



SGI® Altix® UV CMC Controller  
Software User's Guide

007-5636-005

---

#### COPYRIGHT

© 2010, 2011, SGI. All rights reserved; provided portions may be copyright in third parties, as indicated elsewhere herein. No permission is granted to copy, distribute, or create derivative works from the contents of this electronic documentation in any manner, in whole or in part, without the prior written permission of SGI.

---

#### LIMITED RIGHTS LEGEND

The software described in this document is “commercial computer software” provided with restricted rights (except as to included open/free source) as specified in the FAR 52.227-19 and/or the DFAR 227.7202, or successive sections. Use beyond license provisions is a violation of worldwide intellectual property laws, treaties and conventions. This document is provided with limited rights as defined in 52.227-14.

---

#### TRADEMARKS AND ATTRIBUTIONS

Altix, NUMALink, SGI, the SGI logo, Silicon Graphics, and Supportfolio are trademarks or registered trademarks of Silicon Graphics International Corp. or its subsidiaries in the United States and other countries.

---

## Record of Revision

<b>Version</b>	<b>Description</b>
001	June 2010 Initial release.
002	June 2010 Added information for SGI Altix UV 100 systems.
003	October 2010 Updated to support the SGI Foundation Software 2.2 release.
004	Febraury 2011 Updated to support SGI Foundation Software 2.3 release.
005	April 2011 Updated to support SGI Foundation Software 2.4 release.



---

## New Features in This Guide

### Major Documentation Changes

- Added information about enabling the `autopower` utility in “Enabling the autopower Utility” on page 21.
- Added information about upgrading to Altix UV BIOS 2.0 release (or later) in “Upgrading System BIOS” on page 34.
- Added information about the `autopower` command in “autopower” on page 40.
- Added information about the `console -t/terse` option in “config” on page 42.
- Updated the escape code information in “console (uvcon)” on page 45.
- Added information about finding the compute node BMC firmware version or the Base IO node BMC firmware version in “version” on page 51.



# Contents

<b>Record of Revision</b>	iii
<b>Figures</b>	ix
<b>About This Guide.</b>	xi
Related Publications	xii
Obtaining Publications	xiii
Conventions	xiii
Reader Comments.	xiv
<b>1. Introducing Altix UV System Control Topology.</b>	<b>1</b>
Altix UV 1000 Overview	1
Altix UV 100 Overview	6
System Management	7
Chassis Management Controller	8
System Control Network.	9

	Determining Rack Numbers . . . . .	11
	Accessories . . . . .	13
	Altix UV System Controller Software . . . . .	14
<b>2.</b>	<b>Using the Altix UV CMC Software Commands . . . . .</b>	<b>17</b>
	Connecting to the UV System Controller Network . . . . .	17
	Power on and Booting an Altix UV System from Complete Power Off . . . . .	18
	Power off an Altix UV System . . . . .	20
	Power NMI to Drop into KDB . . . . .	21
	Enabling the <code>autopower</code> Utility. . . . .	21
	Viewing Your System Configuration . . . . .	23
	Using the <code>hwcfg</code> Command . . . . .	24
	Finding the CMC IP Address. . . . .	27
	System Partitioning . . . . .	28
	Upgrading System BIOS . . . . .	34
	Hyper-Threading on Altix UV 100 or Altix UV 1000 Systems . . . . .	35
<b>3.</b>	<b>Altix UV CMC Software Commands . . . . .</b>	<b>37</b>
	CMC Command Targets . . . . .	37
	<code>auth.</code> . . . . .	39
	<code>autopower.</code> . . . . .	40
	<code>bios.</code> . . . . .	40
	<code>bmc</code> . . . . .	41
	<code>cmc</code> . . . . .	42
	<code>config.</code> . . . . .	42
	<code>console (uvcon).</code> . . . . .	45
	<code>flashbios.</code> . . . . .	46
	<code>hwcfg</code> . . . . .	47
	<code>log</code> . . . . .	49
	<code>power</code> . . . . .	49
	<code>sensor.</code> . . . . .	51
	<code>uvcon</code> . . . . .	51
	<code>version</code> . . . . .	51
	<b>Index . . . . .</b>	<b>53</b>

---

# Figures

<b>Figure 1-1</b>	Individual Rack Unit . . . . .	3
<b>Figure 1-2</b>	Basic System Building Blocks for Altix UV 1000 Systems . . . . .	4
<b>Figure 1-3</b>	SGI Altix UV System Rack. . . . .	5
<b>Figure 1-4</b>	SGI Altix UV 100 IRU Front View . . . . .	7
<b>Figure 1-5</b>	Chassis Manager Controller . . . . .	9
<b>Figure 1-6</b>	CMC Ethernet Ports on SGI Altix UV 1000 Systems . . . . .	10
<b>Figure 1-7</b>	CMC Ethernet Ports on SGI Altix UV 100 Systems . . . . .	11



---

## About This Guide

This guide describes how to use the controller commands on your chassis manager controller (CMC) to monitor and manage the following systems:

- SGI Altix UV 100 systems
- SGI Altix UV 1000 systems

SGI Management Center (SMC) software running on the system management node (SMN) provides a robust graphical interface for system configuration, operation, and monitoring. This manual describes commands that can be used on systems without an SMN or not running the SMC. For more information on the SMC, see *SGI Management Center System Administrator's Guide*.

The *SGI Altix UV System Management Node Administrator's Guide* describes the system management node (SMN) for SGI Altix UV 1000 and SGI Altix UV 100 series systems. It provides information on how to install, configure, and use software on the SMN to manage and monitor SGI Altix UV systems.

---

**Note:** The UV controller commands described in this manual do not apply to SGI Altix UV 10 systems. For information on the SGI Altix UV 10 system, see the *SGI Altix UV 10 System User's Guide*.

---

The following topics are covered in this guide:

- Chapter 1, “Introducing Altix UV System Control Topology”
- Chapter 2, “Using the Altix UV CMC Software Commands”
- Chapter 3, “Altix UV CMC Software Commands”

## Related Publications

The following publications contain additional information that may be helpful:

- *SGI Altix UV 10 System User's Guide* provides an overview of the Altix UV 10 system components, and it describes how to set up and operate this system. It also describes the standard procedures for powering up and powering down the system, basic troubleshooting information, and it includes important safety and regulatory specifications.
- *SGI Altix UV 100 System User's Guide* provides an overview of the Altix UV 100 system components, and it describes how to set up and operate this system. It also describes the standard procedures for powering up and powering down the system, basic troubleshooting information, and it includes important safety and regulatory specifications.
- *SGI Altix UV 1000 System User's Guide* provides an overview of the Altix UV 1000 system components, and it describes how to set up and operate this system. It also describes the standard procedures for powering up and powering down the system, basic troubleshooting information, and it includes important safety and regulatory specifications.
- *SGI Performance Suite 1.2 Start Here* provides information about the SGI Performance Suite 1.1 release including information about major new features, software installation, and product support.
- *SGI Altix UV Systems Linux Configuration and Operations Guide* provides information for people who manage the operation of SGI UV systems running SGI Performance Suite software. It explains how to perform general system configuration and operations under the Linux operating system used with SGI UV systems.
- *SGI Altix UV System Management Node Administrator's Guide* describes the system management node (SMN) for SGI Altix UV 1000 and SGI Altix UV 100 series systems. It provides information on how to install, configure, and use software on the SMN to manage and monitor SGI Altix UV systems.
- *SGI Management Center Installation and Configuration* is intended for system administrators. It describes how to install and configure the SGI Management Center. A companion manual, *SGI Management Center System Administrator's Guide*, describes general cluster administration.
- *SGI Management Center System Administrator's Guide* describes how you can monitor and control a cluster using the SGI Management Center. A companion manual, *SGI Management Center Installation and Configuration Guide*, describes installing and configuring the SGI Management Center.

## Obtaining Publications

You can obtain SGI documentation in the following ways:

- See the SGI Technical Publications Library at <http://docs.sgi.com>. Various formats are available. This library contains the most recent and most comprehensive set of online books, release notes, man pages, and other information.
- You can view release notes on your system by accessing the `README.txt` file for the product. This is usually located in the `/usr/share/doc/productname` directory, although file locations may vary.
- You can view man pages by typing `man title` at a command line.

## Conventions

The following conventions are used throughout this publication:

<b>Convention</b>	<b>Meaning</b>
<code>command</code>	This fixed-space font denotes literal items such as commands, files, routines, path names, signals, messages, and programming language structures.
<i>variable</i>	Italic typeface denotes variable entries and words or concepts being defined.
<b>user input</b>	This bold, fixed-space font denotes literal items that the user enters in interactive sessions. (Output is shown in nonbold, fixed-space font.)
[]	Brackets enclose optional portions of a command or directive line.
...	Ellipses indicate that a preceding element can be repeated.
manpage(x)	Man page section identifiers appear in parentheses after man page names.
<b>GUI element</b>	This font denotes the names of graphical user interface (GUI) elements such as windows, screens, dialog boxes, menus, toolbars, icons, buttons, boxes, fields, and lists.

## Reader Comments

If you have comments about the technical accuracy, content, or organization of this document, contact SGI. Be sure to include the title and document number of the manual with your comments. (Online, the document number is located in the front matter of the manual. In printed manuals, the document number is located at the bottom of each page.)

You can contact SGI in any of the following ways:

- Send e-mail to the following address:  
techpubs@sgi.com
- Contact your customer service representative and ask that an incident be filed in the SGI incident tracking system.
- Send mail to the following address:  
Technical Publications  
SGI  
46600 Landing Parkway  
Fremont, CA 94538

SGI values your comments and will respond to them promptly.

# Introducing Altix UV System Control Topology

This manual describes controller software commands on SGI Altix UV 100 and SGI Altix UV 1000 systems.

---

**Note:** This manual does not apply to SGI Altix UV 10 systems. For information, see the *SGI Altix UV 10 System User's Guide*.

---

## Altix UV 1000 Overview

The SGI Altix UV 1000 system is a blade-based, cache-coherent non-uniform memory access (ccNUMA), computer system that is based on the Intel Xeon 7500 series processor. The UV 1000 system scales, as follows:

- From 32 to 2048 threads in a single system image (SSI)
- A maximum of 2048 processor cores with hyper-threading turned off
- A maximum of 4096 processor threads (2048 processor cores) with hyper-threading turned on

---

**Note:** Each processor core supports two threads. A processor with hyper-threading enabled is treated by the operating system as two processors instead of one. This means that only one processor is physically present but the operating system sees two logical processors, and shares the workload between them. At initial release, the maximum SSI supported by the Linux operating system is 2048.

---

The main component is an 18U-high individual rack unit (IRU) shown in Figure 1-1 that supports 16 compute blades and is configurable to support multiple topology options.

The compute blades in the IRU are interconnected using NUMALink 5 technology. NUMALink 5 has a peak aggregate bi-directional bandwidth of 15 GB/s. Multiple IRUs are also interconnected with NUMALink 5 technology.

A maximum of two IRUs can be placed into a custom 42U rack as shown in Figure 1-2. Each rack supports a maximum of 512 processor cores; therefore, the largest SSI system requires four racks. A maximum of 128 four rack cells can be interconnected to create a 512 rack system (256K processor cores).

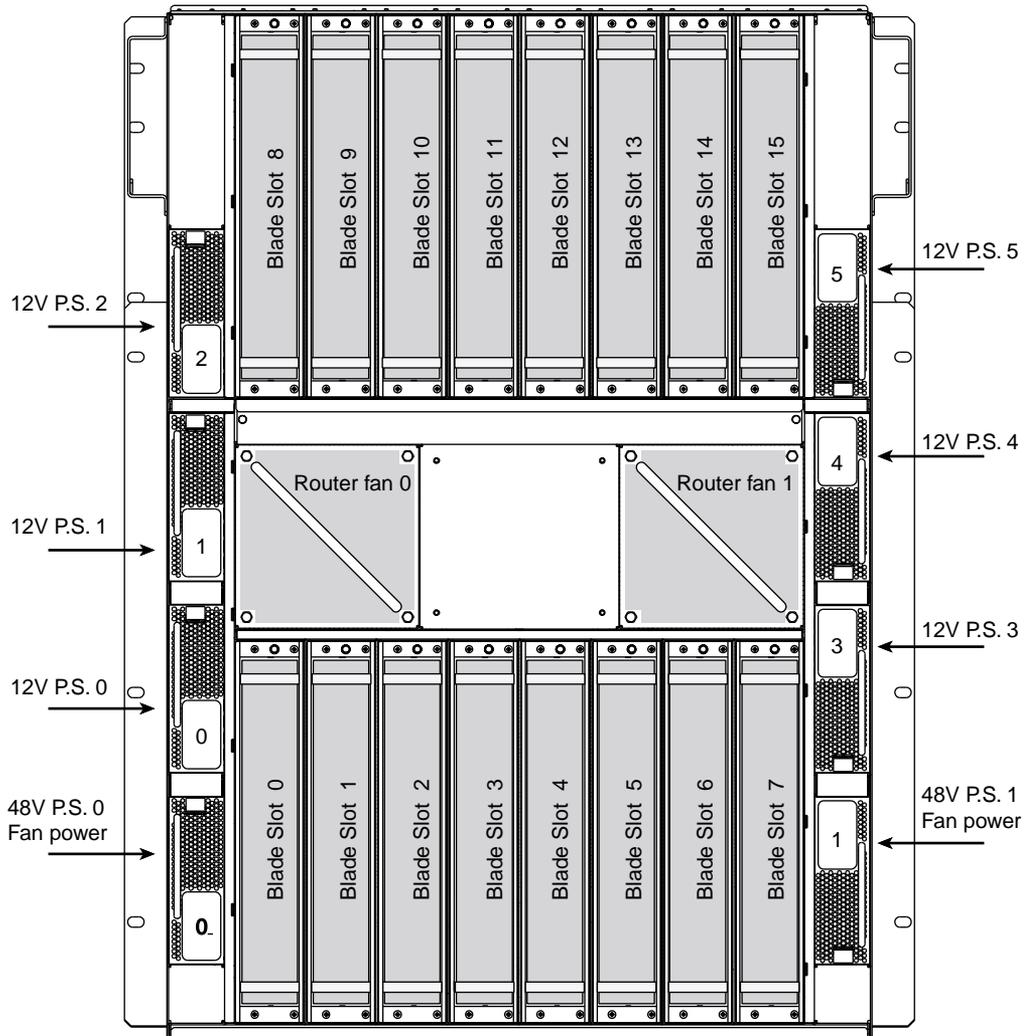
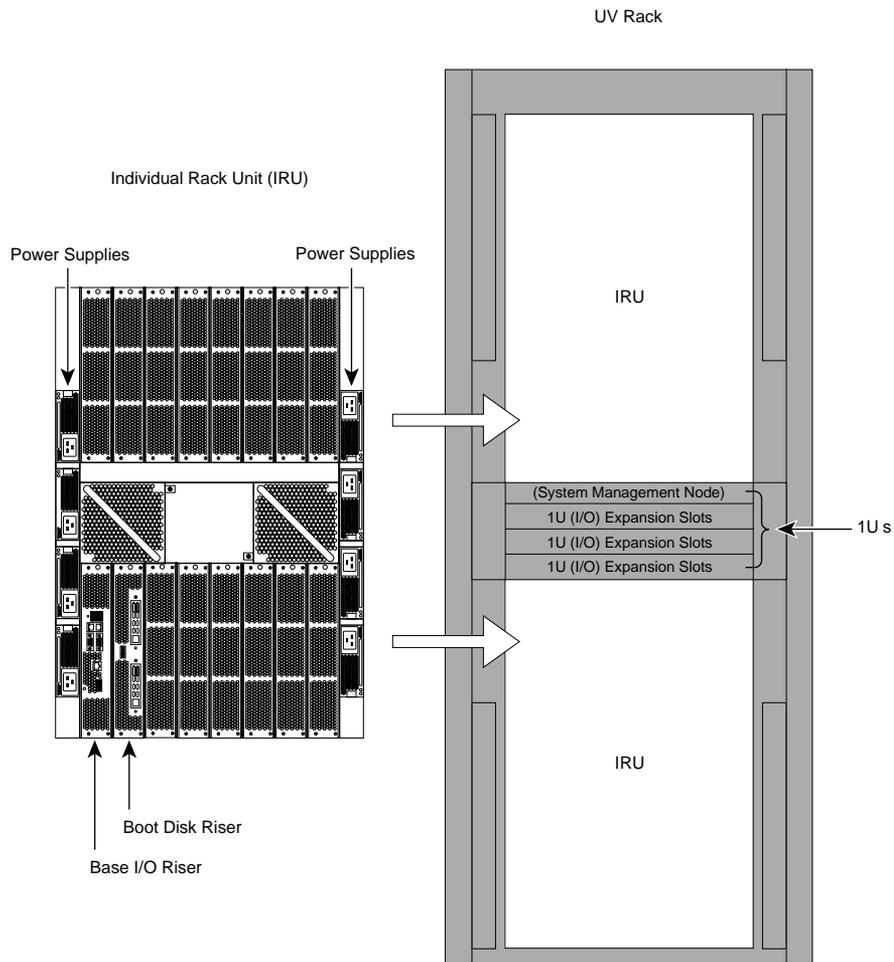
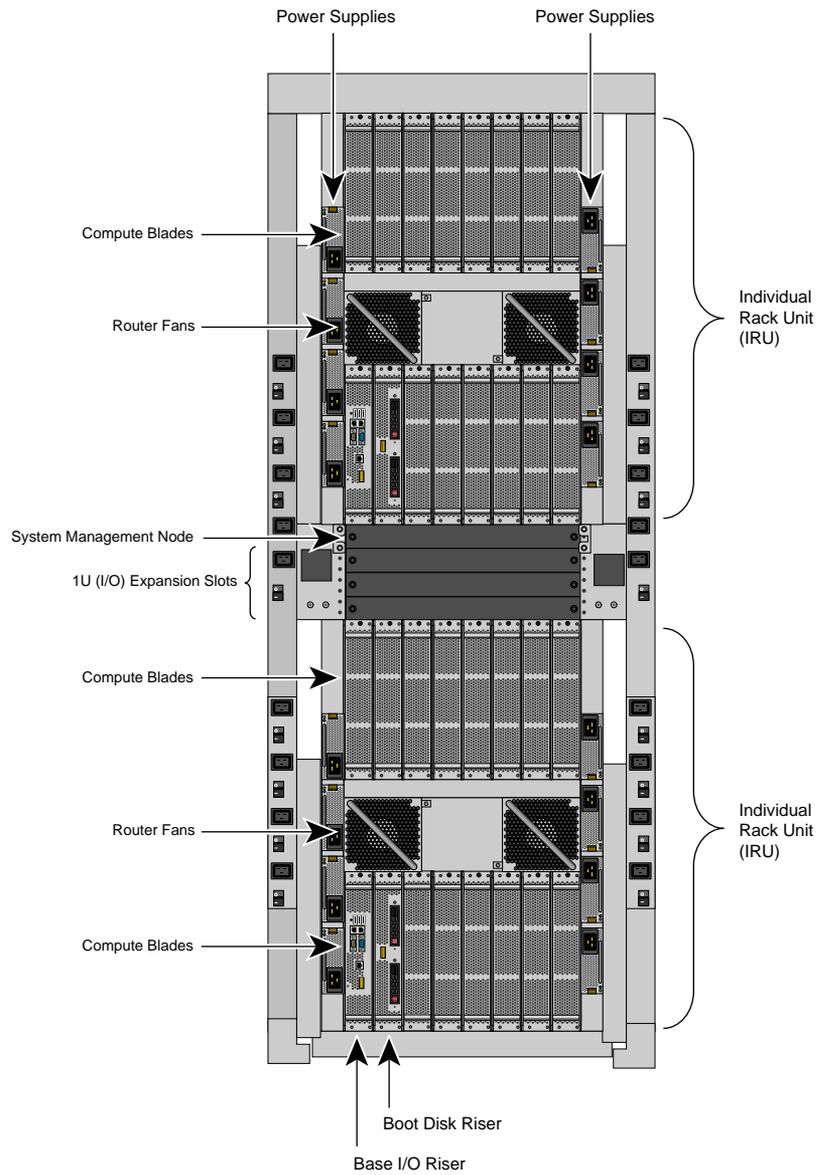


Figure 1-1 Individual Rack Unit



**Figure 1-2** Basic System Building Blocks for Altix UV 1000 Systems

The Altix UV system supports direct attach I/O on the compute blade. The compute blade is designed to host one of four different I/O riser cards. Various PCI express based I/O components are supported. Figure 1-3 shows a full SGI Altix UV system rack.



**Figure 1-3** SGI Altix UV System Rack

For a detailed hardware description, see the *SGI Altix UV 1000 Systems User's Guide*. Figure 1-3 on page 5.

The SGI hardware manuals contain detailed descriptions of Altix system architecture. For a list of these manuals, see “Related Publications” on page xii.

---

**Note:** Online and postscript versions of SGI documentation is available at SGI Technical Publications Library at <http://docs.sgi.com>.

---

## Altix UV 100 Overview

The SGI Altix UV 100 system is a small, blade-based, cache-coherent, non-uniform memory access (ccNUMA), computer system that is based on the Intel Xeon 7500 series processor. The SGI Altix UV 100 system scales, as follows:

A maximum of 768 processor cores

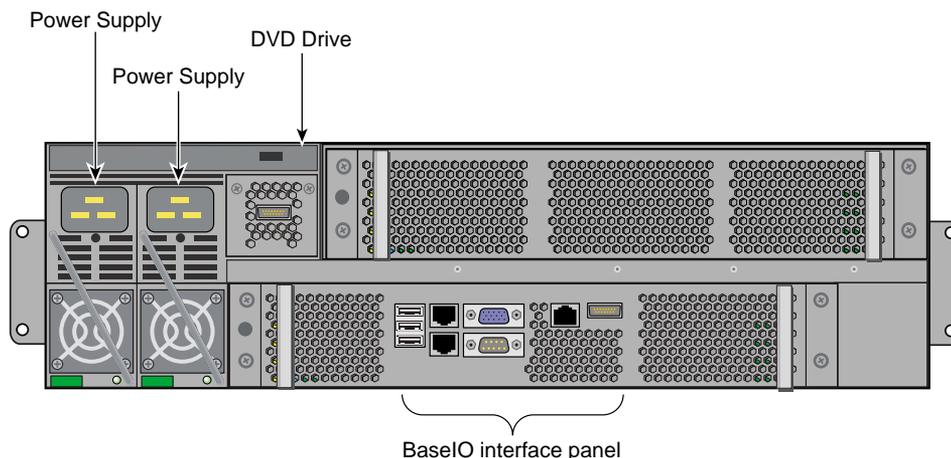
From 16 to 1536 threads in a single system image (SSI)

---

**Note:** Each processor core supports two threads.

---

The main component is a 3U-high IRU, shown in Figure 1-4, that supports two compute blades and is configurable to support multiple topology options.



**Figure 1-4** SGI Altix UV 100 IRU Front View

The two compute blades in the IRU are interconnected using NUMAlink 5 technology. NUMAlink 5 has a peak aggregate bi-directional bandwidth of 15 GB/s. Multiple IRUs are also interconnected with NUMAlink 5 technology.

A maximum of twelve IRUs can be placed into a standard 42U 19" custom tall rack. Each rack supports a maximum of 384 processor cores.

The Altix UV system supports direct attach I/O on the compute blade. The compute blade is designed to host one of four different I/O riser cards. Various PCI express based I/O components are supported. For a detailed hardware description, see the *SGI Altix UV 100 Systems User's Guide*.

## System Management

The system management provides a single control point for system power up, initialization, booting and maintenance. System management on an SGI Altix UV 1000 consists of three levels. The first level of system management is the board management controllers (BMCs) on the node boards. The second level is the chassis management controllers (CMC) in the rear of the IRU. The third level is the system management node (SMN). The SMN is required on SGI Altix UV 1000 series systems. It is not required for the SGI Altix UV 100 series systems.

---

**Important:** The UV 1000 and UV 100 system control network is a private, closed network. It is not to be reconfigured in any way different from the standard UV installation, nor is it to be directly connected to any other network. The UV system control network does not accommodate additional network traffic, routing, address naming other than its own schema, and DHCP controls other than its own configuration. The system control network also is not security hardened, nor is it tolerant of heavy network traffic, and is vulnerable to Denial of Service attacks.

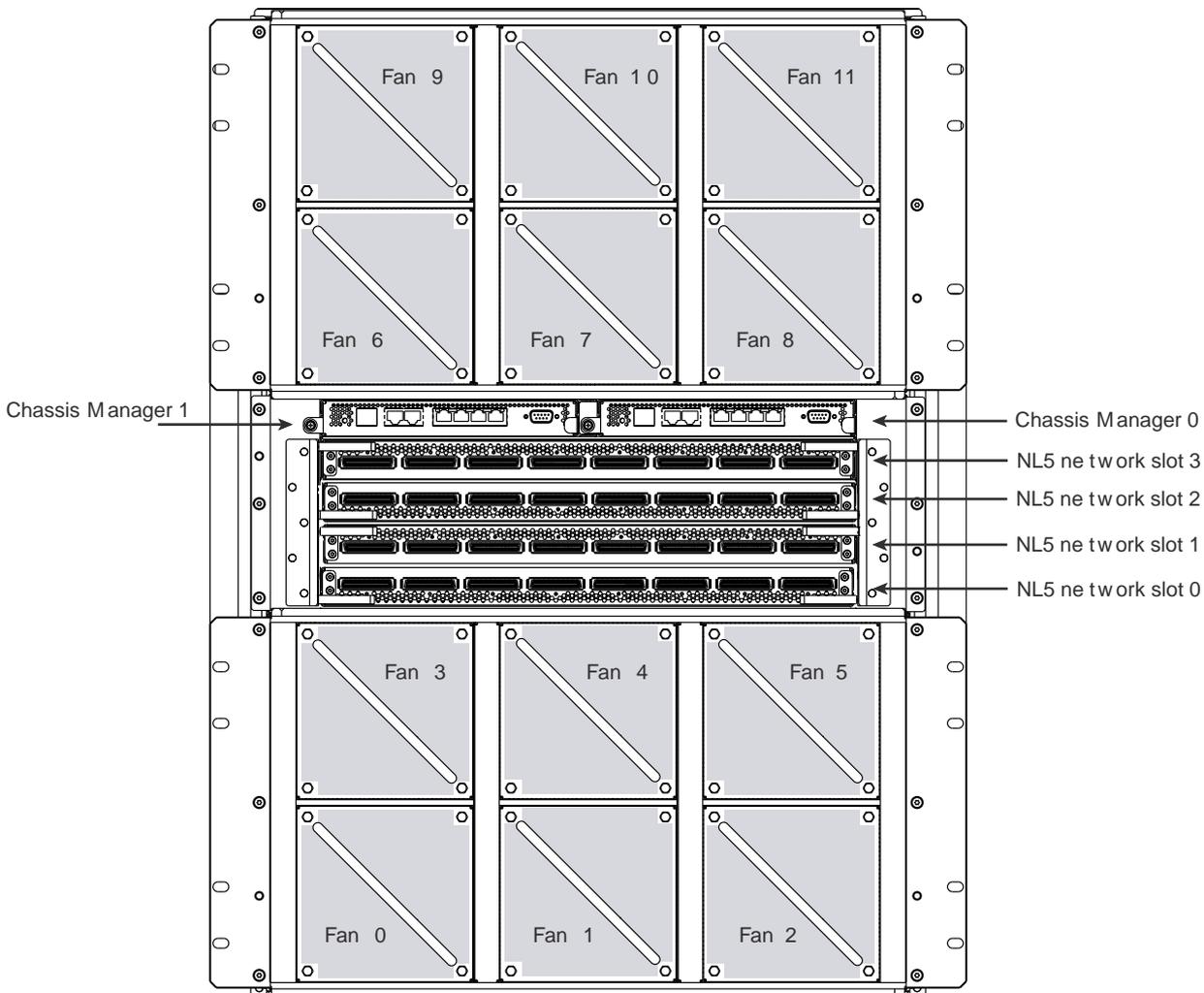
---

The System Management Node acts as a gateway between the UV system control network and any other networks.

SGI Management Center (SMC) software running on the system management node (SMN) provides a robust graphical interface for system configuration, operation, and monitoring. This manual describes commands that can be used on systems without an SMN or not running the SMC. For more information, see *SGI Management Center System Administrator's Guide*.

## Chassis Management Controller

The chassis management controller (CMC) in the rear of the IRU, as shown in Figure 1-5, and Figure 1-6, supports powering up and down of the compute blades and environmental monitoring of all units within the IRU. The CMC sends operational requests to the baseboard manager controller (BMC) on each compute node. The CMC provides data collected from the compute nodes within the IRU to the system management node upon request. The CMC blade on the right side of the IRU is the primary CMC. A secondary CMC is currently not supported.



**Figure 1-5** Chassis Manager Controller

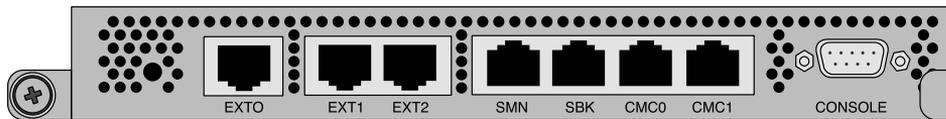
## System Control Network

The chassis management controller (CMC) for SGI Altix UV 1000 systems has seven RJ45 Ethernet ports, as shown in Figure 1-6.

The Ethernet ports are used, as follows:

- **SMN** - the system management node port is used to connect to the SMN.
- **SBK** - Each 16 rack group is called a super block. A building block is four racks. A super block is four building blocks. The SBK connects one super block to another super block.
- **CMC0** and **CMC1** - these two ports are used to interconnect multiple IRUs within a building block together.
- **EXT0, EXT1, EXT2** - connects to external devices such as I/O chassis and smart PDUs.

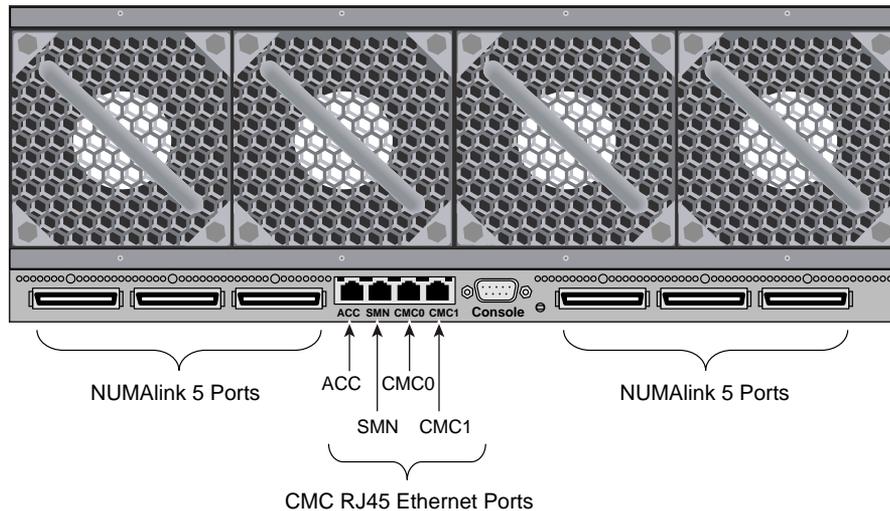
**CONSOLE** - the console connection supports a serial channel connection directly to the CMC for system maintenance.



**Figure 1-6** CMC Ethernet Ports on SGI Altix UV 1000 Systems

For information on finding the CMC IP address and hostname, see “Finding the CMC IP Address” on page 27.

The chassis management controller (CMC) for SGI Altix UV 100 systems is a board assembly integrated into the IRU and has four RJ45 Ethernet ports, as shown in Figure 1-4.



**Figure 1-7** CMC Ethernet Ports on SGI Altix UV 100 Systems

The Ethernet ports are used, as follows:

- **ACC** - the accessory is used to connect miscellaneous devices to the CMC network, for example smart power distribution units (PDUs).
- **SMN** - the system management node port is used to connect to the SMN.
- **CMC0** and **CMC1** - these two ports are used to interconnect multiple IRUs together to form a string topology.

**CONSOLE** - the console connection supports a serial channel connection directly to the CMC for system maintenance.

## Determining Rack Numbers

The system controller network has strict requirements for rack numbering. The requirements minimize the amount of information that must be manually configured for each CMC when it is plugged into an IRU. Currently, only the rack and u-position of the IRU must be set. The u-position is the physical location of the IRU in the rack. The rack and u-position values are found in the `/etc/sysconfig/module_id` file. Besides

uniquely identifying the physical location of the CMCs, the values are used to generate several IP address for the various VLANs on the CMC and are used by any software interacting with the system controller network to target operations.

For large Altix UV 1000 configurations, a building block consists of four racks with two IRUs in each rack with the CMCs in those IRUs interconnected via their CMC0 and CMC1 jacks. In order for racks to be considered part of the same building block, their rack numbers must be consecutive and satisfy the following equation:

$$(\text{rack} - 1) \text{ MOD } 4 = 0, 1, 2 \text{ or } 3$$

or

$$(\text{rack} - 1) \text{ DIV } 4 = \text{the same value for all racks in the building block}$$

For example, a system with four racks numbered 1, 2, 3, and 4 has one building block. Similarly, a system with four racks number 9, 10, 11, and 12 has one building block.

A system with racks numbered 10, 11, 12, 13 would have to two building blocks with 10, 11 and 12 in one building block; 13 is in a second building block. The system controller network must be cabled appropriately for each configuration.

A super block (SBK) consists four building blocks. Two primary CMCs in each building block are used to interconnect the building blocks via their SBK jacks. For racks to be considered part of the same SBK their rack numbers must be consecutive and satisfy the following equation:

$$(\text{rack} - 1) \text{ MOD } 16 = 0,1,2,\dots 15$$

or

$$(\text{rack} - 1) \text{ DIV } 16 = \text{the same value for all racks in the SBK}$$

In summary, a single SBK can support up to four building blocks, or in other words, 16 racks.

## Accessories

Accessories are third party devices that connect to the system controller network. On the Altix UV 1000 chassis CMC, these accessories are connected to the EXT0, EXT1, and EXT2 jacks. On the Altix UV 100 chassis CMC, these accessories are connected to the ACC jack. If there are more accessories than available jacks an external switch can be used.

Currently, two accessories are supported, as follows:

- Magma PCIE Expansion chassis
- Eaton ePDU

Accessories connected to the CMC's accessory jacks should be configured to get their IP address via dynamic host configuration protocol (DHCP). The DHCP server on the CMC will assign IP address in the 10.<rack>.<upos>.100 to 10.<rack>.<upos>.199 range where rack and upos are the rack and upos of the CMC. This is the CMC's VACC VLAN.

The CMC searches for accessories and using Simple Network Management Protocol (SNMP) attempts to determine the accessory type. Specifically, the CMC queries the `sysName.0` Object Identifier (OID) and looks for: "Monitored ePDU" is assumed to be an Eaton ePDU. "Magma Chassis" is assumed to be a Magma PCIE expansion chassis.

A physical location can also be configured into each accessory using the `sysLocation.0` OID. Each accessory must be configured one at a time:

- Connected by a cable from the accessory's SNMP port to an open accessory jack on a CMC
- The CMC `config -v` command will initially show the accessory, as follows:

```
uvl4-cmc CMC:rlilc> config -v
CMCs:          1
               r001i01c UV1000
BMCs:          4
               r001i01b00 IP93-BASEIO
               r001i01b01 IP93-DISK
               r001i01b02 IP93
               r001i01b03 IP93
Partitions:    1
               partition000 BMCs:    4
Accessories:   1
               undefined          10.1.1.101 (Magma PCIE Expansion)
```

- Use the CMC `config -acc` command to set a location, as follows:

```
uvl4-cmc CMC:rlilc> config --acc 10.1.1.101@1.30
==== r001i01c (PRI) ====
10.1.1.101 (Magma Chassis) configured as r001u30io
```

Verify using the config -v command, as follows:

```
uvl4-cmc CMC:rlilc> config -v
CMCs:          1
               r001i01c UV1000
BMCs:          4
               r001i01b00 IP93-BASEIO
               r001i01b01 IP93-DISK
               r001i01b02 IP93
               r001i01b03 IP93
Partitions:    1
               partition000 BMCs: 4
Accessories:   1
               r001u30io          10.1.1.101 (Magma PCIE Expansion)
```

The location description formats of the two types of accessories currently supported are, as follows:

```
Smart PDUs: location description format is r<rack>pdu[0|1]
             Eaton ePDU
```

```
I/O: location description format is r<rack>u<uposition>io
      Magma PCIE Expansion
```

Power control only affects I/O type accessories. When the IRU chassis power is turned on, off, or cycled, the accessories are treated the same.

The CMC power on|off|cycle commands will accept a noio option to exclude the I/O accessories from the power operation.

## Altix UV System Controller Software

The controller is designed to manage and monitor the individual blades in SGI Altix UV systems. Depending on your system configuration, you can monitor and operate the system from the system management node (SMN) or on smaller systems, such as, the Altix UV 100 from the CMC itself. UV 1000 systems up to 16 racks (four building blocks, also called one super block) can also be controlled and monitored from a CMC in the system.

The following list summarizes the control and monitoring functions that the CMC performs. Many of the controller functions are common across both IRU and routers; however, some functions are specific to the type of enclosure.

- Controls and monitors IRU and router fan speeds
- Reads system identification (ID) PROMs
- Monitors voltage levels and reports failures
- Monitors and controls warning LEDs on the enclosure
- Provides the ability to create multiple system partitions (single system image) running their own operating system.
- Provides ability to flash system BIOS



## Using the Altix UV CMC Software Commands

This chapter describes how to use the CMC controllers to power on, manage, and monitor an SGI Altix UV 1000 or UV 100 system in the following sections:

- “Connecting to the UV System Controller Network” on page 17
- “Power on and Booting an Altix UV System from Complete Power Off” on page 18
- “Power off an Altix UV System” on page 20
- “Power NMI to Drop into KDB” on page 21
- “Enabling the autopower Utility” on page 21
- “Viewing Your System Configuration” on page 23
- “Using the hwcfg Command” on page 24
- “Finding the CMC IP Address” on page 27
- “System Partitioning” on page 28
- “Upgrading System BIOS” on page 34
- “Hyper-Threading on Altix UV 100 or Altix UV 1000 Systems” on page 35

### Connecting to the UV System Controller Network

The console type and how these console types are connected to the Altix UV 1000 systems is determined by what console option is chosen. Establish either a serial connection or network/Ethernet connection to the CMC.

#### **Establish a serial connection**

If you have an Altix UV 1000 system and wish to use a serially-connected "dumb terminal", you can connect the terminal via a serial cable to the (DB-9) RS-232-style console port connector on the CMC board of the IRU.

The terminal should be set to the following functional modes:

- pin 2 - receive
- pin 3 - transmit
- pin 5 - ground
- Baud: 115200
- Data bits: 8
- Parity: no
- Stop bits: 1
- No flow control

Note that a serial console is generally connected to the first (bottom) IRU in any single rack configuration. For more information, see the “Console Hardware Requirements” section in the *SGI Altix UV 1000 System User’s Guide*.

**Establish a Network/Ethernet connection** (see SBK port, EXT port, and SMN port in Figure 1-6)

CMCs have their rack and u position set at the factory. The CMC will assign itself IP addresses, as follows:

SBK 172.17.<rack>.<slot>

EXT 10.<rack>.<slot>.1

On the system management node (SMN) port, the CMC is configured to request an IP address via dynamic host configuration protocol (DHCP).

Either connection, serial or network, will present a login prompt. For more information, see the “Levels of System Control” section in the *SGI Altix UV 1000 System User’s Guide*.

## Power on and Booting an Altix UV System from Complete Power Off

To boot an SGI Altix UV 1000 or UV 100 system from complete power off, perform the following steps:

1. Make sure the power breakers are on.
2. Establish a serial connection to the **CONSOLE** on the CMC (see Figure 1-6 on page 10). See “Connecting to the UV System Controller Network” on page 17 or skip to the next step.
3. Establish a network connection to the CMC. “Connecting to the UV System Controller Network” on page 17. Use the `ssh` command to connect to the CMC, similar to the following example:

---

**Note:** This is only valid if your PC is connected to the CMC (via the network connection) has its `/etc/hosts` file setup to include the CMCs.

---

```
ssh root@hostname-cmc
SGI Chassis Manager Controller, Firmware Rev. 0.0.22
```

```
CMC:r111c>
```

Typically, the default password set out of the factory is `root`. The CMC prompt appears. `CMC:r111c` refers to rack 1, IRU 1, CMC (see Figure 1-5 on page 9 and Figure 1-6 on page 10)

If the host name is **not** set up in the PC/workstation’s hosts file, you can simply use the IP address of the CMC, as follows:

```
ssh root@<IP-ADDRESS>
```

4. Power up your Altix UV system using the `power on` command, as follows:

```
CMC:r111c> power on
```

---

**Note:** You can open a second window on the CMC, `ssh root@hostname-cmc` and use the console command (same as `uvcon`) to open a console and watch the system power on.

---

5. Open a second console to the CMC using the `console` command to see the system power on, as follows:

```
ssh root@hostname-cmc
SGI Chassis Manager Controller, Firmware Rev. 0.0.22
```

```
CMC:r111c> console
console: attempting connection to localhost...
console: connection to SMN/CMC (localhost) established.
```

```
console: requesting baseio console access at r001i01b00...
console: tty mode enabled, use 'CTRL-]' 'q' to exit
console: console access established
console: CMC <--> BASEIO connection active
*****
*****  START OF CACHED CONSOLE OUTPUT  *****
*****
***** [20100512.143541] BMC r001i01b10: Cold Reset via NL
broadcast reset
***** [20100512.143541] BMC r001i01b07: Cold Reset via NL
broadcast reset
***** [20100512.143540] BMC r001i01b08: Cold Reset via NL
broadcast reset
***** [20100512.143540] BMC r001i01b12: Cold Reset via NL
broadcast reset
***** [20100512.143541] BMC r001i01b14: Cold Reset via NL
broadcast reset
***** [20100512.143541] BMC r001i01b04: Cold Reset via NL
.....
```

---

**Note:** Use CTRL-] q to exit the console.

---

6. Depending upon the size of your system, it can take 5 to 10 minutes for the Altix UV system to power on. When the **shell>** prompt appears, enter **fs0**, as follows:

```
shell> fs0
```

7. At the **fs0** prompt, enter **boot**, as follows:

```
fs0> boot
```

ELILO Linux Boot loader is called and various SGI configuration scripts are run and the SUSE Linux Enterprise Server 11 SP1 installation program appears.

## Power off an Altix UV System

To power down the Altix UV stem, use the `power off` command, as follows:

```
CMC:r1i1c> power off
==== r001i01c (PRI) ====
```

You can use the `power status` command, to check the power status of your system

```
CMC:r1i1c> power status
==== r001i01c (PRI) ====
on: 0, off: 32, unknown: 0, disabled: 0
```

## Power NMI to Drop into KDB

To send a nonmaskable interrupt (NMI) signal from the power command to the CMC to drop into the kernel debugger (KDB), use the power nmi command, as follows:

```
CMC:r1i1c> power nmi
```

```
Entering kdb (current=0xffff8aa3fella040, pid 0) on processor 7 due to
NonMaskable Interrupt @ 0xffffffff8100ad42
```

```

r15 = 0x0000000000000000      r14 = 0x0000000000000000
r13 = 0x0000000000000000      r12 = 0x0000000000000000
bp  = 0xffffffff81927380      bx  = 0xffff8ac1ff11dfd8
r11 = 0xffffffff8101a2c0      r10 = 0xffff88000bee7fd18
r9   = 0x00000000ffffffff      r8  = 0x0000000000000000
ax   = 0x0000000000000000      cx  = 0x0000000000000000
dx   = 0x0000000000000000      si  = 0xffff8ac1ff11dfd8
di   = 0xffffffff81a2b308      orig_ax = 0xffffffffffffffff
ip   = 0xffffffff8100ad42      cs  = 0x0000000000000010
flags = 0x0000000000000246      sp  = 0xffff88000bee7ff0
ss   = 0x0000000000000018      &regs = 0xffff88000bee7f58
[7]kdb>
```

## Enabling the autopower Utility

---

**Note:** To enable the autopower utility, you need to be running CMC 1.2.7 or later firmware.

---

Use the version command from the CMC prompt to determine the CMC firmware version, as follows:

```
uv44-cmc CMC:r001i01c> version
SGI Chassis Manager Controller, Firmware Rev. 1.3.16 [Bootloader
0.6.0]
```

Use the `autopower` command without any arguments to determine if autopower is enabled or disabled on your system, as follows:

```
uv44-cmc CMC:r001i01c> autopower
==== r001i01c (PRI) ====
auto-power on is disabled
```

Use the `autopower -e` command to enable autopower, as follows:

```
uv44-cmc CMC:r001i01c> autopower -e
==== r001i01c (PRI) ====
auto-power on enabled (120 second delay)
```

Use the `autopower -d` command to disable autopower, as follows:

```
uv44-cmc CMC:r001i01c> autopower -d
==== r001i01c (PRI) ====
auto-power on disabled
```

For a usage statement, perform the following:

```
uv44-cmc CMC:r001i01c> autopower --h

usage: autopower [-edc] [-t <seconds>] [--help]
-e, --reset          enable auto-power on
-t, --time=<seconds> time (seconds) to delay auto-power on
-d, --disable        disable auto-power on
-c, --cancel         cancel auto-power on
--help              display this help and exit
```

Flashing the CMC 1.2.7 or later firmware only adds support for autopower on, it does not enable it. To enable it, an `autopower -e` command must be issued at the CMC prompt once the system controller configuration has stabilized.

The default delay before a `power on` command is issued after auxiliary (aux) power is supplied to CMCs and BMCs is two minutes. This allows for the system controller configuration to stabilize (complete CMC and BMC detection).

Only the lowest numbered (rack and uposition) CMC will initiate the power on sequence. Because this is a CMC base feature, the maximum system configuration supported is 16 racks or less (both UV100 and UV1000).

For more information on using the `autopower` command, see “`autopower`” on page 40.

## Viewing Your System Configuration

To view your system configuration, use the `config -v` command, as follows:

```
CMC:rlilc> config -v
```

```
CMCs:                2
                    r001i01c UV1000
                    r001i02c UV1000

BMCs:                32
                    r001i01b00 IP93-BASEIO
                    r001i01b01 IP93-DISK
                    r001i01b02 IP93-EXTPCIE
                    r001i01b03 IP93-EXTPCIE
                    r001i01b04 IP93
                    r001i01b05 IP93
                    r001i01b06 IP93
                    r001i01b07 IP93
                    r001i01b08 IP93
                    r001i01b09 IP93
                    r001i01b10 IP93
                    r001i01b11 IP93
                    r001i01b12 IP93
                    r001i01b13 IP93
                    r001i01b14 IP93
                    r001i01b15 IP93
                    r001i02b00 IP93-BASEIO
                    r001i02b01 IP93-EXTPCIE
                    r001i02b02 IP93-DISK
                    r001i02b03 IP93-EXTPCIE
                    r001i02b04 IP93-EXTPCIE
                    r001i02b05 IP93-EXTPCIE
                    r001i02b06 IP93-EXTPCIE
                    r001i02b07 IP93-EXTPCIE
                    r001i02b08 IP93-INTPCIE
                    r001i02b09 IP93-INTPCIE
                    r001i02b10 IP93-INTPCIE
                    r001i02b11 IP93-INTPCIE
                    r001i02b12 IP93-INTPCIE
                    r001i02b13 IP93-INTPCIE
                    r001i02b14 IP93-INTPCIE
                    r001i02b15 IP93-INTPCIE
```

```
Partitions:         1
```

```
partition000 BMCs: 32
```

`r001i01b00` refers to rack 0, IRU 1, and blade 0. For a view of the physical layout of an IRU, see Figure 1-1 on page 3, Figure 1-2 on page 4, and Figure 1-3 on page 5.

## Using the `hwcfg` Command

The `hwcfg` command allows you to set hardware configuration overrides. Many of the hardware overrides available on early Altix UV systems are no longer necessary due to firmware enhancements and hardware changes. Most of these overrides are still available but have been “hidden” and require the `-h` flag to make them visible.

To see a list of current override settings, use the “`hwcfg`” command. This will show all overrides set on any blades in the system. If any overrides are set on some blades and not on others, the output will show a count of blades where the override is set.

To see a list of blades where each override is set, use the `hwcfg -v` command.

To see individual list of blades and their overrides, use `hwcfg -vv` command.

To set one or more overrides, use `hwcfg <name>=<value>` command. For example,

```
hwcfg DEBUG_SW=0x4
```

To clear overrides, use `hwcfg -c` command.

- To clear all overrides, use the `hwcfg -c -a` command.
- To clear one or more variables, use `hwcfg -c <name> [...<name>]`.

Multiple `<name>=<value>` pairs can be set in one command

You can show a list of `hwcfg` variables available, as follows:

```
uv44-cmc CMC:rlilc> hwcfg --list
==== 4/4 BMC(s) ====
SOCKET_DISABLE=yes|no|<socket bitmask>
    Socket 0 disable

PARTITION=<numeric value 0-65535>
    Partition number for this blade
```

```
SMT_ENABLE=yes|no
    SMT (HyperThread) enable

MAX_CORES=<numeric value 0-255, 0=no limit>
    Maximum number of cores allowed (per node)

BLADE_DISABLE=yes|no
    Disable this blade

IORISER_DISABLE=yes|no
    Disable the I/O riser on this blade

ICH_DISABLE=yes|no
    Disable ICH10 on this BaseIO

DEBUG_SW=<32-bit value>
    Software debug switches (see "hwcfg --help DEBUG_SW" for details)

HUB_CORE_SPEED=320|367|375|383|400
    Clock frequency of the HUB

NL5_ENABLE=yes|no|<NL5 port bitmask>
    Enable NL5 links

NL5_RATE=1.25|2.5|3.125|5.0|6.25
    NL5 transfer rate

NL5_CABLE_ENABLE=yes|no
    Enable cabled NL5 links

NL5_NEAR_LB=yes|no
    Configure NL5 channels in near loopback

NL5_SCRAMBLE=yes|no|<NL5 port bitmask>
    Enable scramble mode on NL5 links

NL5_HUB2_WAR=yes|no
    Enable the NL5 PHY/BIST war for Hub2.0
```

To show all variables regardless of state, perform the following:

```
uv44-cmc CMC:rlilc> hwcfg --all
BLADE_DISABLE=no
DEBUG_SW=0x0
HUB_CORE_SPEED=375
```

```
ICH_DISABLE=no
IORISER_DISABLE=no
MAX_CORES=0
NL5_CABLE_ENABLE=yes
NL5_ENABLE=yes
NL5_HUB2_WAR=yes
NL5_NEAR_LB=no
NL5_RATE=6.25
NL5_SCRAMBLE=0x0
PARTITION=0
SMT_ENABLE=no
SOCKET_DISABLE=no
uv44-cmc CMC:rlilc>
```

To show a list of hidden hardware overrides, perform the following:

```
uv44-cmc:~ # hwcfg --hidden --all
BACKPLANE_TYPE=default
BLADE_DISABLE=no
DEBUG_SW=0x100
ICH_DISABLE=no
IORISER_DISABLE=no
MAX_CORES=0
NL5_CABLE_ENABLE=yes
NL5_ENABLE=yes
PARTITION=0
ROUTER_TYPE=ordinary
SMT_DISABLE=no
SOCKET_DISABLE=no
{HIDDEN} BIOS_GO_REG=(null):0x0
{HIDDEN} BIOS_HOLD=no
{HIDDEN} BMC_RESET_DEBUG=0x0
{HIDDEN} BOOTMODE=DC
{HIDDEN} HUB_CORE_SPEED=375
{HIDDEN} HUB_DISABLE=no
{HIDDEN} HUB_QPI_BMC_CONFIG=FULL
{HIDDEN} HUB_QPI_SPEED=5.86
{HIDDEN} HUB_XDP=no
{HIDDEN} IOH_QPI_BMC_CONFIG=NONE
{HIDDEN} IOH_QPI_SPEED=6.4
{HIDDEN} IO_XDP=no
{HIDDEN} LTC_FILE=
{HIDDEN} NL5_CABLE_RATE=6.25
{HIDDEN} NL5_CABLE_TX_EMPHASIS=0x0
{HIDDEN} NL5_HUB2_WAR=yes
```

```

{HIDDEN} NL5_NEAR_LB=no
{HIDDEN} NL5_NO_NI3_WAR=no
{HIDDEN} NL5_RATE=6.25
{HIDDEN} NL5_SCRAMBLE=yes
{HIDDEN} NL5_TUNE_ENABLE=yes
{HIDDEN} NL5_TX_EMPHASIS=0x0
{HIDDEN} QPI_CRC_MODE=0x0
{HIDDEN} SKT_QPI_BMC_CONFIG=NONE
{HIDDEN} SKT_QPI_SPEED=6.4
{HIDDEN} SPREAD_SPEC_CLK=no

```

## Finding the CMC IP Address

CMCs have their rack and u position set at the factory. The CMC will assign itself IP addresses, as follows:

SBK 172.17.<rack>.<slot>

EXT 10.<rack>.<slot>.1

On the system management node (SMN) port, the CMC is configured to request an IP address via dynamic host configuration protocol (DHCP).

To find the IP address of the CMC, connect a network cable to the SMN jack and CMC will request and get a DHCP address. See “Connecting to the UV System Controller Network” on page 17.

The IP address and hostname of your system CMC resides in the `/etc/sysconfig/ifcfg-eth0` file, as follows:

```

CMC:r1ilc> cat /etc/sysconfig/ifcfg-eth0
BOOTPROTO=static
IPADDR=137.38.82.88
NETMASK=255.255.255.0
GATEWAY=137.38.82.254
HOSTNAME=uv15-cmc

```

## System Partitioning

A single SGI Altix UV 100 or 1000 series system can be divided into multiple distinct systems, each with its own console, root filesystem, and IP network address. Each of these software-defined group of processors are distinct systems referred to as a partition. Each partition can be rebooted, loaded with software, powered down, and upgraded independently. The partitions communicate with each other over an SGI NUMALink connection. Collectively, all of these partitions compose a single, shared-memory cluster.

The following example shows how to use CMC software to partition a two rack system containing four IRUs into four distinct systems, use the `console` (same as, `uvcon`) command to open a console and boot each partition and repartiton it back to a single system.

---

**Important:** Each partition must have one base I/O blade and one disk blade for booting. `001i01b00` refers to rack 1, IRU 0, and blade00. `r001i01b01` refers to rack 1, IRU 0, and blade01.

---

Base I/O and the boot disk are displayed by the `config -v` command, similar to the following:

```
r001i01b00 IP93-BASEIO
r001i01b01 IP93-DISK
```

1. Use the `hwcfg` command to create four system partitions, as follows:

```
CMC:r1i1c>hwcfg partition=1 "r1i1b*"
CMC:r1i1c>hwcfg partition=2 "r1i2b*"
CMC:r1i1c>hwcfg partition=3 "r2i1b*"
CMC:r1i1c>hwcfg partition=4 "r2i2b*"
```

2. Use the `config -v` command to show the four partitions, as follows:

```
CMC:r1i1c> config -v

CMCs:                4
    r001i01c UV1000 SMN
    r001i02c UV1000
    r002i01c UV1000
    r002i02c UV1000

BMCs:                64
    r001i01b00 IP93-BASEIO P001
```

```
r001i01b01 IP93-DISK P001
r001i01b02 IP93-INTPCIE P001
r001i01b03 IP93 P001
r001i01b04 IP93 P001
r001i01b05 IP93 P001
r001i01b06 IP93 P001
r001i01b07 IP93 P001
r001i01b08 IP93 P001
r001i01b09 IP93-INTPCIE P001
r001i01b10 IP93-INTPCIE P001
r001i01b11 IP93-INTPCIE P001
r001i01b12 IP93-INTPCIE P001
r001i01b13 IP93 P001
r001i01b14 IP93 P001
r001i01b15 IP93 P001
r001i02b00 IP93-BASEIO P002
r001i02b01 IP93-DISK P002
r001i02b02 IP93-INTPCIE P002
r001i02b03 IP93 P002
r001i02b04 IP93 P002
r001i02b05 IP93 P002
r001i02b06 IP93 P002
r001i02b07 IP93 P002
r001i02b08 IP93 P002
r001i02b09 IP93 P002
r001i02b10 IP93 P002
r001i02b11 IP93 P002
r001i02b12 IP93 P002
r001i02b13 IP93 P002
r001i02b14 IP93 P002
r001i02b15 IP93 P002
r002i01b00 IP93-BASEIO P003
r002i01b01 IP93-DISK P003
r002i01b02 IP93 P003
r002i01b03 IP93 P003
r002i01b04 IP93 P003
r002i01b05 IP93 P003
r002i01b06 IP93 P003
r002i01b07 IP93 P003
r002i01b08 IP93 P003
r002i01b09 IP93 P003
r002i01b10 IP93 P003
r002i01b11 IP93 P003
r002i01b12 IP93 P003
r002i01b13 IP93 P003
```

```
r002i01b14 IP93 P003
r002i01b15 IP93 P003
r002i02b00 IP93-BASEIO P004
r002i02b01 IP93-DISK P004
r002i02b02 IP93 P004
r002i02b03 IP93 P004
r002i02b04 IP93 P004
r002i02b05 IP93 P004
r002i02b06 IP93 P004
r002i02b07 IP93 P004
r002i02b08 IP93 P004
r002i02b09 IP93 P004
r002i02b10 IP93 P004
r002i02b11 IP93 P004
r002i02b12 IP93 P004
r002i02b13 IP93 P004
r002i02b14 IP93 P004
r002i02b15 IP93 P004
```

```
Partitions:      4
  partition001 BMCs:   16
  partition002 BMCs:   16
  partition003 BMCs:   16
  partition004 BMCs:   16
```

3. Use can also use the `hwcfg` command to display the four partitions, as follows:

```
CMC:rlilc> hwcfg
NL5_RATE=5.0
PARTITION=1 ..... 16/64
BMC(s)
PARTITION=2 ..... 16/64
BMC(s)
PARTITION=3 ..... 16/64
BMC(s)
PARTITION=4 ..... 16/64
BMC(s)
```

4. To reset the system and boot the four partitions, use the following commands:

```
CMC:r1ilc> power on "p*"
```

---

**Note:** In the `power on "p*"` command, above, quotes are required to prevent shell expansion.

---

The `power on` command, by itself, will not succeed in this instance because it will attempt to power on across partition boundaries. If all four partitions are to be powered on at once, then the `-o|--override` option must be given or `power on "p*"` must be used.

If the system is off, then a `power on "p*"` is required. If the system is already on, a `power reset "p*"` will cause a reboot.

5. Use the `console` (same as, `uvcon`) command to open consoles to each partition and boot the partitions. Open a console to partition one, as follows:

```
CMC:r1ilc> console p1
console: attempting connection to localhost...
console: connection to SMN/CMC (localhost) established.
console: requesting baseio console access at partition 1
(r001i01b00)...
console: tty mode enabled, use 'CTRL-]' 'q' to exit
console: console access established (OWNER)
console: CMC <--> BASEIO connection active
*****
*****  START OF CACHED CONSOLE OUTPUT  *****
*****
***** [20100513.215944] BMC r001i01b15: Cold Reset via NL
broadcast reset
***** [20100513.215944] BMC r001i01b07: Cold Reset via NL
broadcast reset
***** [20100513.215945] BMC r001i01b13: Cold Reset via NL
broadcast reset
***** [20100513.215945] BMC r001i01b05: Cold Reset via NL
broadcast reset
***** [20100513.215945] BMC r001i01b06: Cold Reset via NL
broadcast reset
***** [20100513.215946] BMC r001i01b10: Cold Reset via NL
broadcast reset
***** [20100513.215946] BMC r001i01b09: Cold Reset via NL
broadcast reset
```

```
***** [20100513.215945] BMC r001i01b11: Cold Reset via NL
broadcast reset
***** [20100513.215945] BMC r001i01b12: Cold Reset via NL
broadcast reset
***** [20100513.215945] BMC r001i01b04: Cold Reset via NL
broadcast reset
***** [20100513.215945] BMC r001i01b08: Cold Reset via NL
broadcast reset
***** [20100513.215946] BMC r001i01b02: Cold Reset via NL
broadcast reset
***** [20100513.215945] BMC r001i01b00: Cold Reset via NL
broadcast reset
***** [20100513.215945] BMC r001i01b14: Cold Reset via NL
broadcast reset
***** [20100513.215947] BMC r001i01b09: Cold Reset via ICH
***** [20100513.215946] BMC r001i01b12: Cold Reset via ICH
***** [20100513.215947] BMC r001i01b10: Cold Reset via ICH
***** [20100513.215947] BMC r001i01b11: Cold Reset via ICH
***** [20100513.215947] BMC r001i01b02: Cold Reset via ICH
***** [20100513.215947] BMC r001i01b00: Cold Reset via ICH
***** [20100513.215953] BMC r001i01b03: Cold Reset via NL
broadcast reset
***** [20100513.220011] BMC r001i01b01: Cold Reset via NL
broadcast reset
***** [20100513.220012] BMC r001i01b08: Cold Reset via NL
broadcast reset
***** [20100513.220012] BMC r001i01b07: Cold Reset via NL
broadcast reset
***** [20100513.220011] BMC r001i01b15: Cold Reset via NL
broadcast reset
***** [20100513.220012] BMC r001i01b06: Cold Reset via NL
broadcast reset
***** [20100513.220012] BMC r001i01b05: Cold Reset via NL
broadcast reset
***** [20100513.220012] BMC r001i01b14: Cold Reset via NL
broadcast reset
***** [20100513.220012] BMC r001i01b13: Cold Reset via NL
broadcast reset
***** [20100513.220011] BMC r001i01b04: Cold Reset via NL
broadcast reset
***** [20100513.220012] BMC r001i01b03: Cold Reset via NL
broadcast reset
***** [20100513.220013] BMC r001i01b09: Cold Reset via NL
broadcast reset
```

```

***** [20100513.220013] BMC r001i01b10: Cold Reset via NL
broadcast reset
***** [20100513.220013] BMC r001i01b11: Cold Reset via NL
broadcast reset
***** [20100513.220012] BMC r001i01b12: Cold Reset via NL
broadcast reset
***** [20100513.220012] BMC r001i01b02: Cold Reset via NL
broadcast reset
***** [20100513.220012] BMC r001i01b00: Cold Reset via NL
broadcast reset
***** [20100513.220014] BMC r001i01b09: Cold Reset via ICH
***** [20100513.220014] BMC r001i01b10: Cold Reset via ICH
***** [20100513.220014] BMC r001i01b11: Cold Reset via ICH
***** [20100513.220013] BMC r001i01b12: Cold Reset via ICH
***** [20100513.220013] BMC r001i01b02: Cold Reset via ICH
***** [20100513.220016] BMC r001i01b00: Cold Reset via ICH
***** [20100513.220035] BMC r001i01b14: Cold Reset via NL
broadcast reset
***** [20100513.220035] BMC r001i01b06: Cold Reset via NL
broadcast reset
***** [20100513.220034] BMC r001i01b15: Cold Reset via NL
broadcast reset
***** [20100513.220035] BMC r001i01b05: Cold Reset via NL
broadcast reset
***** [20100513.220034] BMC r001i01b01: Cold Reset via NL
broadcast reset
***** [20100513.220035] BMC r001i01b07: Cold Reset via NL
broadcast reset
.....
Hit [Space] for Boot Menu.
ELILO boot:
...          .....

```

---

**Note:** Use the `console` command to open consoles on the other three partitions and boot them. The system will then have four single system images.

---

6. Use the `hwcfg -c partition` command to clear the four partitions, as follows:

```
CMC:r1i1c> hwcfg -c partition
PARTITION=0 <PENDING RESET>
```

7. To reset the system and boot it as a single system image (one partition), use the following command:

```
CMC:r1i1c> power reset "p*"
```

For detailed instructions on how to use the UV controller commands to partition a system, see “System Partitioning” in the *SGI Altix UV Linux Configuration and Operations Guide*.

## Upgrading System BIOS

To upgrade the compute blade BIOS, perform the following steps:

1. From the CMC prompt, to show the current PROM level perform the following command:

```
CMC:r1ilc> bios
Flashed on Sat May  1 14:14:45 UTC 2010 was bios.latest.fd
(20100429_1603)
```

2. Get the newest PROM image from SupportFolio Online at <http://support.sgi.com/>

---

**Note:** Upgrading to Altix UV BIOS 2.00 release (or later) from a pre-2.00 Altix UV BIOS release requires the setup variables to be cleared using either the `flashbios -e` or `power -c reset` CMC commands. This will also clear any site specific settings, such as M/N values and EFI boot menu changes, and these site specific settings will need to be reconfigured at the 2.00 boot menu.

---

3. Copy the latest BIOS to a directory on the CMC in `/work/bmc/common/` An example directory is, as follows:

```
CMC:r1ilc> ls
bios.latest.fd flashbios
```

4. Use the `flashbios` command to flash the compute blade BIOS, as follows:

```
CMC:r1ilc> flashbios
Using default bios: bios.latest.fd
Checking processor status on all nodes...
Done. System is read for BIOS flash update
Flashing bios bios.latest.fd (20100429_1603) This will take several
minutes.
...
```

There are three firmware flashing commands available from the system management node (SMN) for flashing an entire SGI Altix UV system, as follows:

- `flashcmc`

- flashbmc
- flashiobmc

For more information on how to use these commands, see the “Updating Firmware” section in chapter one of the *SGI Altix UV System Management Node Administrator’s Guide*.

## Hyper-Threading on Altix UV 100 or Altix UV 1000 Systems

*Threading* in a software application splits instructions into multiple streams so that multiple processors can act on them.

Hyper-Threading (HT) Technology, developed by Intel Corporation, provides thread-level parallelism on each processor, resulting in more efficient use of processor resources, higher processing throughput, and improved performance. One physical CPU can appear as two logical CPUs by having additional registers to overlap two instruction streams or a single processor can have dual-cores executing instructions in parallel.

For more information about using HT, see “Using Cpusets with Hyper-Threads” in the *Linux Resource Administration Guide*.



## Altix UV CMC Software Commands

You can use SGI Altix UV controller commands to monitor and manage SGI Altix UV systems. You can use them from the SGI Management Node command line interface (CLI) or the chassis manager controller (CMC) CLI.

Typically, commands available from the SMN or CMC command line are in the form of:

command [options] [targets]

For available commands see below or type "help" at the SMN/CMC prompt. Commands are located in `/sysco/bin` on both the SMN and CMC. For options available with each command, type "command -help".

### CMC Command Targets

Multiple targets may be specified, if no targets are specified, all BMCs or CMCs (as appropriate for a command) are assumed. Some targets may need to be quoted to avoid wildcard expansion by the SMN/CMC shell.

**Table 3-1** CMC Command Targets

Command Target	BMCs (rack,upos,slot,BMC type)	CMCs (rack,upos,CMC type)
*,all	Any,Any,Any,Any	Any,Any,Any
r*	rack,Any,Any,Any	rack,Any,Any
r*i*	rack,upos,Any,IRURTR+IRUCOMP	rack,upos,IRU
r*q*	rack,upos,Any, QCRRTR	rack,u, QCR
r*i*b*	rack,upos,slot,IRUCOMP	N/A
r*i*r*	rack,upos,slot,IRURTR	N/A

**Table 3-1** CMC Command Targets

Command Target	BMCs (rack,upos,slot,BMC type)	CMCs (rack,upos,CMC type)
r*q*r*	rack,upos,slot,QCRRTR	N/A
*c	N/A	Any,Any,Any
r*i*c	N/A	rack,upos,IRU
r*q*c	N/A	rack,upos, QCR
allb,allc	Any,Any,Any,IRUCOMP	N/A
allbi,allci	ANY,ANY,ANY,IRUIO	N/A
allr	Any,Any,Any,IRURTR+QCRRTR	N/A
allri	Any,Any,Any,IRUCOMP+IRURTR	N/A
allrq	Any,Any,Any,QCRRTR	N/A
p*	all IRUCOMP with matching partition	N/A

An asterisk references all values in that position such that r\* references all racks, i\* references all IRUs, etc. A decimal number can replace any asterisk to narrow the target selection. Ranges are not support but a space separated list of targets is supported.

### BMC Types

IRUCOMP - IRU computer blades (blade slots 0 through 15 on Altix UV 1000, or 0 through 1 on Altix UV 100).

IRUIO - IRU computer blades with an IO riser (blade slots 0 through 15 on Altix UV 1000, or 0 through 2 on Altix UV 100)

IRURTR - IRU router blades (router slots 0 through 3)

QCRRTR - Quad Compact Router (QCR) router blades (slots 0 through 3)

### CMC Types

IRU - Altix UV 1000 or Altix UV 100

---

**Note:** Most of the commands (bios, bmc, cmc, config, hwcfg, log, power, uvcon (console)) are available at both the CMC prompt (as user **root**), as well as, the system management node (SMN) prompt (as user **sysco**).

---

The available commands on the CMC are, as follows:

```
CMC:r1i1c> help
```

available commands are:

auth	authenticate SSN/APPWT change
bios	perform bios actions
bmc	access BMC shell
cmc	access CMC shell
config	show system configuration
console	access system consoles
help	list available commands
hel	access hardware error logs
hwcfg	access hardware configuration variable
leds	display system LED values
log	display system controller logs
power	access power control/status

type '`<cmd> --help`' for help on individual command.

## auth

The `auth` command is currently only used to set the APPWT limit for export compliance. You need to obtain the key and reset the limit if needed.

You can get a usage statement, as follows:

```
CMC:r014i01c> auth --help
```

```
usage: auth [-ard] [-s <SSN>] <key> [<key>]... [--help] [TARGET]...
-s, --ssn=<SSN>          system serial number change
-a, --appwt              APPWT change
-r, --reset              set learn mode
-d, --disable            disable authentication
<key>                   authentication keys
--help                  display this help and exit
```

## autopower

The `autopower` command allows your system to power up automatically when power is applied after a power outage. Issuing the `autopower` command without any argument acts a query, with the results displaying the current state (enabled/disabled) and if enabled, the current delay time, and the results of the last attempted auto power on.

You can use the `autopower` command to determine if autopower is enabled, as follows:

```
CMC:r001i01c> autopower
==== r001i01c (PRI) ====
auto-power on is enabled (120 second delay), aborted (power already
on).
```

Use the `autopower -d` command to disable autopower, as follows:

```
uv44-cmc CMC:r001i01c> autopower -d
==== r001i01c (PRI) ====
auto-power on disabled
```

You can get a usage statement, as follows:

```
CMC:r001i01c> autopower --help

usage: autopower [-edc] [-t <seconds>] [--help]
-e, --reset          enable auto-power on
-t, --time=<seconds> time (seconds) to delay auto-power on
-d, --disable        disable auto-power on
-c, --cancel         cancelauto-power on
--help              display this help and exit
```

## bios

Use the `bios` command to get BIOS information for your system after it is powered on, as follows:

```
CMC:r1i1c> bios
==== 4/4 BMC(s) ====
Last booted: SGI BIOS Version 2 Revision 0 built in 20101012_1502 by
ajm on Oct 12 2010 at 15:08:29
```

You can get a usage statement, as follows:

```
CMC:r1i1c> bios --help
```

```
usage: bios [-rsuv] [--help] [TARGET]...
-r, --revision          display last BIOS revision banner
-s, --state             display current BIOS state
-u, --uptime           display time since last BIOS reset
-v, --verbose          verbose output
--help                 display this help and exit
```

## bmc

Allows you to send a command to one or more board management controllers (BMCs). It is a shell command similar in behavior to the `ssh` command.

You can get a usage statement, as follows:

```
CMC:r1i1c> bmc --help
```

```
usage: bmc [-t] exec <command> [--timeout=<seconds>] [TARGET]...
exec          executes command on BMC(s)
<command>    command to execute
--timeout=<seconds>  timeout value
-t, --terse   terse mode
```

```
usage: bmc list [TARGET]...
list          list active shells on BMC(s)
```

```
usage: bmc kill [TARGET]...
kill          kill all active shells on BMC(s)
```

```
usage: bmc [-v] runtime [TARGET]...
runtime       show daemon run time
-v, --verbose verbose mode
```

```
usage: bmc reboot [TARGET]...
reboot        initiates controller reboot
```

```
usage: bmc --help
--help        display this help and exit
```

## cmc

Allows you to send a command to one or more chassis manager controllers (CMCs). It is a shell command similar in behavior to the `ssh` command.

```
CMC:r1i1c> cmc --help
```

```
usage: cmc [-t] exec <command> [--timeout=<seconds>] [TARGET]...
exec                                     executes command on CMC(s)
<command>                               command to execute
--timeout=<seconds>                     timeout value
-t, --terse                             terse mode
```

```
usage: cmc list [TARGET]...
list                                     list active shells on CMC(s)
```

```
usage: cmc kill [TARGET]...
kill                                     kill all active shells on CMC(s)
```

```
usage: cmc [-v] runtime [TARGET]...
runtime                                 show daemon run time
-v, --verbose                           verbose mode
```

```
usage: cmc reboot [TARGET]...
reboot                                  initiates controller reboot
```

```
usage: cmc --help
--help                                  display this help and exit
```

## config

The CMC `config` command shows your system configuration, as follows:

```
CMC:r1i1c> config -v
```

```
CMCs:          1
               r001i01c UV1000 SMN

BMCs:          4
               r001i01b00 IP93-BASEIO
               r001i01b01 IP93-DISK
               r001i01b02 IP93
               r001i01b03 IP93
```

```

Partitions:      1
                partition000 BMCs:    4

Accessories:    0

```

You can get a usage statement, as follows:

```

CMC:r1i1c> config --help
usage: config [-tvd] [-a <ip>@<location>] [--pcheck] [--plist]
[<TARGET>]... [--help]
-t, --terse                terse output
-v, --verbose              verbose output
-d, --debug                debug output
-a, --acc=<ip>@<location>
                           configure accessory location
                           <location> is <rack>.<uposition> for IO
                           <location> is <rack>.<0|1> for PDUs (0=left, 1=right)

--pcheck                   check for targets crossing partitions
--plist                    list partitions targeted
--help                     display this help and exit

```

The `-t/--terse` option produces a terse configuration report.

On an SGI Altix UV 100 system with two IRUs, it produces a terse configuration report similar to the following:

```

SSN: UVL-00000001, CMCs: 2, BMCs 4/4, Partitions: 1, Accessories: 0
r001i01c b-00-01
r001i02c b-00-01

```

On an SGI Altix UV 100 system with eight IRUs, it produces a terse configuration report similar to the following:

```

SSN: UV-00000048, CMCs: 16, BMCs 192/192, Partitions: 5, Accessories: 0
r001i01c b-00-01-02-03-04-05-06-07-08-09-10-11-12-13-14-15
r-00-01-02-03
r001i23c b-00-01-02-03-04-05-06-07-08-09-10-11-12-13-14-15
r-00-01-02-03
r001q42c r-00-01-02-03
r001q43c r-00-01-02-03
r002i01c b-00-01-02-03-04-05-06-07-08-09-10-11-12-13-14-15
r-00-01-02-03

```

```
r002i23c b-00-01-02-03-04-05-06-07-08-09-10-11-12-13-14-15
r-00-01-02-03
r002q42c r-00-01-02-03
r002q43c r-00-01-02-03
r003i01c b-00-01-02-03-04-05-06-07-08-09-10-11-12-13-14-15
r-00-01-02-03
r003i23c b-00-01-02-03-04-05-06-07-08-09-10-11-12-13-14-15
r-00-01-02-03
r003q42c r-00-01-02-03
r003q43c r-00-01-02-03
r004i01c b-00-01-02-03-04-05-06-07-08-09-10-11-12-13-14-15
r-00-01-02-03
r004i23c b-00-01-02-03-04-05-06-07-08-09-10-11-12-13-14-15
r-00-01-02-03
r004q42c r-00-01-02-03
```

**For a single Altix UV 1000 IRU without blades in slots 0 through 7 (or router blades), this output shows the 'holes' for blades not present, similar to the following:**

```
SSN: UV-00000021, CMCs: 1, BMCs 8/8, Partitions: 1, Accessories: 2
r001i01c b- - - - - - - - -08-09-10-11-12-13-14-15 r- - - -
```

**Also, on VT100 compliant terminals, it will use the following character attributes:**

Reverse red - blade is connected, but not configured

Reverse yellow - blade is disabled (see hwcfg command)

## console (uvcon)

The console (uvcon) command allows you to open a console window on the CMC or BMC or even a compute blade. Use CTRL -] q to exit. A usage statement is, as follows:

```
CMC:r014i01c> console --help
```

```
usage: console [-bnd23] [--steal] [--spy] [--kill] [--notty] [--nocache] [--clear] [TARGET]
-b, --baseio          specifies baseio bmc console
-n, -0, --normal      specifies node BMC console (normal channel)
-d, -1, --debug       specifies node BMC console (debug channel)
-2, --chan2           specifies node BMC console (channel 2)
-3, --chan3           specifies node BMC console (channel 3)
--steal               steal the console
--spy                 spy the console
--kill                kill all other uvcon sessions
--notty               disables tty interventions
--nocache             don't return cached output
--clear               clear cached output
TARGET                console target
```

escape codes:

```
ctrl-] s    steal console
ctrl-] r    release console
ctrl-] c    connection status
ctrl-] ?|h  this help
```

```
usage: console -d[band23cCft] [-l <count>] [TARGET]...
-d, --dump          dump cached console output
-b, --baseio        specifies baseio bmc console
-a, --all           all node BMC consoles
-n, -0, --normal    specifies node BMC console (normal channel)
-d, -1, --debug     specifies node BMC console (debug channel)
-2, --chan2         specifies node BMC console (channel 2)
-3, --chan3         specifies node BMC console (channel 3)
-l, --lines=<count> limit output to last <count> lines
-c, --clear         clear cached output after dumping
-C, --CLEAR         clear cached output without dumping
-f, --full          full dump (default is since last reset)
-t, --tag           prefix each line with source
TARGET              console target
```

```
usage: console [-bnd23] -i <input> [TARGET]...
-i, --input=<input>          backdoor console input
-b, --baseio                 specifies baseio bmc console
-n, -0, --normal             specifies node BMC console (normal channel)
-d, -1, --debug              specifies node BMC console (debug channel)
-2, --chan2                  specifies node BMC console (channel 2)
-3, --chan3                  specifies node BMC console (channel 3)
TARGET                       console target
```

```
usage: console --help
--help                       display this help and exit
```

Note that the following character sequences are available interactively one a console session is started:

```
console: escape codes:
console:  ctrl-] s    steal console
console:  ctrl-] r    release console
console:  ctrl-] c    connection status
console:  ctrl-] ?|h  this help
```

## flashbios

Use the `flashbios` command to flash the latest BIOS located in the `/work/bmc/common` directory on your CMC. For an example of how this command is used, see “Upgrading System BIOS” on page 34.

You can get a usage statement, as follows:

```
CMC:r1ilc> flashbios --help
```

```
Illegal option --
```

```
NAME
```

```
flashbios -- Flash UV BIOS from CMC
```

```
SYNOPSIS
```

```
flashbios [-e] [-n] [-p] [-o] <flashfile> [<target> [...<target>]]
flashbios -r <flashfile>
```

```
DESCRIPTION
```

The file specified by `flashfile` is flashed into the flash devices on all nodes of the UV system. By default the system will be automatically reset after the flash completes.

The BIOS image file <flashfile> must be located in the directory /work/bmc/common on the CMC.

By default, BIOS is updated on all compute blades in the system. The flash can be restricted to a subset of the full system by specifying one or more targets. Some examples of blade targets:

```

r1i23b4      blade 4 in the IRU @ rack 1, u-position 23
r2i1b*      all blades in IRU @ rack 2i1
r4i*        all blades in all IRUs in rack 4
p2          all blades in partition 2

```

(These are the same targets accepted by other CMC commands). Any number or combination of targets can be used.

The following options are available:

- e Erase BIOS variables and BIOS scratch space while resetting the system. This option will be ignored if '-n' is also specified.
- n Do not automatically reset the system.
- o Allow flashing across partitions in a multiple partition system
- p Power cycle after flashing.
- r Do not flash, instead report the Revision (BIOS banner) string from the BIOS image file.

## hwcfg

The `hwcfg` command can be used to access hardware override variables.

```

CMC:r1i1c> hwcfg -a -v
BLADE_DISABLE=no
DEBUG_SW=0x0
HUB_CORE_SPEED=400
ICH_DISABLE=no
IORISER_DISABLE=no
MAX_CORES=0
NL5_CABLE_ENABLE=yes

```

```
NL5_ENABLE=yes
NL5_HUB2_WAR=no
NL5_NEAR_LB=no
NL5_RATE=6.25
NL5_SCRAMBLE=0x0
PARTITION=0
SMT_ENABLE=no
SOCKET_DISABLE=no
```

Here is an example showing a system with four partition.

```
uv32-cmc CMC:rlilc> hwcfg -a -v
NL5_RATE=5.0
PARTITION=1 ..... 16/64
BMC(s)
PARTITION=2 ..... 16/64
BMC(s)
PARTITION=3 ..... 16/64
BMC(s)
PARTITION=4 ..... 16/64
BMC(s)
```

You can use `hwcfg -c` to clear the four partitions, as follows:

```
uv32-cmc CMC:rlilc> hwcfg -c partition
PARTITION=0 <PENDING RESET>
```

You can get a usage statement, as follows:

```
CMC:rlilc> hwcfg --help
```

```
usage: hwcfg [-lhadcv] [var[=val]]... [--help] [TARGET]...
```

var[=val]	variable [and value to set]
-l, --list	list variables available
-h, --hidden	include hidden variables
-a, --all	show variable(s) regardless of override state
-d, --default	show variable(s) default value if overridden
-c, --clear	clear overridden variable(s) to default value
-v, --verbose	show per BMC output, repeat for maximum verbosity
--help	display this help and exit, use with variable(s) to get specific
help	help

## log

Provides a log of various operations performed on the CMC.

You can get a usage statement, as follows:

```
uv44-cmc CMC:r001i01c> log --help

usage: log [-pacewnidmr] [--help] [TARGET]...
-p, --panic           select log type
-a, --alert           select log type
-c, --critical        select log type
-e, --error           select log type
-w, --warning         select log type
-n, --notice          select log type
-i, --info            select log type
-d, --debug           select log type
-m, --merge           prevent merging of selected logs
-r, --reset           reset log
--help                display this help and exit
```

## power

The CMC `power` command allows you to power on, power off, reset, cycle, get status and invoke the kernel debugger (KDB). When using the `power` command, you no longer have to power up the individual rack unit (IRU). When you issue the `power` command, it checks to see if the IRU is powered on, if not, it will power it up first and then the compute blades.

You can get a usage statement, as follows:

```
CMC:rlilc> power --help

usage: power [-vchosw] on|up [bmc] [--noio] [TARGET]...
on|up           turn power on
bmc             turn aux power on
--noio          do not power on accessories (IO)
-v, --verbose   verbose output
-c, --clear     clear EFI variables (system/partition targets only)
-h, --hold     hold in reset
-o, --override  override partition check
-s, --single   single node boot
-w, --watch    watch boot progress
```

```
usage: power [-vo] off|down [bmc] [--noio] [--nochassis] [TARGET]...
off|down          turn power off
bmc              turn aux power off
--noio          do not power off accessories (IO)
--nochassis     do not power off chassis power
-v, --verbose   verbose output
-o, --override  override partition check
```

```
usage: power [-vchosw] reset [bmc|iobmc] [TARGET]...
reset            system reset
bmc|iobmc       BMC reset
-v, --verbose   verbose output
-c, --clear     clear EFI variables (system/partition targets only)
-h, --hold     hold in reset
-o, --override  override partition check
-s, --single   single node boot
-w, --watch    watch boot progress
```

```
usage: power [-vchosw] cycle [bmc] [--noio] [--nochassis] [TARGET]...
cycle           cycle power off on
bmc            cycle aux power
--noio        do not power cycle accessories (IO)
--nochassis   do not power cycle chassis power
-v, --verbose verbose output
-c, --clear   clear EFI variables (system/partition targets only)
-h, --hold   hold in reset
-o, --override override partition check
-s, --single single node boot
-w, --watch  watch boot progress
```

```
usage: power [-vl0ud] [status] [TARGET]...
status      show power status
-v, --verbose verbose output
-l, --on    show only blades with on status
-0, --off   show only blades with off status
-u, --unknown show only blades with unknown status
-d, --disabled show only blades with disabled status
```

```
usage: power [-ov] nmi|debug [TARGET]...
nmi|debug      issue NMI
-o, --override override partition check
-v, --verbose  verbose output
usage: power [-v] margin [high|low|norm|<value>] [TARGET]...
margin         power margin control
```

```

high|low|norm|<value>    margin state
-v, --verbose            verbose output

usage: power cancel [TARGET]...
cancel                  cancel outstanding power action

usage: power --help
--help                  display this help and exit

```

## sensor

Only valid for CMC data in this form. Use the `sensor` command to get system temperatures, fan speed, and voltage information and so on, as follows:

```
CMC:r1i1c> sensor
```

Use the command from the `cmc` or `bmc`, as follows:

```
cmc sensor
bmc sensor
```

## uvcon

The `uvcon` command is now just an alias to the `console` command. See “console (`uvcon`)” on page 45.

## version

Use the `version` command to determine the CMC firmware version, as follows:

```
uv44-cmc CMC:r001i01c> version
SGI Chassis Manager Controller, Firmware Rev. 1.3.16 [Bootloader
0.6.0]
```

You can also use the `version` command to determine the BMC firmware version, as follows:

```
uv44-cmc CMC:r001i01c> bmc version
==== r001i01b00 ====
```

```
SGI UV BMC, Firmware Rev. 1.3.20 [Bootloader 0.5.0, FPGA Rev. 0x41]
==== r001i01b01 ====
SGI UV BMC, Firmware Rev. 1.3.20 [Bootloader 0.5.0, FPGA Rev. 0x41]
```

**You can also ssh to the BMC and use the version command, as follows:**

```
uv44-cmc CMC:r001i01c> ssh bmc0
```

```
SGI UV BMC, Firmware Rev. 1.3.20
```

```
BMC:r001i01b00> version
```

```
SGI UV BMC, Firmware Rev. 1.3.20 [Bootloader 0.5.0, FPGA Rev. 0x41]
```

**You can also ssh to BaseIO blade BMC and use the version command, as follows:**

```
uv44-cmc CMC:r001i01c> ssh ibmc0
```

```
SGI UV BMC, Firmware Rev. 1.3.9
```

```
r001i01b00i> version
```

```
SGI Chassis Manager Controller, Firmware Rev. 1.3.9 [Bootloader
0.3.0]
```

---

# Index

## A

### accessories

- ePDU, 13
- PCIE expansion chassis, 13

## B

BMC types, 38

booting, 18

## C

chassis manager controller, 8

CMC command targets, 37

CMC controller software overview, 14

CMC types, 38

### commands

- auth, 39
- autopower, 40
- bios, 34, 40
- bmc, 41
- cmc, 42
- config, 23, 28, 42
- flashbios, 34, 46
- help, 37
- hwcfg, 28, 47
- log, 49
- power, 49
- power reset, 31

sensor, 51

uvcon, 31, 51

version, 51

connecting to the controller network, 17

## D

determining rack numbers, 11

determining the BaseIO node BMC firmware version, 52

determining the CMC firmware version, 51

determining the compute node BMC firmware version, 51

## E

enabling autopower, 21

## F

finding the CMC IP address, 27

## H

hyper-threading, 35

## I

- individual rack unit (IRU), 2
- introduction, 1
- IP address
  - CMC, 27

## K

- kernel debugger (KDB), 21

## N

- nonmaskable interrupt (KDB) kernel debugger, 21

## P

- PCIE expansion chassis, 13
- power off, 20
- power on, 18

## Q

- Quad Compact Router (QCR), 38

## S

- SGI Altix UV 100 system overview, 6
- SGI Altix UV 1000 system overview, 1
- SGI Management Center (SMC), 8
- SGI Management Node (SMN), 8
- system control network, 9, 17
- system management, 7
- system partitioning, 28

## U

- upgrading system PROM, 34
- upgrading to Altix UV BIOS 2.0 release (or later), 34
- using the hardware configuration command, 24

## V

- viewing your system configuration, 23