
7. When the third script has stopped the installation is ready. The newly created disk image has the following file name: `abisko-sol<version>-install.disk`. There are also a number of *persistent state* files.
8. To boot a machine with the newly installed Solaris OS, run the `abisko-common.simics` and make sure that the variable `$os` is set to "solaris10", "solaris9" or "solaris8" depending on the operating system version installed). Add a line like the following first in that simics script:

```
$os = "solaris10"
```

9. An optional last step is to compress the disk image with the `craff` utility to save some disk space.

## Chapter 7

# Miscellaneous Notes

### 7.1 Notes on Solaris for Serengeti

- For information about system administration of Solaris, see the <http://docs.sun.com> web site.
- Remember to boot with the `-r` flag after changing a machine configuration. Example:

```
simics> system_cmp0.set-prom-env boot-command "boot disk -rv"
```

- Booting from a disk in a different location that it was setup for is not recommended. It is possible, but requires some knowledge of Solaris administration.
- When using multiple Ethernet adapters in a Serengeti system, all will be assigned the same system-wide MAC address by Solaris. To avoid this, the OBP variable `local-mac-address?` can be set to `true`. Setting this variable from the Simics command-line is done using the following command:

```
simics> system_cmp0.set-prom-env local-mac-address? true
```

### 7.2 Multiple Network Devices

By default, only the first network device is connected to a simulated network when running with `$create_network` set to `yes`. To run with multiple network devices, the `<device>.connect` command should be used to connect each additional device to the Ethernet link. If several network devices have the same MAC address (default unless the `local-mac-address?` OBP variable is set) they must be connected to different simulated links.

## 7.3 Changing the Processor Clock Frequency

The clock frequency of a simulated processor can be set arbitrarily in Simics. This will not affect the actual speed of simulation, but it will affect the number of instructions that need to be executed for a certain amount of simulated time to pass. If your execution only depends on executing a certain number of instructions, increasing the clock frequency will take the same amount of host time (but a shorter amount of target time). However, if there are time based delays of some kind in the simulation, these will take longer to execute.

At a simulated 1 MHz, one million target instructions will correspond to a simulated second (assuming the simple default timing of one cycle per instruction). At 100 MHz, on the other hand, it will take 100 million target instructions to complete a simulated second. So with a higher clock frequency, less simulated target time is going to pass for a certain period of host execution time.

If Simics is used to emulate an interactive system (especially one with a graphical user interface) it is a good idea to set the clock frequency quite low. Keyboard and mouse inputs events are handled by periodic interrupts in most operating systems, using a higher clock frequency will result in longer delays between invocations of periodic interrupts. Thus, the simulated system will feel slower in its user response, and update the mouse cursor position etc. less frequently. If this is a problem, the best technique for running experiments at a high clock frequency is to first complete the configuration of the machine using a low clock frequency. Save all configuration changes to a disk diff (like when installing operating systems). Then change the configuration to use a higher a clock frequency and reboot the target machine.

Note that for a lightly-loaded machine (for example, working at an interactive prompt on a serial console to an embedded Linux system), Simics will often execute quickly enough at the real target clock frequency that there is no need to artificially lower it.

## **Chapter 8**

# **Limitations**

### **8.1 Limitations of the Simulated Model**

- The following UltraSPARC registers are not implemented:
  - The ECC error registers.
  - Cache diagnostic registers.
  - Performance control registers, and counters.
- The System Controller in Simics is only emulated, and has no support for system management such as dynamic reconfiguration and domain configuration.
- System reset is not supported.

### **8.2 Other Limitations**

- The Solaris version of SimicsFS does not support truncating files.

# Index

## Symbols

[simics], [6](#)

[workspace], [6](#)

## A

add-connector

namespace command

std-service-node, [41](#)

## C

component, [51](#)

configuration

tips, [59](#)

create-pci-bcm5703c, [37](#)

create-pci-bcm5704c, [38](#)

create-pci-sym53c875, [39](#)

create-pci-sym53c876, [39](#)

create-sample-gcache, [49](#)

create-sample-ma-model, [49](#)

create-sample-ooo-model, [50](#)

create-serengeti-3800-chassis, [15](#)

create-serengeti-4800-chassis, [17](#)

create-serengeti-4810-chassis, [19](#)

create-serengeti-6800-chassis, [20](#)

create-serengeti-cluster-chassis, [22](#)

create-serengeti-cpci4-board, [29](#)

create-serengeti-pci8-board, [28](#)

create-serengeti-sp-cpci6-board, [29](#)

create-serengeti-us-iii-cpu-board, [24](#)

create-serengeti-us-iii-plus-cpu-board, [25](#)

create-serengeti-us-iv-cpu-board, [26](#)

create-serengeti-us-iv-plus-cpu-board, [27](#)

create-simple-fc-disk, [44](#)

create-std-ethernet-link, [40](#)

create-std-graphics-console, [47](#)

create-std-scsi-bus, [42](#)

create-std-scsi-cdrom, [43](#)

create-std-scsi-disk, [43](#)

create-std-server-console, [46](#)

create-std-service-node, [41](#)

create-std-text-console, [45](#)

create-std-text-graphics-console, [48](#)

create-sun-cpci-hme-isp, [30](#)

create-sun-cpci-isp-isp, [31](#)

create-sun-cpci-qlc-qlc, [32](#)

create-sun-pci-ce, [33](#)

create-sun-pci-hme, [33](#)

create-sun-pci-hme-isp, [34](#)

create-sun-pci-pgx64, [35](#)

create-sun-pci-qlc, [35](#)

create-sun-pci-qlc-qlc, [36](#)

## G

get-nvram-hostid

namespace command

serengeti-3800-chassis, [16](#)

serengeti-4800-chassis, [17](#)

serengeti-4810-chassis, [19](#)

serengeti-6800-chassis, [20](#)

serengeti-cluster-chassis, [22](#)

get-nvram-mac

namespace command

serengeti-3800-chassis, [16](#)

serengeti-4800-chassis, [17](#)

serengeti-4810-chassis, [19](#)

serengeti-6800-chassis, [21](#)

serengeti-cluster-chassis, [22](#)

get-prom-env

namespace command

serengeti-3800-chassis, [16](#)

serengeti-4800-chassis, [17](#)

serengeti-4810-chassis, [19](#)

serengeti-6800-chassis, [21](#)

serengeti-cluster-chassis, [22](#)

**I**

## info

## namespace command

- pci-bcm5703c, [37](#)
- pci-bcm5704c, [38](#)
- pci-sym53c875, [39](#)
- pci-sym53c876, [39](#)
- sample-gcache, [49](#)
- sample-ma-model, [50](#)
- sample-ooo-model, [50](#)
- serengeti-3800-chassis, [16](#)
- serengeti-4800-chassis, [17](#)
- serengeti-4810-chassis, [19](#)
- serengeti-6800-chassis, [21](#)
- serengeti-cluster-chassis, [22](#)
- serengeti-cpci4-board, [29](#)
- serengeti-pci8-board, [28](#)
- serengeti-sp-cpci6-board, [30](#)
- serengeti-us-iii-cpu-board, [24](#)
- serengeti-us-iii-plus-cpu-board, [25](#)
- serengeti-us-iv-cpu-board, [26](#)
- serengeti-us-iv-plus-cpu-board, [27](#)
- simple-fc-disk, [44](#)
- std-ethernet-link, [40](#)
- std-graphics-console, [47](#)
- std-scsi-bus, [42](#)
- std-scsi-cdrom, [43](#)
- std-scsi-disk, [43](#)
- std-server-console, [47](#)
- std-service-node, [42](#)
- std-text-console, [46](#)
- std-text-graphics-console, [48](#)
- sun-cpci-hme-isp, [30](#)
- sun-cpci-isp-isp, [31](#)
- sun-cpci-qlc-qlc, [32](#)
- sun-pci-ce, [33](#)
- sun-pci-hme, [33](#)
- sun-pci-hme-isp, [34](#)
- sun-pci-pgx64, [35](#)
- sun-pci-qlc, [35](#)
- sun-pci-qlc-qlc, [36](#)

interactive use of simulated machines, [59](#)

**N**

new-sample-gcache, [49](#)

new-sample-ma-model, [49](#)

new-sample-ooo-model, [50](#)

new-serengeti-3800-chassis, [16](#)

new-serengeti-4800-chassis, [17](#)

new-serengeti-4810-chassis, [19](#)

new-serengeti-6800-chassis, [20](#)

new-serengeti-cluster-chassis, [22](#)

new-std-ethernet-link, [40](#)

new-std-graphics-console, [47](#)

new-std-server-console, [46](#)

new-std-service-node, [41](#)

new-std-text-console, [46](#)

new-std-text-graphics-console, [48](#)

**P**

pci-bcm5703c, [37](#)

pci-bcm5704c, [38](#)

pci-sym53c875, [39](#)

pci-sym53c876, [39](#)

processor clock frequency, [59](#)

**S**

sample-gcache, [49](#)

sample-ma-model, [49](#)

sample-ooo-model, [50](#)

serengeti-3800-chassis, [15](#)

serengeti-4800-chassis, [17](#)

serengeti-4810-chassis, [18](#)

serengeti-6800-chassis, [20](#)

serengeti-cluster-chassis, [21](#)

serengeti-cpci4-board, [28](#)

serengeti-pci8-board, [28](#)

serengeti-sp-cpci6-board, [29](#)

serengeti-us-iii-cpu-board, [24](#)

serengeti-us-iii-plus-cpu-board, [25](#)

serengeti-us-iv-cpu-board, [26](#)

serengeti-us-iv-plus-cpu-board, [27](#)

set-nvram-hostid

namespace command

serengeti-3800-chassis, [16](#)

serengeti-4800-chassis, [18](#)

serengeti-4810-chassis, [19](#)

serengeti-6800-chassis, [21](#)

serengeti-cluster-chassis, [22](#)

set-nvram-mac

namespace command

- serengeti-3800-chassis, [16](#)
  - serengeti-4800-chassis, [18](#)
  - serengeti-4810-chassis, [19](#)
  - serengeti-6800-chassis, [21](#)
  - serengeti-cluster-chassis, [22](#)
  - set-prom-env
    - namespace command
      - serengeti-3800-chassis, [16](#)
      - serengeti-4800-chassis, [18](#)
      - serengeti-4810-chassis, [19](#)
      - serengeti-6800-chassis, [21](#)
      - serengeti-cluster-chassis, [23](#)
  - simple-fc-disk, [44](#)
  - status
    - namespace command
      - pci-bcm5703c, [37](#)
      - pci-bcm5704c, [38](#)
      - pci-sym53c875, [39](#)
      - pci-sym53c876, [40](#)
      - sample-gcache, [49](#)
      - sample-ma-model, [50](#)
      - sample-ooo-model, [50](#)
      - serengeti-3800-chassis, [16](#)
      - serengeti-4800-chassis, [18](#)
      - serengeti-4810-chassis, [19](#)
      - serengeti-6800-chassis, [21](#)
      - serengeti-cluster-chassis, [23](#)
      - serengeti-cpci4-board, [29](#)
      - serengeti-pci8-board, [28](#)
      - serengeti-sp-cpci6-board, [30](#)
      - serengeti-us-iii-cpu-board, [24](#)
      - serengeti-us-iii-plus-cpu-board, [25](#)
      - serengeti-us-iv-cpu-board, [26](#)
      - serengeti-us-iv-plus-cpu-board, [27](#)
      - simple-fc-disk, [45](#)
      - std-ethernet-link, [41](#)
      - std-graphics-console, [48](#)
      - std-scsi-bus, [42](#)
      - std-scsi-cdrom, [43](#)
      - std-scsi-disk, [43](#)
      - std-server-console, [47](#)
      - std-service-node, [42](#)
      - std-text-console, [46](#)
      - std-text-graphics-console, [48](#)
      - sun-cpci-hme-isp, [31](#)
  - sun-cpci-isp-isp, [31](#)
  - sun-cpci-qlc-qlc, [32](#)
  - sun-pci-ce, [33](#)
  - sun-pci-hme, [33](#)
  - sun-pci-hme-isp, [34](#)
  - sun-pci-pgx64, [35](#)
  - sun-pci-qlc, [36](#)
  - sun-pci-qlc-qlc, [36](#)
  - std-ethernet-link, [40](#)
  - std-graphics-console, [47](#)
  - std-scsi-bus, [42](#)
  - std-scsi-cdrom, [43](#)
  - std-scsi-disk, [42](#)
  - std-server-console, [46](#)
  - std-service-node, [41](#)
  - std-text-console, [45](#)
  - std-text-graphics-console, [48](#)
  - sun-cpci-hme-isp, [30](#)
  - sun-cpci-isp-isp, [31](#)
  - sun-cpci-qlc-qlc, [32](#)
  - sun-pci-ce, [32](#)
  - sun-pci-hme, [33](#)
  - sun-pci-hme-isp, [34](#)
  - sun-pci-pgx64, [35](#)
  - sun-pci-qlc, [35](#)
  - sun-pci-qlc-qlc, [36](#)
- ## T
- top-component, [52](#)



**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>