

**SGI® L1 and L2 Controller
Software User's Guide**

007-3938-006

CONTRIBUTORS

Written by Linda Rae Sande

Revised by Terry Schultz

Illustrated by Chrystie Danzer

Production by Terry Schultz

Engineering contributions by Don Adams, Michael T. Brown, Dick Brownell, Jason Chang, Steve Hein, Jill Heitpas, Nancy Heller, Matt Hoy, Hao Pham, Craig Schultz, and Lisa Steinmetz.

COPYRIGHT

© 2002, 2003, 2004, 2005, 2006, Silicon Graphics, Inc. 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 Silicon Graphics, Inc.

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

Silicon Graphics, SGI, the SGI logo, Altix, are registered trademarks and NUMAflex, NUMAlink, and SGIconsole are trademarks of Silicon Graphics, Inc., in the U.S. and/or other countries worldwide. All other trademarks mentioned herein are the property of their respective owners.

New Features in This Guide

Major Documentation Changes

The following sections were revised for this release:

Note: This manual does not apply to SGI Altix XE systems. It supports the Altix 3000 series, Altix 450, and Altix 4700 systems only. For information about SGI legacy systems, see an earlier version of this manual. From the current 007-3938-006 version of this manual on the SGI Technical Publications Library, select the **additional info** link. Click on **007-4633-005** under **Other Versions** :

- Added information about blade system architecture in “L1 Controller” on page 1.
- Updated information about upgrading firmware for SGI Altix systems in “Upgrading System Controller Firmware” on page 35.
- Updated information about upgrading firmware for SGI Altix systems in “Upgrading Prom Firmware” on page 36.
- Added a new section called “System Partitioning” on page 44.
- Added a new section called “Hyper-Threading on Altix 450 or Altix 4700 Systems” on page 45.
- Updated information and examples about L1 controller commands in “L1 Controller Commands” on page 52.
- Added a new section called “L1 Controller Commands for Expert Users” on page 95.
- Updated information and examples about L2 controller commands in “L2 Controller Commands” on page 111.

Record of Revision

Version	Description
001	May 2002 Initial revision.
002	February 2003 Updated to support the SGI Origin 3900 and SGI Altix 3000 series of servers.
003	May 2004 Updated to support the SGI Altix 350 systems.
004	July 2005 Updated to support SGI Altix 3700 Bx2 systems and Silicon Graphics Prism Visualization Systems.
005	April 2006 Updated to support the SGI Altix 4700 sytem.
006	October 2006 Updated to support the SGI ProPack 5 for Linux release.

Operating the L1 Controller on Altix 3000 Systems23
L1 Mode24
Viewing System Configuration24
Command Targeting27
Viewing Information, Warnings, and Error Messages27
Powering On, Powering Off, and Resetting the Brick28
Console Mode from L128
Console Selection on Systems29
Operating the L1 Controller on Altix 450 and Altix 4700 Systems30
L1 Mode>31
Direct Connection to an L131
L1 Mode on Altix 450 and Altix 4700 Systems31
Viewing System Configuration (from an IRU's Perspective)32
Command Targeting32
Console Mode from L1 on Altix 450 or Altix 4700 Systems33
L1 Console Selection on SGI Altix 450 and Altix 4700 Systems34
Viewing Information, Warnings, and Error Messages on Altix 4700 Systems35
Upgrading System Controller Firmware35
Upgrading Prom Firmware36
Identifying Bricks39
Identifying Blades40
System Partitioning44
Hyper-Threading on Altix 450 or Altix 4700 Systems45
Status and Error Messages46
L1 Controller Tasks and Messages46
L2 Controller Tasks48
3. L1 and L2 Controller Commands51
Commands for Different Devices.51
L2 Commands52
L1 Commands52
L1 Controller Commands52
* (asterisk character)53
autopower54

blade	54
brick	55
b2b	57
config	59
cpu	61
date	62
debug	63
display	65
env	66
fan	72
flash	72
help	73
history	75
l1	76
leds	76
link	77
log	77
network	79
nmi	80
partdb	80
pbay	80
pci	82
port	84
power	87
reboot_l1	90
reset	91
router	91
select	92
serial	93
softreset	94
verbose	95
version	95
L1 Controller Commands for Expert Users	95
eeprom	95

ioport96
istat98
junkbus	100
l1dbg and l2dbg	100
margin	103
scan	105
rmmr	108
tdr	108
test	109
uart	110
usb	111
L2 Controller Commands	111
autopower	111
config	112
date	116
destination	117
dhcpserver	119
env	119
help	120
ip.	120
l1.	120
l2.	122
l2find	123
log	123
multisys.	124
nvram reset.	124
password	125
pbay.	125
ping	125
power	126
quit	127
rackid	127
reboot_l2	128
reset	128

select128
serial131
shell.131
smp132
sysname133
version133
Index135

Figures

Figure 2-1	Individual Rack Unit (IRU) Blade Slot IDs	41
Figure 2-2	Rack and Module IDs on SGI Altix 4700 System	43

Tables

Table 2-1	Valid Blade Types	40
Table 2-2	L1 Controller Messages	47
Table 3-1	Valid Brick Types	60
Table 3-2	Virtual Debug Switch Settings	64
Table 3-3	Valid Brick Types115

Examples

Example 3-1	* version Command Output	54
Example 3-2	autopower Command Output	54
Example 3-3	blade Command Output	55
Example 3-4	brick Command Output	56
Example 3-5	brick rack <rack number> Command Output	56
Example 3-6	brick slot <slot number> Command Output	57
Example 3-7	brick rackslot <rack number> <slot number> Command Output.	57
Example 3-8	brick partition none Command Output	57
Example 3-9	brick partition <partition> Command Output	57
Example 3-10	b2b Command Output	59
Example 3-11	config Command Output	60
Example 3-12	config Command Output	60
Example 3-13	cpu Command Output on a Blade-Based Altix System	61
Example 3-14	cpu Command Output on a Non-Blade Altix System.	62
Example 3-15	date Command Output	63
Example 3-16	debug Command Output	63
Example 3-17	debug <switches> Command Output	63
Example 3-18	display Command Output	65
Example 3-19	env Command Output	70
Example 3-20	env altitude high Command Output	72
Example 3-21	env reset Command Output	72
Example 3-22	fan Command Output	72
Example 3-23	flash status Command Output	73
Example 3-24	help Command Output	74
Example 3-25	history Command Output	75
Example 3-26	ll Command Output	76
Example 3-27	leds Command Output on a Blade-Based Altix System	76

Example 3-28	leds Command Output on a Non-Blade Altix System.77
Example 3-29	log Command Output78
Example 3-30	log reset Command Output78
Example 3-31	log insert Start the Test Command Output79
Example 3-32	network Command Output79
Example 3-33	network usb Command Output79
Example 3-34	nmi Command Output80
Example 3-35	pbay Command Output82
Example 3-36	pbay version Command Output82
Example 3-37	pci Command Output83
Example 3-38	pci d Command Output84
Example 3-39	pci 1 u Command Output84
Example 3-40	pci 1 2 u Command Output84
Example 3-41	pci reset Command Output84
Example 3-42	pci 1 reset Command Output84
Example 3-43	pci 1 2 reset Command Output84
Example 3-44	port Command Output for an R-brick85
Example 3-45	port Command Output for a C-brick85
Example 3-46	port v Command Output for Blade-Based Altix System85
Example 3-47	port -v Command Output for non-Blade Altix Systems86
Example 3-48	power Command Output on Blade-Based Altix System88
Example 3-49	power Command Output on Non-Blade Altix System89
Example 3-50	power check Command Output90
Example 3-51	power up Command Output90
Example 3-52	power vrm Command Output90
Example 3-53	reset Command Output91
Example 3-54	router Command Output92
Example 3-55	select Command Output93
Example 3-56	softreset Command Output94
Example 3-57	version Command Output95
Example 3-58	eeeprom Command Output95
Example 3-59	ioport Command Output for a Compute Brick96
Example 3-60	ioport Command Output for an I/O Brick.97

Example 3-61	<code>istat memory</code> Command Output	98
Example 3-62	<code>istat queues</code> Command Output	98
Example 3-63	<code>istat tasks</code> Command Output	99
Example 3-64	<code>istat pmalloc</code> Command Output	99
Example 3-65	<code>junkbus</code> Command Output	100
Example 3-66	<code>lldbg</code> Command Output	103
Example 3-67	<code>lldbg junkbus on</code> Command Output	103
Example 3-68	<code>margin</code> Command Output	105
Example 3-69	<code>scan chip</code> Command Output	108
Example 3-70	<code>test intr</code> Command Output	110
Example 3-71	<code>uart</code> Command Output	111
Example 3-72	<code>usb</code> Command Output	111
Example 3-73	<code>autopower</code> Command Output	112
Example 3-74	<code>config</code> Command output	114
Example 3-75	<code>config summary</code> Command output	115
Example 3-76	<code>config list</code> Command output	116
Example 3-77	<code>config snap</code> Command output	116
Example 3-78	<code>config diff</code> Command output	116
Example 3-79	<code>date</code> Command Output	117
Example 3-80	<code>destination reset</code> Command Output.	118
Example 3-81	<code>destination</code> Command Output	118
Example 3-82	<code>rack <rng> slot <rng> destination</code> Command Output. .118	118
Example 3-83	<code>dhcpserver</code> Command Output	119
Example 3-84	<code>env summary</code> Command Output	119
Example 3-85	<code>ip</code> Command Output	120
Example 3-86	<code>rack <rng> slot <rng> l1</code> Command Output	121
Example 3-87	<code>rack <rng> slot <rng> l1 config</code> Command Output . .121	121
Example 3-88	<code>l2</code> Command Output	122
Example 3-89	<code>l2find</code> Command Output	123
Example 3-90	<code>log insert</code> Start the Test Command Output.	124
Example 3-91	<code>multisys</code> Command Output	124
Example 3-92	<code>power</code> Command Output	127
Example 3-93	<code>power summary</code> Command Output	127

Example 3-94	rackid Command Output	128
Example 3-95	select Command Output	130
Example 3-96	select subchannel <a b c d> Command Output	130
Example 3-97	select <rack>.<slot> Command Output	130
Example 3-98	select filter on Command Output	130
Example 3-99	serial Command Output	131
Example 3-100	serial set <str> Command Output	131
Example 3-101	shell Command Output	131
Example 3-102	smp Command Output	132
Example 3-103	sysname Command Output	133
Example 3-104	sysname <str> Command Output	133
Example 3-105	version Command Output	134

About This Guide

This guide describes how to use the L1 and L2 controller commands at your system console to monitor and manage the following systems:

- SGI Altix 330 systems
- SGI Altix 350 systems
- SGI Altix 3000 series systems
- SGI Altix 450 system
- SGI Altix 4700 system

Note: The L1 and L2 controller commands do not apply to SGI Altix XE systems. This manual supports the Altix 3000 series , Altix 450, and Altix 4700 systems. For information about SGI legacy systems, see an earlier version of this manual. From the current 007-3938-006 version of this manual on the SGI Technical Publications Library, select the **additional info** link. Click on **007-4633-005** under **Other Versions** :

The following topics are covered in this guide:

- Chapter 1, “Introducing the L1 and L2 Controllers,” describes the function of the L1 and L2 controllers.
- Chapter 2, “Using the L1 and L2 Controllers,” describes how to use the L1 and L2 controller to monitor and manage the systems.
- Chapter 3, “L1 and L2 Controller Commands,” describes the L1 and L2 system controller commands.

Related Publications

The following publications contain additional information that may be helpful:

- *SGI Altix 330 System User's Guide* provides an overview of the Altix 350 system components, and it describes how to set up and operate this system.
- *SGI Altix 350 System User's Guide* provides an overview of the Altix 350 system components, and it describes how to set up and operate this system.
- *SGI Altix 350 Quick Start Guide* provides a guide for a knowledgeable user through the installation, setup, and simple configuration of most SGI Altix 350 systems.
- *SGI Altix 3000 User's Guide* provides an overview of the architecture and descriptions of the major components that make up the SGI Altix 3000 computer 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 3700 Bx2 User's Guide* provides an overview of the architecture and descriptions of the major components that compose the SGI Altix 3700 Bx2 family of servers. It also provides the standard procedures for powering on and powering off the system, basic troubleshooting information, and important safety and regulatory specifications.
- *SGI Altix 450 System User's Guide* provides an overview of the architecture and descriptions of the major components that make up the SGI Altix 450 computer 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 4700 System User's Guide* provides an overview of the architecture and descriptions of the major components that make up the SGI Altix 4700 computer 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 ProPack 5 for Linux Start Here* provides information about the SGI ProPack 5 for Linux releases including information about major new features, software installation, and product support.
- *SGI ProPack 4 for Linux Start Here* provides information about the SGI ProPack 4 for Linux releases including information about major new features, software installation, and product support.

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:

Convention	Meaning
<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.
user input	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.
<code>manpage(x)</code>	Man page section identifiers appear in parentheses after man page names.
GUI element	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
- Use the Feedback option on the Technical Publications Library Web page:
<http://docs.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
1200 Crittenden Lane, M/S 535
Mountain View, California 94043-1351

SGI values your comments and will respond to them promptly.

Introducing the L1 and L2 Control Levels

This manual describes L1 and L2 control levels on SGI Altix 3000 series, Altix 450, and Altix 4700 systems.

Note: This manual does not apply to SGI Altix XE systems. It supports the Altix 3000 series, Altix 450, and Altix 4700 systems. For information about SGI legacy systems, see an earlier version of this manual. From the current 007-3938-006 version of this manual on the SGI Technical Publications Library, select the **additional info** link. Click on **007-4633-005** under **Other Versions:**

L1 Controller

The L1 controller is designed to manage and monitor the individual bricks or blades in SGI Altix systems.

Note: The L1 and L2 controller commands do not apply to SGI Altix XE systems.

The basic building block for SGI Altix 3000 series systems is the NUMALink interconnect and the C-brick, which is sometimes referred to as the compute node. A C-brick contains two processor nodes. Other bricks, such as the PX-bricks and IX-bricks, provide I/O functionality, M-bricks provide additional memory, R-bricks are router bricks allowing systems to scale are some examples.

In the new SGI Altix 4000 series systems, functional blades - interchangeable compute, memory, I/O, and special purpose blades in an innovative blade-to-NUMAlink architecture are the basic building blocks for the system. Compute blades with a bandwidth configuration have one processor socket per blade. Compute blades with a density configuration have two processor sockets per blade. Cost-effective compute density is one advantage of this compact blade packaging. The SGI hardware manuals

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

Depending on your system configuration, you can monitor and operate the system using the L1 controller from either the L1 LCD display or the system console serial port.

See Chapter 2, “Using the L1 and L2 Controllers,” for details on using the L1 controller, and see Chapter 3, “L1 and L2 Controller Commands,” for a list of L1 controller commands.

The following are some of the functions performed by the L1 controller. (For a list of functions that are specific to a product or component, see your server user’s guide.

- Drives the L1 controller display.
- Reports all failure conditions.
- Monitors and controls LEDs.
- Monitors the power On/Off switch.
- Monitors the reset switch, and the nonmaskable interrupt (NMI) switch.
- Monitors and acts on the state of the power, reset, and NMI switches.
- Drives the reset, soft reset, and NMI signals to the local node electronics on command from software.
- Provides the time of day (TOD).
- Reads and reports memory and processor configuration within the node. Reads dual inline memory module (DIMM) and system identification (ID) PROMs.
- Reads and writes contents of module identification EEPROMs.
- Controls voltage regulator modules (VRMs).
- Monitors voltage and reports failures.
- Controls and sequences DC power output of voltage regulators. Monitors failures and reports them. Drives DC on and failure LEDs.
- Controls voltage margining within the brick or server.
- Controls and monitors fan speed and reports the number of the failed fan.
- Monitors and reports operating temperature and status of 48 VDC input power and DC power.

- Provides a high-speed serial channel to communicate between the system controller network and the hub ASIC.
- Provides a high-speed serial channel to communicate with the L1 controller as a second brick or server. This is implemented as an RS-485 ICMB interface in the NUMAflex cable.
- Controls Joint Test Action Group (JTAG) boundary scan.
- Logs information in Non-Volatile Random Access Memory (NVRAM).
- Provides a USB slave port to communicate with upstream system controllers.
- Provides an external high-speed serial console port.
- Provides the ability to flash the system controller firmware, which can be updated.
- Reports the population of the PCI cards and the power levels of the PCI slots.
- Powers on the PCI slots and their associated LEDs.
- Monitors the slot power for PCI boards (currently takes no action).
- Reports the power levels of the XIO slots.
- Controls the termination voltage margins of the XIO cards.

L2 Controller

The L2 controller is designed to monitor and manage one or more L1 system controllers in SGI Altix systems.

Note: The L1 and L2 controller commands do not apply to SGI Altix XE systems.

All configurations can be monitored and or operated using the L2 controller using the L2 network port.

See Chapter 2, “Using the L1 and L2 Controllers,” for details on using the L2 controller, and see Chapter 3, “L1 and L2 Controller Commands,” for a list of L2 controller commands you can use to manage and monitor the different systems. The following are functions performed by the L2 controller. For a list of functions specific to a product or component, see your system user’s guide (user’s guides for SGI Altix systems are listed in “Related Publications” on page xxi).

- Controls the L1 controllers on the server.
- Maintains controller configuration and topology information between multiple L1 and L2 controllers.
- Enables the entering of commands to toggle between L2, L1, and the console mode. This means you can power on your server with L1 or L2 controller commands, and monitor the power-on activity by changing to the console mode.

In a system with more than one L2 controller, all L2 controllers are peers and each propagates configuration information to the other L2 controllers. Each L2 controller monitors its associated L1 controllers and propagates this information to the other L2 controllers.

System Control on SGI Altix 450 and Altix 4700 Systems

System control on SGI Altix 450 and Altix 4700 systems has changed from previous SGI Altix systems. Their system enclosures contain an embedded microprocessor board and display assembly called the system controller. This microprocessor runs an embedded version of the Linux operating system. The system controller runs off standby power and is running as long as the enclosure is connected to an active power source.

There are two primary applications that run on the system controller. The L1, or Level 1 system controller, is an application that provides control and monitoring functionality for each individual rack unit enclosure (IRU), and communication to other L1s in adjacent enclosures connected via NUMALink 4 cables. The L1 is always resident.

The L2, or Level 2 system controller is an application that provides control over multiple L1s and communication to other L2s. The L2 is resident when the enclosure is connected by an Ethernet connection to a Local Area Network (LAN).

In all Altix 450 and 4700 servers, all the system controllers communicate with each other in the following ways:

- All enclosures within an Altix 450 or Altix 4700 system communicate with each other through the NUMALink connections using low voltage differential signaling (LVDS).
- When connecting to the L2 host Ethernet connection on the system control board of an IRU or to a Dense router, the system controller spawns an L2 application providing L2 functionality.

The system controller network provides the following functionality:

- Powering the entire system on and off.
- Powering individual IRUs and Dense routers on and off.
- Monitoring the environmental state of the system
- Viewing the system's status and error message information generated and displayed by the SGI system's L1 controller.

For detailed information on system control on an SGI Altix 450 system, see chapter 4, "System Control" in the *SGI Altix 450 System User's Guide*.

For detailed information on system control and the levels of control available on an SGI Altix 4700 system, see chapter 1, "Operation Procedures" and chapter 2, "System Control" in the *SGI Altix 4700 System User's Guide*.

Using the L1 and L2 Controllers

This chapter describes how to use the L1 and L2 controllers to monitor and manage an SGI Altix system in the following sections:

- “Monitoring Feedback and Entering Commands” on page 8
- “Operating the L2 Controller” on page 8
- “Operating the L2 Controller on Altix 450 and 4700 Systems” on page 13
- “Operating the L1 Controller on Altix 3000 Systems” on page 23
- “Operating the L1 Controller on Altix 450 and Altix 4700 Systems” on page 30
- “Upgrading System Controller Firmware” on page 35
- “Upgrading Prom Firmware” on page 36
- “Identifying Bricks” on page 39
- “Identifying Blades” on page 40
- “System Partitioning” on page 44
- “Hyper-Threading on Altix 450 or Altix 4700 Systems” on page 45
- “Status and Error Messages” on page 46

Note: Most, but not all, of the L1 controller commands can be used with all systems. You may also find that some specific examples using L1 controller commands in this chapter may show components not applicable to your system, but this does not indicate that these commands cannot be used to monitor and manage your system.

Monitoring Feedback and Entering Commands

The LCD on the front of certain enclosures can be used to provide limited control and monitoring capabilities. The LCD provides the functionality to power up, power down, and reset the system. See the appropriate SGI hardware manual for your system for a detailed description of the LCD.

A serial or network connection to the system controller network allows you to control and monitor your system using a command line interface. Additionally, this connection also accesses the system console, providing an interface to boot your system.

Note: See “Upgrading System Controller Firmware” on page 35 for instructions to upgrade your L1 controller firmware and L2 controller firmware.

See Chapter 3, “L1 and L2 Controller Commands,” for a list of L1 and L2 controller commands you can use to monitor and manage the various devices.

Operating the L2 Controller

The L2 controller operates in one of the following three modes, each of which is discussed in the sections that follow. For information on operating the L2 controller on the new SGI Altix 4700 system, see “Operating the L2 Controller on Altix 450 and 4700 Systems” on page 13.

- The L2 prompt is visible and all input is directed to the L2 command processor (see “L2 Mode” on page 8)
- Output from the system is visible and all input is directed to the system (“Console Selection” on page 9).
- The prompt from a single L1 is visible, and all input is directed to that L1 command processor (see “L1 Mode from L2” on page 12.)

L2 Mode

After a connection to the L2 controller, a variation of the following prompt appears, indicating that the L2 is ready to accept commands:


```
L2>
```

When a system is partitioned, the L2 supports targeting a command to a single partition. To set the destination to a single partition precede the targeting information with the “partition <number>”, such as partition 1. This sets the default destination to all bricks in partition 1.

Another option is to use the L2 controller select partition <number> command, such as the following:

```
L2> select partition 1
```

This sets the default destination to all the bricks in partition 1 and also selects the correct system console for partition 1, used when the L2 controller is put in system console mode.

The L2 controller provides a fail-safe mechanism for certain commands that could produce undesirable results when you try to apply certain commands to a partition.

Command Interpretation

Some L2 commands are the same as the L1 commands. In many cases, this is intentional because the L2 provides sequencing that is necessary for a command to function correctly.

When L1 and L2 commands are similar, you can assure that an L1 command is entered for the bricks in the current destination by preceding <L1 command> with the L1 command (this is a one-time destination), as follows:

```
L2> r 2-4,7 11 <L1 command>
```

Console Selection

The following are common subchannels associated with console communications on an SGI Altix system:

- Subchannel 0A specifies node | blade 0, CPU A.
- Subchannel 0C specifies node | blade 0, CPU C.
- Subchannel 1A* specifies node | blade 1, CPU A.
- Subchannel 1C specifies node | blade 1, CPU C.

- Subchannel 2A specifies node | blade 2, CPU A.
- Subchannel 2C specifies node | blade 2, CPU C.
- Subchannel 3A specifies node | blade 3, CPU A.
- Subchannel 3C specifies node | blade 3, CPU B.
- node | blade 0 console subchannel.
- node | blade 1 console subchannel.
- node | blade 2 console subchannel
- node | blade 3 console subchannel

On blade-based systems, the numeric part of the console descriptor refers to the blade slot, on non-blade based systems, it refers to the node. (There is only one node on a compute blade which is node 0.) The alphabetic portion refers to the CPU index which can be A, B, C, or D or “console” which identifies the appropriate CPU to receive console input.

*This can be A, B, C, or D, depending on the system.

To select a subchannel as the current subchannel, for a specific node/blade and CPU, perform the following:

```
L2> select sub 0a
```

or

```
L2> select sub 0c
```

or

```
L2> select sub 2a
```

and so on.

Note: Under normal conditions, it is not necessary to manually select a system console beyond selecting a partition on a partitioned system. The following instructions would be required in a debugging environment. To restore the console to the default value, use the `select reset` command.

For console subchannels, perform the following:

```
L2> select sub console0
```

or

```
L2> select sub console1
```

and so on.

The output `console input: 001c05 console0` shows that the L2 will send console input to brick 001c05 and the subchannel to be used is the console0 subchannel.

To change system console status from one brick to the attached C-brick, use the `select <rack> <slot>` command:

```
001c05-L2> select r 2 s 1  
console input: 001c05 console1  
console output: not filtered.  
001c05-L2>
```

When selecting the rack and slot for the console, you could also select `<rack>.<slot>`, as follows:

```
001c05-L2> select 2.4
```

To change the subchannel used on the selected brick, use the `select` command followed by the subchannel number or the word `console`:

```
001c05-L2> select sub 0A
console input: 001c05 CPU 0A
console output: not filtered.
001c05-L2>
```

L1 Mode from L2

In L1 mode, the prompt from a single L1 is visible, and all input is directed to that L1 command processor.

To enter L1 mode, type the `l1` command and specify a rack and a slot, as follows:

```
L2> r 2 s 1 l1
entering L1 mode 002c01, <CTRL-T> to escape to L2

002c01-L1>
```

A shorter command line version is, as follows:

```
L2> 2.111
entering L1 mode 002c01, <CTRL-T> to escape to L2

002c01-L1>
```

To return to L2 mode, press `Ctrl+T`, as follows:

```
002c01-L1> Ctrl+T
escaping to L2 system controller, <CTRL-T> to send escape to L1
L2>
```

At this point, you can enter any L2 command. When the command completes execution, the L2 returns to L1 mode, as follows:

```
002c01-L1>
```

To permanently engage the L2 mode, press `Ctrl+T` and type the `l2` command, as follows:

```
002c01-L1> Ctrl+T
escaping to L2 system controller, <CTRL-T> to send escape to L1
L2> l2
L2 command processor engaged, <CTRL-T> for console mode.
L2>
```

Operating the L2 Controller on Altix 450 and 4700 Systems

The system controller in an Altix 450 or Altix 4700 system can provide L2 functionality.

An Ethernet cable can be plugged into the RJ45 connector on the IRU enclosure or on the Dense router. Connecting the IRU or Dense router to an active LAN via the L2 host connector will cause the system controller to spawn an L2. This connection provides network access to the system controller through the L2. For information on setting up system control on an SGI Altix 4700 system, see “System Controller Network” in chapter 1, “Operation Procedures” in the *SGI Altix 4700 System User’s Guide*.

Caution: The L2 network connection must be kept on a private and physically secure network connection. The System Controller does not provide software security, authentication, or safeguards against malicious or careless users. The consequences of unauthorized access to the System Controllers is potential system unavailability, system reconfiguration, and detailed access to every aspect of the system. The recommended configuration for the L2 network connection is a separate, private and physically secure network attachment to an equally separate and physically secure PC. The Linux system hosting the L3 software must be administered to prevent unauthorized access.

This section describes L2 operation on SGI Altix 4700 systems and covers these topics:

- “Configuring an L2’s IP Address on an Altix 450 or Altix 4700 System” on page 14
- “Viewing System Configuration on Altix 450 or Altix 4700 System” on page 16
- “Setting Command Targeting on Altix 450 or Altix 4700 System” on page 17
- “Viewing Information, Warnings, and Error Messages on Altix 450 or Altix 4700 Systems” on page 19
- “Powering On, Powering Off, and Resetting the System From the L2” on page 19

- “Console Mode from the L2 on Altix 450 or Altix 4700 Systems” on page 20
- “Console Selection on Altix 450 or Altix 4700 Systems” on page 21
- “L1 Mode From L2 on Altix 450 or Altix 4700 Systems” on page 23

Configuring an L2's IP Address on an Altix 450 or Altix 4700 System

This section refers to setting the IP address on the individual rack unit (IRU) enclosure when using an Ethernet connection.

Setting the IP address of the L2 on the target IRUs should be done before connecting the IRUs to the network as follows:

Connect a serial cable to the serial console port on the target IRU and get the Linux shell prompt.

To see if the L2 is running (it will be if the LAN cable is plugged in and the cable is connected to an active LAN).

At the # prompt type:

```
# llcmd l2
L2 Controller is running.
#
```

If the L2 is not running type:

```
# init 4
```

This switches the system controller to run level 4 and forces the L2 to be started whether or not the LAN is plugged in.

Verify the L2 is running again as above.

To set the IP address on the L2 type:

```
#llcmd l2 ip a.b.c.d 255.255.255.0 a.b.c.255
```

To set the L2 controllers gateway, perform the following command:

```
#llcmd l2 IP gw e.f.g.h
```

Verifying that the system serial number is set on the L2:

```
#l1cmd l2 serial
```

To set the L2 system serial number:

```
#l1cmd001c01-L1> l2 serial set <serial number>
```

Verifying that `msys` is enabled (this allows multiple L2s in a system to exist peacefully with other L2s from another system on the same subnet)

```
#l1cmd l2 multisys
```

If `msys` is off, turn it on:

```
#l1cmd l2 multisys on
```

Note: `msys` is an abbreviated version of the `multisys` command (see “`multisys`” on page 124).

Reboot the system controller to make the IP address change take effect.

```
#reboot
```

Once this is done for all target IRUs and/or Dense routers (sometimes called the Quad Dense Router (QDR)) as specified in the system configuration guide, connect them to the network (using an optional Ethernet switch if necessary). For descriptions of individual rack units (IRUs) and Dense routers on an Altix 4700 system, see *SGI Altix 4700 System User's Guide*.

The L1 rack ID is rack x 100 + slot (rack times 100 plus slot). An example L2 `rackid` is 5 * 100 + 41 or 541.

Once the L2 is running, you can telnet to the L2, or use an optional SGI console.

After the connection to the L2 controller is established, the following prompt appears, indicating that the L2 is ready to accept commands:

```
olympic-101-L2>
```

Common operations are discussed in the subsections that follow.

Viewing System Configuration on Altix 450 or Altix 4700 System

You can use the L2's `config` command to view the your system configuration from an IRU level:

```
olympic-101-L2> config
L2 127.0.0.1: - 001 (LOCAL)
L1 127.0.0.1:0:0 - 001c31
L1 127.0.0.1:0:1 - 001c21
L1 127.0.0.1:0:2 - 001c11
L1 127.0.0.1:0:3 - 001c01
L2>
```

As shown above, `config` produces a list of IRUs and their locations in the system and the system controller address of each IRU and Dense router. This is similar to the output from using the `config` command on the L1 with the addition of the L2 IP address, L1 connection, and L1 index. The structure of the IRU and Dense router address is as follows:

a.b.c.d:x:y

where:

a.b.c.d is the IP address of the L2. (In the example above, the IP address is 127.0.0.1.)

x connection number is only 0 for Altix 4700

y is the L1 index

rrrbss.p

where:

rrr is the rack number.

b is the enclosure type.

ss is the slot location of the enclosure.

p is the partition of the enclosure (not present if the system is not partitioned).

In the example shown above, 001c01 is an IRU in rack 001 and slot position 01.

Setting Command Targeting on Altix 450 or Altix 4700 System

If a command is not understood by the L2 system controller, in general it is passed to the L1 system controllers. The destination determines which L1s receive the command. A destination, specified by the following, is a range of racks and slots:

```
rack <rack list> slot <slot list>
```

The <rack list> specifies a list of racks. This can be a list delimited by commas, such that 2,4,7 specifies racks 2, 4, and 7. You can use a dash to specify a range of racks, such that 2-4 specifies racks 2, 3, and 4. Both nomenclatures can be combined, such that 2-4,7 specifies racks 2, 3, 4, and 7.

You can specify the <slot list> using the same nomenclature. The slot number, sometimes referred to as a bay number, is the unit position number located on the rack, slightly above where the bottom of the IRU sits. Each rack unit position number is located toward the top of the two lines that mark the unit position that the number represents. For example, the rack numbering for a IRU located in slot 10 would appear on the left front side of the rack.

The slot <slot list> is optional; if not given, then all slots in the specified rack(s) are implied. You should avoid specifying a rack list and a slot list that includes multiple racks and slots, such as rack 2-4,7 slot 1-8,11,13. Generally, you specify a rack and slot together to specify an individual IRU or Dense router.

You can use the aliases *r* and *s* to specify *rack* and *slot*, respectively. You can use the alias *all* or *** in both the <rack list> and the <slot list>, or by themselves, to specify all racks and all slots.

To send a command to all IRUs in a partition, enter the following:

```
partition <partition> <cmd>
```

Individual IRU and Dense routers can also be targeted with a short <rack>.<slot> prefix. As in 1.11 <command>.

To target individual blades in an IRU use the following syntax:

```
olympic-101-L2> 1.11 b1 power down
```

Executing the above command will power down the blade in blade slot 1 of the IRU in rack 001 U position 11.

Default Destination

When the L2 starts, the default destination is set to all racks and all slots. You can determine the default destination by using the `destination` command:

```
L2> destination
all racks, all slots
L2>
```

The following command sets the destinations to rack 2 and 3, all slots:

```
L2> r 2,3 destination
2 default destination(s) set
L2>
```

The following example shows what IRUs are found in the default destination. If you enter a command not understood by the L2, the command is sent to these IRUs.

Note: If you add an IRU to either rack 2 or 3, it is not automatically included in the default destination. You would need to reset the default destination.

```
L2> destination
001c01 (127.0.0.1:0:2)
001c01 (127.0.0.1:0:0)
L2>
```

The following command resets the default destination to all racks and all slots:

```
L2> destination reset
default destination reset to all racks and slots
L2>
```

Current Destination

The current destination is a range of racks and slots for a given command. For example, the following command sends the command `<L1 command>` to all IRUs in racks 2, 3, 4, and 7:

```
L2> r 2-4,7 <L1 command>
```

This is a one-time destination.

Command Interpretation

Some L2 commands are the same as the L1 commands. In many cases, this is intentional because the L2 provides sequencing that is necessary for a command to function correctly.

When L1 and L2 commands are similar, you can ensure that an L1 command is entered for the IRUs current destination by preceding the command `<L1 command>` with the `l1` command:

```
L2> r 2-4,7 l1 <L1 command>
```

This is a one-time destination.

Viewing Information, Warnings, and Error Messages on Altix 450 or Altix 4700 Systems

All information, warnings, and error messages generated by any of the system controllers are in the following form:

```
001c01 ERROR: invalid arguments for 'ver' command, try "help ver"
```

The general format includes an IRU identification and the type of message, followed by the message. A message may be the result of an invalid command, as shown in the example, or the result of tasks running on the L1, such as the environmental monitor.

Each L1 has a log of local events. Use the L1 command `log` to view events on any of the L1s.

Powering On, Powering Off, and Resetting the System From the L2

You can power on and power off the system with the `power` command. This command is interpreted by the L2, because the IRUs must be powered on in a specific order.

```
L2> power up  
L2>
```

The `power` command may require several seconds to several minutes to complete. In the example above, all racks and slots in the default destination are affected. Any errors or warnings are reported as described above in TBD. To power on or power off a specific IRU, specify a current destination:

```
L2> r 2 s 5 power up  
L2>
```

To power on or power off all IRUs in a partition, enter the following:

```
L2> partition <partition number> <power up or power down>
```

To reset the system, enter the following:

```
L2> reset
L2>
```

This command restarts the system by resetting all registers to their default settings and rebooting the system controllers. Resetting a running system will cause the operating system to reboot and all data in memory will be lost.

Console Mode from the L2 on Altix 450 or Altix 4700 Systems

In console mode, all output from the system is visible and all input is directed to the system.

To enter console mode from L2, press **Ctrl+D** at the L2 prompt and observe the response:

```
L2> Ctrl+D
entering system console mode (001c01 console0),
<CTRL_T> to escape to L2
.
<system output appears here>
.
```

To return to L2 mode from console mode, press **Ctrl+T**:

```
Ctrl+T
escaping to L2 system controller
L2>
```

At this point, you can enter any L2 or L1 command. When the command completes, the L2 returns to console mode:

```
Re-entering system console mode (001c01 console0),
<CTRL_T> to escape to L2
```

To permanently engage the L2 mode, press **Ctrl+T** and then enter the **l2** command:

```
Ctrl+T
escaping to L2 system controller
L2> l2
```

```
L2 command processor engaged, <CTRL_D> for console mode.  
L2>
```

Console Selection on Altix 450 or Altix 4700 Systems

When in console mode, the L2 communicates with the IRU set with the `select` command to be the system console or global master. All input from the console is directed to that IRU. You can set and view the system console with the `select` command.

The L2 chooses an IRU as the default console in the following order of priority:

- The IRU in the lowest numbered rack and slot, which has previously produced console output.
- The IRU in the lowest numbered rack and slot.

The `select` command by itself shows the current console mode settings:

```
L2> select  
known system consoles (non-partitioned)  
  
    001c01-L2 detected  
  
current system console  
  
console input: 001c01 CPU 0A  
console output: not filtered
```

The following are ten common subchannels associated with console communications:

- Subchannel 0A specifies Blade 0, CPU A.
- Subchannel 0C specifies Blade 0 CPU C.
- Subchannel 1A specifies Blade 1, CPU A.
- Subchannel 1C specifies Blade 1, CPU C.
- Subchannel 2A specifies Blade 2, CPU A.
- Subchannel 2C specifies Blade 2, CPU C.
- Subchannel 3A specifies Blade 3, CPU A.
- Subchannel 3C specifies Blade 3, CPU C.

- Subchannel console0 Blade 0 console subchannel.
- Subchannel console1 Blade 1 console subchannel.

The select command output: “console input: 001c01 console0” shows that the L2 will send console input to IRU 001c01 blade 0 and the console subchannel will be used.

To change the IRU that will be the system console, use the `select <rack>.<slot>` command, where `<rack>` is the rack and `<slot>` is the slot where the IRU is located:

```
L2> select 1.1
console input: 001c01 console
console output: no filtered
console detection: L2 detected
```

To change the subchannel used on the selected IRU, use the `select subchannel <0A|0C|1A|1C>` command. (Use the `select subchannel console` to select the current console as the subchannel of the IRU to be the system console.) For example, to select blade 1, CPU A as the subchannel of the IRU to be the system console, enter the following:

```
L2> select subchannel 1A
console input: 001c01 console CPU1A
console output: not filtered
```

During the boot process on a system with multiple CPUs, there is a window of time in which the CPUs are all producing output. This can result in a somewhat jumbled output from the L2. However, you can filter console output so that the L2 will show output from only the IRU chosen to receive console input. You can turn on filtering with the `select filter on` command and turn off filtering with the `select filter off` command.

If you attempt to communicate with an IRU chosen to receive console input but that is not responding, a time-out condition results:

```
L2> Ctrl+D
entering console mode 001c01 CPU1A, <CTRL_T> to escape to L2

no response from 001c01 Junk bus CPU1A system not responding
no response from 001c01 Junk bus CPU1A system not responding
```

When this time-out condition occurs, either the IRU is hung or the subchannel is not correct.

L1 Mode From L2 on Altix 450 or Altix 4700 Systems

In L1 mode, the prompt from a single L1 is visible, and all input is directed to that L1 command processor.

To enter L1 mode, enter the rack and a slot followed by l1 :

```
L2> r 2 s 1 l1
```

An alternate method is:

```
L2> 2.1 l1
```

entering L1 mode 001c01, <CTRL-T> to escape to L2

```
001c01-L1>
```

To return to L2 mode, press Ctrl+T:

```
001c01-L1> Ctrl+T
```

escaping to L2 system controller, <CTRL-T> to send escape to L1

```
L2>
```

At this point, you can enter any L2 command. Once the command is executed, the L2 returns to L1 mode:

re-entering L1 mode 001c01, <CTRL-T> to escape to L2

```
001c01-L1>
```

To permanently engage the L2 mode, press Ctrl+T and enter the l2 command:

```
001c01-L1> Ctrl+T
```

escaping to L2 system controller, <CTRL-T> to send escape to L1

```
L2> l2
```

L2 command processor engaged, <CTRL-T> for console mode.

```
L2>
```

Operating the L1 Controller on Altix 3000 Systems

This section describes the L1 controller on SGI Altix 3000 series systems.

The L1 controller operates in one of the following two modes, each of which is discussed in the sections that follow:

- **L1 mode.** The L1 prompt is visible and all input is directed to the L1 command processor.
- **Console mode from L1 mode.** Output from the system is visible and all input is directed to the system.

Note: The console mode from L1 mode is not supported if the system contains an L2 controller.

For information on operating the L1 controller on an SGI Altix 450 and Altix 4700 systems, see .

L1 Mode

The brick with which the L1 communicates in console mode is the system console or global master, and you can view and set it with the `select` command. By default, the C-brick attempts to communicate with its local CPUs when console mode is entered. If the system has been powered on and either one of the bricks received a request to be the system console, then the C-brick attempts to communicate with that brick.

When you see a prompt of the following form, the L1 is ready to accept commands.

```
001c19-L1>
```

Common operations include the following and are discussed in the sections that follow:

- “Viewing System Configuration” on page 24
- “Command Targeting” on page 27
- “Viewing Information, Warnings, and Error Messages” on page 27
- “Powering On, Powering Off, and Resetting the Brick” on page 28

Viewing System Configuration

An L1 has limited knowledge of the system configuration. A C-brick only has information about its attached I/O brick and, if another C-brick is attached to it,

information about that C-brick and its attached I/O brick. An I/O brick only has information about its attached C-brick. An R-brick only has information about itself.

You can view a brick's configuration information with the `config` command, as follows:

```
003c01-L1> config
:0 - 003c01
:1 - 004i01
:2 - 002c01
:3 - 001x01
003c01-L1>
```

Bricks are referenced by their racks and slot or bay locations. These values are stored in nonvolatile memory on the L1. Virtually all system controller communications require that each brick has a valid and unique rack and slot.

If a brick is not set with its rack and slot number, it appears in the output of an L2 `config` command, as shown in the following example:

```
L2> config
137.38.88.82.1.0 ---c-- (no rack/slot set)
L2>
```

To set the rack and slot for a brick, address it by its IP address, USB port, and L1 controller index. Note the following example:

```
L2> 137.38.88.82:1:0 brick rack 1
L2> 137.38.88.82:1:0 brick slot 8
L2> 137.38.88.82:1:0 reboot_l1
INFO: closed USB /dev/sg11_0
INFO: opened USB /dev/sg11_0
L2>config
137.38.88.82:1:0 001c08
L2.
```

The following example shows how to set rack 1, slot 8, for the C-brick with an IP address of 127.0.0.1:

```
L2> config
127.0.0.1:
127.0.0.1:0:0 - ---c--
127.0.0.1:0:1 - 001i01
127.0.0.1:0:5 - 001c05
L2> :0:0 brick rack 1
brick rack set to 001.
```

```
L2> :0:0 brick slot 8
brick slot set to 08.
L2> :0:0 reboot_l1
INFO: closed USB /dev/sgill_0
INFO: opened USB /dev/sgill_0
L2>
L2> config
127.0.0.1:
127.0.0.1:0:0 - 001c05
127.0.0.1:0:1 - 001i01
127.0.0.1:0:5 - 001c08
L2>
```

To set the rack and slot from the L1 prompt, simply use the `brick rack` and `brick slot` commands. To set the rack and slot on one of the attached bricks (an attached I/O brick, C-brick, or a C-brick's I/O brick), use the L1 targeting commands `iia`, `iib` or `nia`, `nib`, as follows:

```
001c05-L1> config
:0 - 001c05
:1 - ---i--
:5 - 001c08
:6 - 001p01
001c05-L1> iia brick rack 4
---i--:
brick rack set to 004.
001c05-l1> iia brick slot 1
---i--
brick slot set to 01
001c05-l1> iia reboot_l1
001c05 ERROR: no response from ---i--
001c05-L1> config
:0 - 001c05
:1 - 004i01
:5 - 001c08
:6 - 001p01
001c05-L1>
```

In the preceding code, the number after the colon symbol (:) indicates the following:

```
0 = local brick
1 = IIA (II0)
2 = IIB (II1)
5 = NIA (N10)
10 = NIB (N11)
```

Only 0 has a valid meaning, other values are arbitrary based on the system type.

To obtain a detailed configuration explanation from the L1 perspective, enter the following:

```
001c05-L1> config verbose
```

Command Targeting

All commands affect only the local brick, unless the command is prefixed with an asterisk (*). To target a command to all bricks (including the local brick), prefix the command, as shown in the following example:

```
003c01-L1> * version
003c01:
L1 0.7.37 (Image A), Built 05/24/2001 14:59:42 [P1 support]
004i01:
L1 0.7.37 (Image A), Built 05/24/2001 14:59:42 [P1 support]
002c01:
L1 0.7.37 (Image A), Built 05/24/2001 14:59:42 [P1 support]
001x01:
L1 0.7.37 (Image A), Built 05/24/2001 14:59:42 [P1 support]
003c01-L1>
```

The L1 also supports a L1> <rack>.<slot> <command> target. For example,

```
003c01-L1> 4.1 version
004i01:
L1 0.7.37 (Image A), Built 05/24/2001 14:59:42 [P1 support]
003c01-L1>
```

You can also target commands to a single attached brick with either the *ia*, *ib*, or *nia*, *nib* command, as follows:

```
003c01-L1> ia version
004i01:
L1 0.7.37 (Image A), Built 05/24/2001 14:59:42 [P1 support]
003c01-L1>
```

Viewing Information, Warnings, and Error Messages

All information, warnings, and error messages generated by any of the system controllers are in the following form:

```
002c01 ERROR: invalid arguments for 'ver' command, try "help ver"
```

The general format of the message includes a brick identification (this is not present if the command was to the local brick only), type of message, and the message. These messages can be the result of an invalid command (as shown in the example) or from tasks running on the L1, such as the environmental monitor.

Each L1 has a log of local events. Use the L1 command `log` to view the event on any of the L1s.

Powering On, Powering Off, and Resetting the Brick

You can power on and power off the brick with the `power` command, as follows:

```
003c01-L1> power up  
003c01-L1>
```

If an L2 is not present, you need to power on, power off, and reset the system from one of the C-bricks. You do so by targeting all bricks, as follows:

```
003c01-L1> * power up  
003c01-L1>
```

This command can require from several seconds to several minutes to complete.

You can enter the `power off` and `reset` commands in similar ways.

Console Mode from L1

In console mode, output from the system is visible and all input is directed to the system.

To enter console mode, press `Ctrl+D` at the L1 prompt, as follows:

```
003c01-L1> Ctrl+D  
entering console mode 003c01 console, <CTRL-T> to escape to L1  
.  
<system output appears here>  
.
```

To return to L1 mode, press `Ctrl+T`, as follows:

```
Ctrl+T  
escaping to L1 system controller
```

```
003c01-L1>
```

At this point, you can enter any L1 command. When the command completes execution, the L1 returns to console mode, as follows:

```
re-entering console mode 003c01 console, <CTRL-T> to escape to L1
```

To permanently engage the L1 mode, press **Ctrl+T** and then type the **l1** command, as follows:

```
Ctrl+T
```

```
escaping to L1 system controller
```

```
003c01-L1> l1
```

```
L1 command processor engaged, <CTRL-D> for console mode.
```

```
003c01-L1>
```

Console Selection on Systems

The brick with which the L1 communicates in console mode is the system console or global master, and you can view and set it with the `select` command. By default, the C-brick attempts to communicate with its local CPUs when it enters console mode. If the system has been powered on and either one of the bricks has a request to be the system console, then the C-brick attempts to communicate with that brick. Enter the `select` command alone to show the current console mode settings, as follows:

```
003c01-L1> select
```

```
console input: 003c01 console
```

```
console output: not filtered.
```

The following are common subchannels associated with console communications on an SGI Altix system:

- Subchannel 0A specifies Node 0, CPU A.
- Subchannel 0C specifies Node 0, CPU B.
- Subchannel 1A specifies Node 1, CPU A.
- Subchannel 1C specifies Node 1, CPU B.
- Subchannel 2A specifies Node 2, CPU A.
- Subchannel 2C specifies Node 2, CPU B.
- Subchannel 3A specifies Node 3, CPU A.
- Subchannel 3C specifies Node 3, CPU B.
- Node 0 console subchannel.
- Node 1 console subchannel.
- Node 2 console subchannel
- Node 3 console subchannel

Operating the L1 Controller on Altix 450 and Altix 4700 Systems

Each IRU and Dense router in the SGI Altix 450 or Altix 4700 system has an updated and enhanced system control implementation. This updated system controller provides both L1 and L2 functionality. The system controller utilizes an embedded version of the Linux operating system. L1 functionality is provided by an application that is always running on the system controller. When the enclosure is connected to a LAN via the L2 host connector, the system controller spawns an application that provides L2 functionality.

The L1 operates in one of these two modes, which are discussed in the sections that follow:

- **"L1 Mode>"**
- **"Direct Connection to an L1"**

Note: “Direct Connection to an L1” is supported only if the system console L1 port is connected directly to the console system (laptop, PC, etc.).

L1 Mode>

The L1 prompt is visible and all input is directed to the L1 command processor. The Altix 4700 server L1 system control can perform the following:

- Managing power and sequencing control
- Environmental monitoring and control functions
- Initiation of system resets
- Read/write storage for identification and configuration information
- Provides console/diagnostic and scan interface

The L1 controller in each of the enclosures is a complete and fully functional system controller. All the blades are interconnected via a low-voltage differential (LVD) signal integrated into the NUMAlink cable and each shares its system control information with all other system controllers over this connection.

Direct Connection to an L1

Output from the system is visible and all input is directed to the system console.

Note: “Direct Connection to an L1” is supported only if the L1 controller is accessed from the Console L1 port via a serial cable connection.

L1 Mode on Altix 450 and Altix 4700 Systems

If you see a prompt of the following form, the L1 is ready to accept commands.

```
001c01-L1>
```

Common operations are discussed in the following sections:

- “Viewing System Configuration (from an IRU’s Perspective)” on page 32
- “Command Targeting” on page 32

Viewing System Configuration (from an IRU’s Perspective)

An L1 has limited knowledge of the system topology, depending on the system’s configuration. Typically, an L1 has information only about L1s that are directly NUMALink connected.

In large configurations with more than one L1, the L1 may have knowledge of only a portion of the L1s in the system. These configurations require the use of the L2, see “Operating the L2 Controller on Altix 450 and 4700 Systems” on page 13 for further details.

You can view an IRU’s configuration information with the `config` command as in the following:

```
001c01-L1> config
:0 001c01 LOC
001c01-L1>
```

This example is a system with one IRU. The *<number>* that follows the colon (0, 1, and 2, from top to bottom in this example), refers to which local port the IRU is connected to or accessed through. The local (LOC) IRU is the IRU that is processing the command.

On all IRUs :0 is the local IRU, with other values referring to various ports. The specific port description follows the IRU’s rack/type/slot field: (i.e. LOC, U-F, U-G, etc.)

```
021c01-L1> config
:0 021c01 LOC
:2 021c11 L1H
:8 021r41 L0G
:5 022r41 R3G
021c01-L1>
```

Command Targeting

All commands entered affect only the local IRU. You can target a command to all IRUs (including the local IRU) by prefixing the command with an asterisk (*).

```
001c01-L1> * version
001c01:
L1 0.7.37 (Image A), Built 11/24/2005 14:59:42 [2MB image]
```



```
001c11:
L1 0.7.37 (Image A), Built 11/24/2005 14:59:42 [2MB image]
001c21:
L1 0.7.37 (Image A), Built 11/24/2005 14:59:42 [2MB image]
001c01-L1>
```

Commands can be targeted to other L1s by preceding the command with a rack and slot:

```
001c01-L1> 1.11 version
```

The command above issues a version command to IRU in rack 001, U position 11.

Some commands can be targeted to a specific blade within an IRU. Precede the command with the blade designator:

```
001c01-L1> b1 power down
```

The command above issues a power down command to the blade in blade slot 1 of the IRU in rack 001, U position 01.

Console Mode from L1 on Altix 450 or Altix 4700 Systems

In console mode, output from the system boot process or OS is visible and all input is directed to the system. To enter console mode, press **Ctrl+D** at the L1 prompt:

```
001c01-L1> Ctrl+D
entering console mode 001c01 console, <CTRL-T> to escape to L1
.
<system output appears here>
.
```

To return to L1 mode, press **Ctrl+T**:

```
Ctrl+T
escaping to L1 system controller
001c01-L1>
```

While in L1 mode, you can enter any L1 command. Once the command is executed, the L1 returns to console mode:

```
re-entering console mode 001c01 console, <CTRL-T> to escape to L1
```

To permanently engage the L1 mode, press **Ctrl+T** and then enter the **l1** command:

```
Ctrl+T
```

```
escaping to L1 system controller
001c01-L1> l1
L1 command processor engaged, <CTRL-D> for console mode.
001c01-L1>
```

L1 Console Selection on SGI Altix 450 and Altix 4700 Systems

If the system contains more than one IRU and a serial connection is utilized for the console, the serial cable must be connected to the IRU that is located in the lowest rack and slot position. You can use L2 commands to manage multiple L1 controllers for systems with multiple IRUs.

The `select` command shows the current console mode settings:

```
001c01-L1> select
console input: 001c01 console0
console output: not filtered
```

The following are common subchannels associated with console communications:

- Subchannel 0A specifies Node/Blade 0, CPU A.
- Subchannel 0C specifies Node/Blade 0 CPU C.
- Subchannel 1A specifies Node/Blade 1, CPU A.
- Subchannel 1C specifies Node/Blade 1, CPU C.
- Subchannel 2A specifies Node/Blade 2, CPU A.
- Subchannel 2C specifies Node/Blade 2, CPU C.
- Subchannel 3A specifies Node/Blade 3, CPU A.
- Subchannel 3C specifies Node/Blade 3, CPU C.
- Subchannel console0 Blade 0 console subchannel.
- Subchannel console1 Blade 1 console subchannel.

The output from the `select` command:

```
"console input: 001c01 console0"
```

shows that the system controller will send input to IRU 001c01 blade 0 and the subchannel to be used is the console subchannel.

During the boot process, there is a window of time in which all processors may be producing output. This output can produce a somewhat jumbled output at the L1.

However, you can filter the console output so that the L1 shows output from only the processor chosen to receive console input. You can turn filtering on and off with the `select filter` command.

If you attempt to communicate with a IRU that is not responding, a time-out condition results:

```
001c01-L1>
```

```
entering console mode 001c01 console, <CTRL-T> to escape to L1
no response from 001c01 junk bus console UART:UART_TIMEOUT
```

When this time-out condition occurs, either the IRU is hung or the subchannel is incorrect. An IRU is identified by its rack, type, and slot (001c01).

Viewing Information, Warnings, and Error Messages on Altix 4700 Systems

All information, warnings, and error messages generated by any of the system controllers are in the following form:

```
001c01 ERROR: invalid arguments for 'ver' command, try "help ver"
```

The general format includes a IRU identification and the type of message, followed by the message. A message may be the result of an invalid command, as shown in the example, or the result of tasks running on the L1, such as the environmental monitor.

Each L1 has a log of local events. Use the L1 command `log` to view events on any of the L1s.

Upgrading System Controller Firmware

There are two types of firmware to consider on your SGI Altix system, as follows:

- The L1 firmware; this image contains an embedded version of the Linux operating system and L1 and L2 applications and various support applications
- The system PROM (each node has a system PROM and each C-brick has two nodes).

On an SGI Altix 450 or SGI Altix 4700 system, the L1 firmware consists of three parts: the boot image, A image, and B image. For more information, see the “Upgrading L1 Firmware” section in either the *SGI Altix 450 User’s Guide* or *SGI Altix 4700 User’s Guide*.

The system controller firmware provides support for managing and monitoring the power, cooling, and testing functions for a brick and system compute rack.

You can always download the latest system controller firmware and PROM firmware via Supportfolio at <http://support.sgi.com/linux/>

You can also find system controller firmware by using the search term “system controller” at: http://support.sgi.com/search_request/downloads/index.

PROM is released as patches and the latest firmware can also be found at: http://support.sgi.com/browse_request/linux_patches_by_os

Upgrading Prom Firmware

Note: Systems running SLES10 and SGI ProPack 5 should use PROM 1.3 (or later) for systems, such as, the Altix 4700 with the next generation SHub ASIC, or PROM 4.53 (or later) which ship with SGI ProPack 5. The updated PROM contains key bug fixes, along with support for new kernel features, such as memory error recovery. Software support contract customers can download the most recent PROM versions from SGI Supportfolio: <https://support.sgi.com/>

To update or "flash" the system PROM, you first need to download the prom RPM from Supportfolio as described above.

Note: The PROM image on your hard drive is automatically updated when you upgrade your system to the latest SGI ProPack for Linux release. The installation of the `snprom` RPM (step 1) happens automatically as part of the SGI ProPack installation. Flashing it (steps 2-4) must be performed.

Once you have the latest PROM RPM, for instruction on how to flash the PROM are, as follows:

Note: Instructions for flashing the PROMs do not apply to SGI Altix XE systems.

1. If you have already installed SGI ProPack 5, you already have a version of the PROM firmware RPM in place. You can find and install the latest version of the PROM rpm by using the **online_update** mechanism, which will retrieve the latest prom RPM if a newer one is available.
2. If you do not have SGI ProPack 5 installed, or wish to upgrade the PROM prior to installation of SGI ProPack 5, retrieve the appropriate PROM RPM as described earlier in this section.
3. To install the RPM manually, use the following command:

```
rpm -Uvh snprom-XX.X-XXXX.ia64.rpm
```

4. Because there is no way to flash the PROM from Linux, you need to reboot your system and flash from EFI. As the system resets, you will see various messages from the system PROM. Eventually, you will be presented with a boot menu. It will look similar to the following:

```
EFI Boot Manager ver 1.10 [14.62]
```

```
Partition 1:
  CBricks      1      Nodes      1      0
  RBricks      0      CPUs       2      0
  IOBricks     1      Mem(GB)    2      0
```

```
Please select a boot option
```

```
SUSE Enterprise Linux Server 10
EFI Shell [Built-in]
Boot option maintenance menu
```

```
Arrow down to EFI Shell and hit enter.
```

5. Arrow down to EFI Shell [Built-in] and hit the ENTER key to get to the Shell> prompt.
6. You will now be at the Shell> prompt. If you just have one operating system installed and that you do not have a CD or DVD in your DVD drive, you can follow this example. If you have a disk in the DVD drive or multiple operating systems, you may need to change to a different EFI filesystem to find the appropriate PROM. This example assumes fs0. Change to fs0, as follows:

```
Shell> fs0:
```

7. If you have an SGI Altix 4000 series system, such as SGI Altix 4700, perform the following command (otherwise, skip this step):

```
cd efi\sgi
```

8. At this point, SGI Altix 4000 series customers are in the efi\sgi directory and Altix 3000 series customers are in \. You can verify the version of the PROM you are currently running, as follows:

```
fs0:\> flash -R
Running PROM version: X.XX
```

9. You can check the version of the PROM binary by running the appropriate command, as follows:

- For SGI Altix 4000 series systems:

```
fs0:\efi\sgi> flash -V ip65prom.bin
SGI PROM Flashing Utility
Version of prom image in file X.XX
```

- For all other Altix systems:

```
fs0:\> flash -V snprom.bin
SGI PROM Flashing Utility
Version of prom image in file X.XX
```

10. You are now ready to flash the PROM on all the nodes, as follows:

- For SGI Altix 4000 series systems:

```
fs0:\efi\sgi> flash -a ip65prom.bin
```

- For all other Altix systems:

```
fs0:\> flash -a snprom.bin
```

Caution: Always allow the flash process to complete before attempting any other system activities. Flashing the PROMs may take several minutes (depending on system size). Interrupting a flash process can result in disabled processors and nodes.

The all the nodes comment in step 10 includes both partitioned and unpartitioned systems. For partitioned systems, this means that it is important to upgrade all nodes (including all nodes in all partitions) before resetting any partitions. In other words, do not reset any partitions until all nodes in all partitions have been upgraded.

Identifying Bricks

Bricks are referenced by their racks and slot or bay locations. These values are stored in non-volatile memory on the L1. Virtually all system controller communication requires that each brick have a valid and unique rack and slot. If a brick does not have these, the output of an L1 or L2 `config` command will reflect that as shown in the following example:

```
L2> config
137.38.88.82.1.0 ---c-- (no rack/slot set)
L2>
```

If the brick currently has a rack and slot that is not duplicated in the configuration, you can simply target a command to that brick. To rename 001c1 to 004c11, perform the following:

```
L2> 1.1 brick rackslot 4 11
001c01:
brick rack set to 004 (take effect on next L1 reboot/power cycle)
brick slot set to 11 (take effect on next L1 reboot/power cycle)
L2> 1.1 reboot_l1
```

If the new brick's node and slot are a duplicate of an existing brick, power off the existing brick and rename the brick as described above. If there is no rack and slot currently set, the brick or IRU needs to be accessed via the L1 console port and issued the following command:

```
brick rackslot <new rack> <new slot>
```

To set the rack and slot from the L1 prompt, simply use the `brick rack` and `brick slot` commands. To set the rack and slot on one of the attached bricks (an attached I/O brick, C-brick, or a C-brick's I/O brick), use the L1 targeting commands `iia`, `iib`, or `nia`, `nib`. See the following example.

```
003c01-L1> config verbose
:0 - 003c01 (local)
:1 - ---i-- (IIA)
:5 - 002c01 (NIA)
:6 - 001x01 (IIA NIA)
003c01-L1> iia brick rack 4
---i--:
brick rack set to 004.
003c01-l1> iia reboot_l1
003c01 ERROR: no response from ---i--
003c01-L1> config
```

```
:0 - 003c01 (local)
:1 - 004i01 (IIA)
:5 - 002c01 (NIA)
:6 - 001x01 (IIA NIA)
003c01-L1>
```

Identifying Blades

SGI Altix 450 or Altix 4700 system blade types have three main categories:

- Compute/memory
- Input/output
- Reconfigurable Application-Specific Computing (RASC)

Table 2-1 shows the current blade types for Altix 450 and Altix 4700 systems.

Table 2-1 Valid Blade Types

Type	Description
C1	Single-processor with memory (SHub 2.x based)
C2	Dual-processor with memory (SHub 2.x based)
M2	Memory-only compute blade
IA	Double-height, BaseIO (1 per system/partition required)
I2E	Single-height PCIe or graphics-expansion
I3X	3-slot PCI-X expansion (supports hot-plugging)
I4E	4-slot double-height PCI-X, PCIe, or graphics-expansion
RC	FPGA compute blade

Each blade in individual rack unit (IRU) has a IRU blade slot ID as show in Figure 2-1.

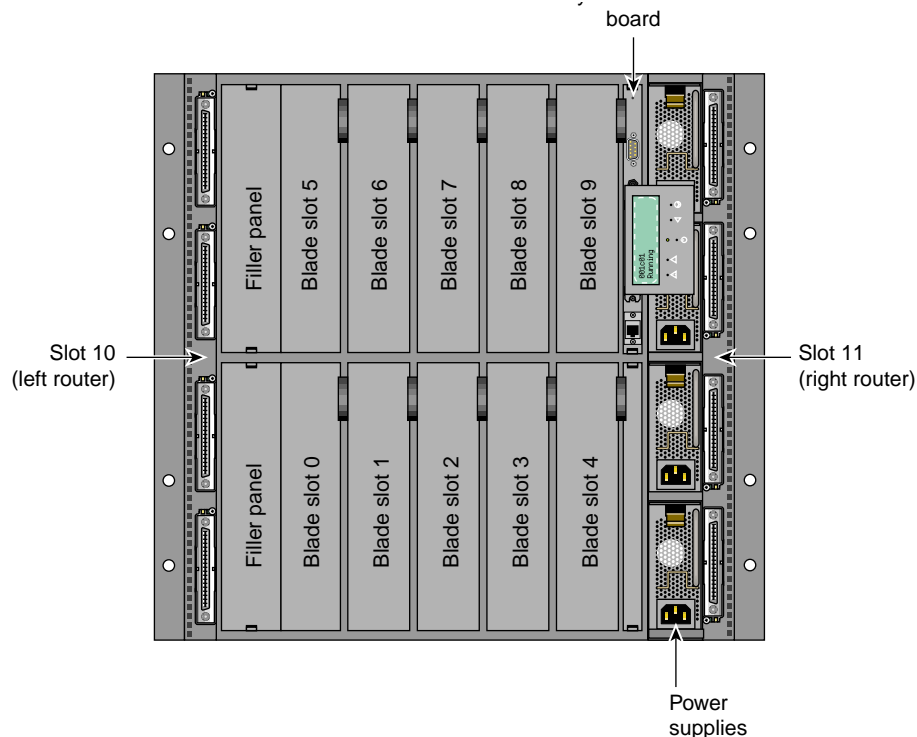


Figure 2-1 Individual Rack Unit (IRU) Blade Slot IDs

On an Altix 450 and Altix 4700 system, each IRU and Dense router must have a valid and unique rack and module ID. Module IDs should match the UPOS number on the 42U rack that aligns with the bottom of the IRU or Dense router (see Figure 2-2). You can assign a rack ID to the L1 controller, but not the L2, as follows:

$$\text{RackID} = 100 * \text{L1 rack number} + \text{L1 slot number}$$

The L2 generates its rack ID from the rack and slot of the local L1.

You can set the L1 slot number with the L1 `brick rs <rack> <slot>` command. The L1 controller must be rebooted for the new rack and slot number to take effect.

Blades are referenced by the unique rack and module ID of an IRU or racks and slot or bay locations. These values are stored in non-volatile memory on the L1. Virtually all system controller communication requires that each blade have a valid and unique rack and slot. If a brick does not have these, the output of an L2 `config` command will reflect that as shown in the following example:

```
L2> config  
137.38.88.82.1.0 ---c-- (no rack/slot set)  
L2>
```

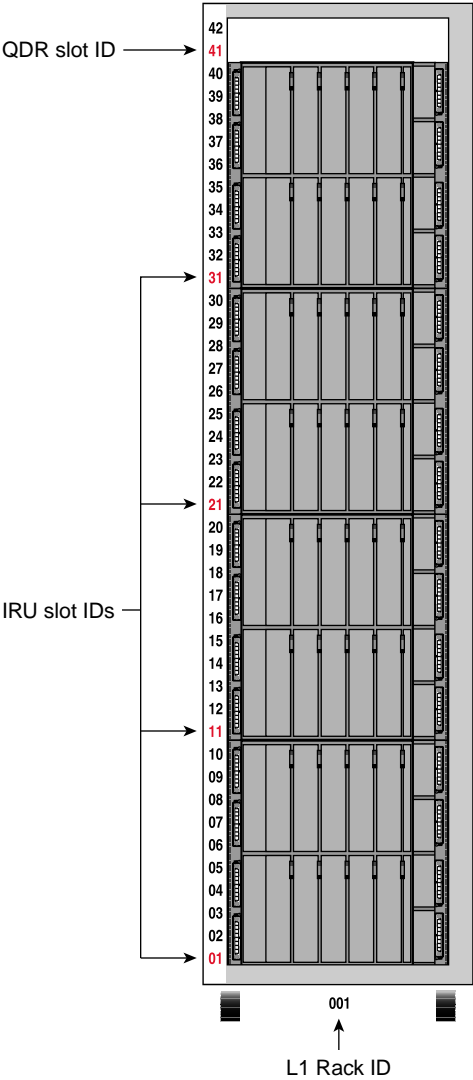


Figure 2-2 Rack and Module IDs on SGI Altix 4700 System

If you set the rack and slot ID of an L1 to 001.07, the L2 prompt displays the following rack ID:

```
hostname-107-L2>
```

You can use the L1 blade command to display information about blades on your Altix 450 or Altix 4700 system, as follows:

```
001c03-L1>blade
Slot# [name]      Enabled      SN          Blade Type      Current State
-----
0 [B0]           Enabled      NTT911      BaseIO          Power Off
1 [B1]           Enabled
2 [B2]           Enabled      NSH576      IP73_667        Power Off
3 [B3]           Enabled      NSH435      IP73_667        Power Off
4 [B4]           Enabled
5 [B5]           Enabled
6 [B6]           Enabled      NSH444      IP73_667        Power Off
7 [B7]           Enabled      NSH438      IP73_667        Power Off
8 [B8]           Enabled      NSH430      IP73_667        Power Off
9 [B9]           Enabled
10 [RTRL]        Enabled      NSV633      DUAL_ROUTER     Power Off
11 [RTRR]        Enabled      NSV627      DUAL_ROUTER     Power Off
001c03-L1>
```

For more information on the L1 blade command, see “blade” on page 54.

For more information on blades, IRUs and Dense routers, see the *SGI Altix 4700 System User's Guide*.

System Partitioning

A single SGI ProPack for Linux server 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.

You can use the **Connect to System Controller** task of the SGIconsole Console Manager GUI to connect to the L2 controller of the system you want to partition. The L2 controller must appear as a node in the SGIconsole configuration. For information on how to use SGIconsole, see the *Console Manager for SGIconsole Administrator's Guide*.

For detailed instructions on how to use the L2 controller commands to partition a system, see “System Partitioning” in the *Linux Configuration and Operations Guide*.

Hyper-Threading on Altix 450 or Altix 4700 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.

On dual-socket SGI Altix 450 or Altix 4700 systems, the basic CPU layout is as follows:

```
CPU A Socket 0 Primary Core
CPU B Socket 0 Secondary Core
CPU C Socket 1 Primary Core
CPU D Socket 1 Secondary Core
```

In this configuration, Hyper-Threading is not supported. Even if you disable secondary cores using the `cpu d` command, you cannot turn HT on because you have disabled that `cpu` (HT or secondary) via the L2 controller. Direct control of CPUs with the L2 `cpu` commands are available to alter this basic combination.

On single-socket SGI Altix 450 or Altix 4700 systems, the basic CPU layout is as follows:

```
CPU A Socket 0 Primary Core
CPU B Socket 0 HT
CPU C Socket 0 Secondary Core
CPU D Socket 0 HT
```

In this configuration you can enable or disable HT with the `ht e` or `ht d` commands, respectively, and affect the CPU B/D combination. Direct control of CPUs with L2 `cpu` commands are available to alter this basic combination. The system must be powered down when HT is enabled or disabled, as follows:

```
pwr d
r * s * cpu ht [e|d]
pwr u
```

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

Status and Error Messages

This section lists and describes the status and error messages generated by the L1 and L2 controllers. It also explains how to resolve the errors, if action is necessary.

L1 Controller Tasks and Messages

The L1 controller front panel display, located on the front panel of individual bricks, consists of a 2-line, 12-character liquid crystal display (LCD) that provides the following:

- Brick identification
- System status
- Warning of required service or failure
- Identification of failed components

Note: Besides the L1 control display, if you have an L2 controller, you can see the L1 controller messages on the L2 controller touch display located on the front door of the leftmost compute rack (position 001). If you have a system console, you can also see the L1 controller messages on your system console.

Table 2-2 lists the L1 controller messages.

Note: Note that in Table 2-2, a voltage warning occurs when a supplied level of voltage is below or above the nominal (normal) voltage by 10 percent. A voltage fault occurs when a supplied level is below or above the nominal by 20 percent.

Table 2-2 L1 Controller Messages

L1 System Controller Message	Message Meaning and Action Needed
Internal voltage messages:	
ATTN: x.xV high fault limit reached @ x.xxV	30-second power off sequence for the brick (or system, if no backup is available), server, or module.
ATTN: x.xV low fault limit reached @ x.xxV	30-second power off sequence for the brick (or system, if no backup is available), server, or module.
ATTN: x.xV high warning limit reached @ x.xxV	A higher than nominal voltage condition is detected.
ATTN: x.xV low warning limit reached @ x.xxV	A lower than nominal voltage condition is detected.
ATTN: x.xV level stabilized @ x.xV	A monitored voltage level has returned to within acceptable limits.
Fan messages:	
ATTN: FAN # x fault limit reached @ xx RPM	A fan has reached its maximum RPM level. The ambient temperature may be too high. Check to see if a fan has failed.
ATTN: FAN # x warning limit reached @ xx RPM	A fan has increased its RPM level. Check the ambient temperature. Check to see if the fan stabilizes.
ATTN: FAN # x stabilized @ xx RPM	An increased fan RPM level has returned to normal.
Temperature messages: low alt.	
ATTN: TEMP # advisory temperature reached @ xxC xxF	The ambient temperature at the brick's, server's, or module's air inlet has exceeded 30 °C.
ATTN: TEMP # critical temperature reached @ xxC xxF	The ambient temperature at the brick's, server's, or module's air inlet has exceeded 35 °C.
ATTN: TEMP # fault temperature reached @ xxC xxF	The ambient temperature at the brick's or server's air inlet has exceeded 40 °C.

Table 2-2 L1 Controller Messages **(continued)**

L1 System Controller Message	Message Meaning and Action Needed
Temperature messages: high alt.	
ATTN: TEMP # advisory temperature reached @ xxC xxF	The ambient temperature at the brick's, server's, or module's air inlet has exceeded 27 °C.
ATTN: TEMP # critical temperature reached @ xxC xxF	The ambient temperature at the brick's, server's, or module's air inlet has exceeded 31 °C.
ATTN: TEMP # fault temperature reached @ xxC xxF	The ambient temperature at the brick's, server's, or module's air inlet has exceeded 35 °C.
Temperature stable message:	
ATTN: TEMP # stabilized @ xxC/xxF	The ambient temperature at the brick's, server's, or module's air inlet has returned to an acceptable level.
Power off messages:	
Auto power down in xx seconds	The L1 controller has registered a fault and is shutting down. The message displays every 5 seconds until shutdown.
Brick or server appears to have been powered down	The L1 controller has registered a fault and has shut down.

L2 Controller Tasks

The L2 controller performs the following functions:

- Controls resource sharing.
- Controls L1 controllers.
- Resets the system.
- Issues non-maskable interrupts (NMIs).
- Displays voltage margin information.

- Routes data between upstream devices and downstream devices.

Upstream devices (for example, rack display, console, and modem) provide control for the system, initiate commands for the downstream devices, and act on the messages that they receive from downstream devices.

Downstream devices (for example, C-bricks, the USB hub of the R-brick, and L1 controllers of the bricks) perform the actions that are specified by the L2 controller commands, send responses to the L2 controller that indicate the status of the commands, and send error messages to the L2 controller.

- Allows remote maintenance.

You use the L2 controller touch display to do the following:

- Power the system on and off.
- Monitor voltage margins.
- Reset the system
- Enter a non-maskable interrupt (NMI).

The L2 controller also monitors and generates status and error messages related to the rack chassis items, such as the power bay and other rack items. The L2 controller also displays status and error messages generated by each individual brick's L1 controller. (See "L1 Controller Tasks and Messages" on page 46 for L1 controller message descriptions.)

The L2 controller information is displayed on the L2 controller touch display located in the front door of your server system. (The actual L2 controller is located on the top of your rack enclosure.)

Note: If you have a system console, you can also see the L2 controller messages on the system console.

L1 and L2 Controller Commands

You can use L1 and L2 controller commands to monitor and manage SGI servers and graphic or visualization systems. This chapter consists of the following sections:

- “Commands for Different Devices” on page 51
- “L1 Controller Commands” on page 52
- “L2 Controller Commands” on page 111

Note: Online command information is available. To view a list of L1 commands, enter `help` at the L1 prompt. To view a list of L2 commands, enter `help` at the L2 prompt. To view information about an individual command, enter `help` and the command name at the appropriate prompt.

Commands for Different Devices

This section explains which L1 and L2 controller commands can be used with which devices in the following sections:

- “L2 Commands” on page 52
- “L1 Commands” on page 52
- “L1 Controller Commands for Expert Users” on page 95

Note: The L1 and L2 controller commands do not apply to SGI Altix XE systems.

L2 Commands

You can use all of the L2 controller commands to monitor and manage all SGI servers that utilize the L2 controller. You can connect to the L2 host Ethernet connection on the system control board of an IRU or to a Dense router and the system controller spawns an L2 application providing L2 functionality.

Note: Although some of the specific examples of using L2 controller commands in this chapter may show components not applicable to your device, all L2 controller commands in this chapter are applicable to all servers and graphics systems supported.

L1 Commands

You can use most, but not all, of the L1 controller commands to monitor and manage all devices (SGI servers and Silicon graphics or visualization systems). Each controller command description in “L1 Controller Commands” on page 52, gives the devices each command supports.

Note: If you try to use an L1 controller command that is not available for a particular system or brick component, a message will inform you of this.

L1 Controller Commands

The following subsections describe the L1 controller commands and command sets, which are listed alphabetically. Examples of output are included where applicable.

For a list of all L1 controller commands, enter the following:

```
001c01-L1>help
Commands are:
check                fru                deadlock            shcfg
shubcfg|shub        systemace|ace      tiocfg             l2
daytona              abacus             athena             tio
hour                 nicfg              nlcfg              rmmr|mnr_rd
wmmr|mnr_wr         tdr                pktgen             tune
lbi                  psic               promise            promver|promversion
node                 pic                war                shdbg
mbrick               ssn                !|shell           !!|shellout
ssi                  pwm                isync              syscom
config|cfg          lldb               slit               psmi
error                blade              pci                 *                autopower|apwr
syscom|junkbus|jb|bedrockbrick
partdb              cpu                b2b                config|cfg
debug               display|dsp        button|btn          env
fan                 help|hlp           history|hist        lldb
link                log                ioport|ioprt       istat
l1                  leds               margin|mgn          network
pimm                port|prt           power|pwr           quit
reset|rst           nmi                softreset|softrst  select|sel
serial              sysstate           eeprom             uart
usb                 router|rtr         service            date
nvram               security           flash              reboot_l1
version|ver         fru|pci|node       test|tst           scan
enter 'hlp <cmd>' for more help on a single command.
001c01-L1>
```

For more information on a single command, enter the following:

```
T1-001c01-L1>hlp <command>
```

* (asterisk character)

Use * <command> to broadcast the command specified. For systems with a C-brick, a command that is broadcasted is sent to all of the bricks that are attached to the C-brick that issued the command. For systems that do not have a C-brick, a command that is broadcasted is sent to all systems connected to the system that issued the command. Example 3-1 shows sample output from the * version command.

Example 3-1 * version Command Output

```
001c07-L1>* version
001c07:
L1 0.7.27 (Image A), Built 04/28/2000 13:06:43 [P1 support]
001i21:
L1 0.8.xx (Image B), Built 06/13/2000 09:54:32 [P1 support]
```

autopower

The following `autopower` command set enables, disables, and aborts the feature that enables the system to automatically power up if power is lost. The `autopower` command also displays the current autopower setting (see Example 3-2).

- `autopower`
Shows current atopower setting.
- `autopower on`
Enables autopower feature.
- `autopower off`
Disables autopower feature.
- `autopower abort`
Aborts the feature that enables automatic power up if power is lost.

Example 3-2 autopower Command Output

```
001c20-L1>autopower
auto power on is disabled.
```

blade

The following `blade` command displays individual rack unit (IRU) slot configuration, enables or disables a blade.

- `blade`
Shows IRU slot configuration.
- `blade <num> e`
Enables a blade specified by `<num>`.

- `blade <num> d`
Disables a blade specified by *<num>*.

Example 3-3 shows the output from the blade command.

Example 3-3 blade Command Output

001c03-L1>**blade**

Slot#	[name]	Enabled	SN	Blade Type	Current State
0	[B0]	Enabled	NTT911	BaseIO	Power Off
1	[B1]	Enabled		<empty slot>	
2	[B2]	Enabled	NSH576	IP73_667	Power Off
3	[B3]	Enabled	NSH435	IP73_667	Power Off
4	[B4]	Enabled		<empty slot>	
5	[B5]	Enabled		<empty slot>	
6	[B6]	Enabled	NSH444	IP73_667	Power Off
7	[B7]	Enabled	NSH438	IP73_667	Power Off
8	[B8]	Enabled	NSH430	IP73_667	Power Off
9	[B9]	Enabled		<empty slot>	
10	[RTRL]	Enabled	NSV633	DUAL_ROUTER	Power Off
11	[RTRR]	Enabled	NSV627	DUAL_ROUTER	Power Off

001c03-L1>

brick

The following brick command set provides the status of the brick and sets the brick location and type.

- `brick`
Shows brick location and type. Example 3-4 shows sample output.
- `brick type <str>`
Shows brick type for system name *<str>*.
- `brick rack <rack number>`

Sets the rack number where the brick is located. Example 3-5 shows sample output.

- `brick slot <slot number>`

Sets the brick slot number in the rack. The variable `<slot number>` is a unit number from 01 to 39. Example 3-6 shows sample output.

- `brick rackslot <rack number> <slot number>`

Sets the brick rack and slot number. Example 3-7 shows sample output.

- `brick partition none`

Clears the brick partition number. Example 3-8 shows sample output.

- `brick partition <partition number>`

Sets the brick partition number. Example 3-9 shows sample output.

Note: For this command set, you can use `bay` or `upos` instead of `slot`, `rs` instead of `rackslot`, and `part` or `p` in the place of `partition`.

Example 3-4 `brick` Command Output

```
001c07-L1>brick
rack: 001 slot: 07 partition:0type: C source: EEPROM
```

Example 3-5 `brick rack <rack number>` Command Output

```
001c07-L1>brick rack 1
brick rack set to 001.
```


Example 3-6 brick slot *<slot number>* Command Output

```
001c07-L1>brick slot 07
brick slot set to 07.
```

Example 3-7 brick rackslot *<rack number>* *<slot number>* Command Output

```
001c07-L1>brick rackslot 01 07
brick rack set to 01
brick slot set to 07
```

Example 3-8 brick partition none Command Output

```
001c07-L1>brick partition none
brick partition cleared.
```

Example 3-9 brick partition *<partition>* Command Output

```
001c07-L1>brick partition 1
brick partition set to 1.
```

b2b

The b2b command is used to send an L1 command to an attached brick or enclosure. This command is used for debugging or maintenance situations.

```
b2b <port> <command>
```

Valid ports can be seen in the output of the `config` command, as follows:

```
L1> config
001c01:
:0 001c01 LOC
:4 001c02 U-A
:2 001c04 U-G
:34 002i02 IIB
L1>
```

Possible ports will vary by brick type.

For Altix 3000 systems with NUMALink 4 routers, valid port types are, as follows:

N0A	N1A	N2A	N3A
N0B	N1B	N2B	N3B
N0C	N1C	N2C	N3C
N0D	N1D	N2D	N3D
N0E	N1E	N2E	N3E
N0F	N1F	N2F	N3F
N0G	N1G	N2G	N3G
N0H	N1H	N2H	N3H

For Altix 3700 Bx2 systems, valid port types are, as follows:

U-F	L-F	IIB
U-G	L-G	IID
U-H	L-H	
U-A	L-A	

For Altix 350 systems, valid port types are, as follows:

NIA	IIA
NIB	IIB
NIC	
NID	

For Altix 4700 systems, valid port types are, as follows:

L0G	R3G
-----	-----

L1H R2H
L2G R1G
L3H R0H

For Altix 450 systems, valid port types are, as follows:

LG RG
LH RH

Example 3-10 shows sample output for the b2b command.

Example 3-10 b2b Command Output

```
001c01-L1>b2b 10g brick
001c11:
rack: 001, slot: 11, partition: none, type: 191010 Chassis [2MB flash],
serial:NTT949, source: EEPROM
001c01-L1>
```

config

Use the following `config` command set to view and reset L1 controller configuration information, and to check network interface (NI) ring configuration.

- `config`
Shows L1 controller configuration information. Example 3-12 shows sample output.
- `config verbose`
Shows an expanded view of the L1 controller configuration information.
- `config reset`
Resets the L1 controller configuration which initiates a new system controller discovery process.

Note: For this command set, you can use `cfg` instead of `config` and `v` instead of `verbose`.

Example 3-11 `config` Command Output

```
001c20-L1>config
0: - 001c20
```

Example 3-12 `config` Command Output

```
003c01-L1>config
:0 - 003c01
:1 - 004i01
:2 - 002c01
:3 - 001x01
003c01-L1>
```

Bricks are referenced by their racks and slot or bay locations. These values are stored in non-volatile memory on the L1. Virtually all system controller communications require that each brick has a valid and unique rack and slot.

The number that follows the L1 index, after the dash, is the brick identification number (for example, 003c01). The first three digits of the brick identification indicate the rack in which the brick resides. The fourth digit indicates the type of brick (Table 3-1). The last two digits indicate the slot position in which the brick resides.

Table 3-1 Valid Brick Types

Type	Description
c	C-brick
i	I-brick
p	P-brick
r	R-brick
x	X-brick
n	N-brick
g	G-brick
?	Unknown brick type

- Use `config reset` to reset the L1 controller configuration. This initiates a new system controller discovery process.
- Use `config ring` to check for network interface (NI) ring configuration.

For information on using the `config` command on an SGI Altix blade-based systems, see “Viewing System Configuration on Altix 450 or Altix 4700 System” on page 16.

cpu

Use the following `cpu` command set to enable, disable, and provide the status of the central processing units in a C-brick or server bricks performing the compute function.

- `cpu`
Shows status of all CPUs.
- `cpu e`
Enables all CPUs.
- `cpu d`
Disables all CPUs.
- `cpu <exp> e`
Enables the number of the CPU entered.
- `cpu <exp> d`
Disables the number of the CPU entered.

Use the `cpu` command to view the status of all CPUs. Example 3-13 shows a sample output from the `cpu` command on a blade-base Altix system.

CPU is in the format of `<number> <letter>` such as 0A, 0C, 3A, and so on. The number refers to the blade number on blade systems. On non-blade systems, number refers to the node number. (Note that a compute blade has four nodes). The letter refers to the CPU on the blade or node.

Example 3-13 `cpu` Command Output on a Blade-Based Altix System

```
001c01-L1>cpu
CPU      Present Enabled
-----  -
1A       1         1
1B       0         0
1C       0         0
1D       0         0
6A       1         1
```

```
6B    0    0
6C    0    0
6D    0    0
CPU Hyperthreading is ENABLED, but due to CPU configurations it has
been disabled on blade(s):    B1,B6
001c01-L1>
```

Example 3-14 shows a sample output from the `cpu` command on a non-blade Altix system.

Example 3-14 `cpu` Command Output on a Non-Blade Altix System

```
001c01-L1>cpu
CPU      Present Enabled
-----
0A      1        1
0C      1        1
1A      1        1
1C      1        1
2A      1        1
2C      1        1
3A      1        1
3C      1        1
001c01-L1>
```

date

Use the following `date` command set to view and set the current date and time used by the L1 controller.

- `date`
Shows the current date and time value used by the L1 controller. Example 3-15 shows sample output.
- `date <str>`
Sets the date and time value used by the L1 controller.
The variable `<str>` is a time value in the form `yyyymmddHHMMSS` (where `yyyy` is the four-digit year, `mm` is a two-digit month, `dd` is a two-digit day, `HH` is a two-digit hour, `MM` is a two-digit minute, and `SS` is a two-digit second).
- `date tz`
Displays the time zone offset used by the L1 controller.

- `date tz <str>`

Sets set the time zone offset used by the L1 controller.

The variable `<str>` is a maximum of +12 (for 12 hours ahead of GMT) and a minimum of -12 (for 12 hours behind GMT).

Note: The time does not automatically update for daylight savings time. You need to manually set the time using the `date <str>` command.

Example 3-15 `date` Command Output

```
001r19-L1>date
09/14/4655 22:47:07
```

debug

The following `debug` command set provides the status of and sets the virtual debug switches. (In systems that include C-bricks, these commands are valid only for C-bricks.)

- `debug`
Determines the current settings for the virtual debug switches.
- `debug <switches>`
Sets the virtual debug switches. The variable `<switches>` is a hexadecimal value for the switches.

Example 3-16 shows sample output of the `debug` command.

Example 3-16 `debug` Command Output

```
001c07-L1>debug
debug switches set to 0x0000
```

Use to Example 3-17 shows sample output of the `debug <switches>` command.

Example 3-17 `debug <switches>` Command Output

```
001c07-L1>debug 0x0001
debug switches set to 0x0001
```

Table 3-1 shows valid virtual debug switch settings.

Table 3-2 Virtual Debug Switch Settings

HEX Value	Description
Diagnostic Testing	
0	Normal Testing
1	No Testing
2	Heavy Testing
3	Manufacturing-level testing
Diagnostic Output Level	
4	Verbose - Information level set to verbose
Boot Stop Point	
0	Normal setting - Do not stop
8	Global POD
10	Local POD - Boot stop requested at local POD
18	Memoryless POD - Boot stop requested at no memory POD
Environment Variables	
20	Default Env - Ignore environment variables
80	Do Not Clear Errors
100	No Disable - Override disabled CPU or memory
200	Output prefixes (cpu & nasid - 0A 000 POD>)
400	Plain "vanilla" console - not EFI manager
800	Disable NUMALink discovery
1000	Dumps hardware error state at system boot time
2000	IO PROM ignores the autoboot environment variable
4000	Disable I/O discovery

display

The following `display` command set displays text on the front panel display, controls the LEDs on the front panel display, and shows the state of the LEDs.

- `display`
Shows the front panel display status. Example 3-18 shows sample output.
- `display <exp> <command>`
Use `display <exp> <command>` to write text on the L1 controller display (`<exp>` value is 1 or 2).
- `display power on`
Turns on the power LED.
- `display attention on`
Turns on the attention LED.
- `display failure on`
Turns on the failure LED.
- `display power off`
Turns off the power LED.
- `display attention off`
Turns off the attention LED.
- `display failure off`
Turns off the failure LED.

Note: For this command, you can use `dsp` instead of `display`, `pwr` instead of `power`, `attn` instead of `attention`, and `fail` for `failure`.

Example 3-18 `display` Command Output

```
001c07-L1>display
line 1: 001c10
line 2: powered up
```

env

The following `env` command set provides the status of environmental conditions and turns environmental monitoring on and off.

- `env`
Shows the status of the voltage, fan speed, and internal temperature of a brick. The output of the `env` command is divided into four areas: environmental monitoring and auto start status, voltage status, fan status, and temperature status. Example 3-19 shows sample output.
- `env check`
Shows a brief environmental status.
- `env on|off`
Enables or disables environmental monitoring (not recommended under normal operation).
- `env temp on|off`
Turns on or turns off temperature monitoring (not recommended under normal operation).
- `env altitude high|low`
Enables a high altitude (greater than 5000 feet/1524 meters) or low-altitude (less than 5000 feet/1524 meters) temperature limit setting for the environmental monitoring. Example 3-20 shows sample output.
- `env reset`
Resets all current warnings and faults that are set. Example 3-21 shows sample output.

Note: For this set of commands, you can use `rst` in place of `reset`.

On the blade enclosures for RASC, Altix 450 systems, and Altix 4700 systems, the `env` output appears similar to Example 3-19. The main change from previous Altix systems is in the area of the temperature control. An `Autofan` control appears under the `Advisory` and `Critical` temperature columns for some entries.

For enclosures on blade-based systems, there is one or more zones (sets) of fans (at least one (1) fan in each zone) and one or more zones of temperature sensors (at least one (1) temperature sensor in each zone). Each fan zone may be controlled by one or more of the temperature sensor zones, but only by one at any given time.

Example 3-19 shows output below from an Altix 4700 system; the last set of data shows the relationships between these fan zones and temperature sensor zones.

On the Altix 4700 system enclosure, the first line shows the fan zone. There is only one fan zone on the Altix 4700 enclosure and it consists of fans 0 through 5, as the first line indicates. The following indented lines show the temperature sensor zones that can control that fan zone. At any given time, only one temperature zone can be in control, and it is indicated by the `Controlling zone` text. The controller temperature zone is the one closest to the limit values. Basically, the hottest temperature zone is in control of the fan zone.

The environmental monitoring and auto start status area has two statements. The first statement describes the current state of environmental monitoring. The environmental monitoring state is one of the following:

- “disabled - initialization error”
- “disabled - configuration error”
- “disabled - start error”
- “disabled”
- “disabled - unknown error (0x00)”
- “running”

The second statement describes the current state of the auto start. The auto start state is “enabled” or “disabled.”

The voltage status contains five columns as follows:

- The Description column provides the name of the supply voltage (2.5 V, 3.3 V, 5 V, etc.).
- The State column provides the current state of the supply voltage (Enabled, Fault, Warning, Waiting, or Disabled).
- The Warning Limits column provides the range of voltages that, when exceeded, causes a supply voltage to be in a Warning state.
- The Fault Limits column provides the range of voltages that, when exceeded, causes a supply voltage to be in a Fault state.
- The Current column provides the current value of the supply voltage.

The fan status contains four columns, as follows:

- The Description column provides the name of each fan.
- The State column provides the current state of the fan (Warning, Enabled, Waiting, or Disabled).
- The Warning RPM column provides the lowest revolutions-per-minute allowed before a fan enters a Warning state.
- The Current RPM column provides the current value of the revolutions-per-minute for the fan.

The temperature status contains six columns as follows:

- The Description column provides the name of a temperature sensor.
- The State column provides the current state of the temperature sensor (Fault, Warning, Enabled, or Disabled).
- The Advisory Temp column provides the temperature that, when exceeded, causes a temperature advisory state.
- The Critical Temp column provides the temperature that, when exceeded, causes a critical temperature state.
- The Fault Temp column provides the temperature that, when exceeded, causes the temperature sensor to be in a Fault state.
- The Current Temp column provides the current temperature reading from the temperature sensor.

Example 3-19 env Command Output

001c01-L1>env

Environmental monitoring is enabled (60 devices).

Description	State	Warning Limits		Fault Limits		Current
B1 12V	Enabled	10%	10.80/ 13.20	20%	9.60/ 14.40	11.856
B1 1.85V AUX	Enabled	10%	1.67/ 2.04	20%	1.48/ 2.22	1.849
B1 1.5V AUX	Enabled	10%	1.35/ 1.65	20%	1.20/ 1.80	1.494
B1 1.85V	Enabled	10%	1.67/ 2.04	20%	1.48/ 2.22	1.803
B1 1.2V	Enabled	10%	1.08/ 1.32	20%	0.96/ 1.44	1.125
B6 12V	Enabled	10%	10.80/ 13.20	20%	9.60/ 14.40	11.856
B6 1.85V AUX	Enabled	10%	1.67/ 2.04	20%	1.48/ 2.22	1.868
B6 1.5V AUX	Enabled	10%	1.35/ 1.65	20%	1.20/ 1.80	1.503
B6 1.85V	Enabled	10%	1.67/ 2.04	20%	1.48/ 2.22	1.797
B6 1.2V	Enabled	10%	1.08/ 1.32	20%	0.96/ 1.44	1.140
B7 12V	Enabled	10%	10.80/ 13.20	20%	9.60/ 14.40	11.887
B7 1.2V	Enabled	10%	1.08/ 1.32	20%	0.96/ 1.44	1.197
B7 2.5V	Enabled	10%	2.25/ 2.75	20%	2.00/ 3.00	2.485
B7 3.3V	Enabled	10%	2.97/ 3.63	20%	2.64/ 3.96	3.314
B7 1.2V	Enabled	10%	1.08/ 1.32	20%	0.96/ 1.44	1.202
B7 1.85V	Enabled	10%	1.67/ 2.04	20%	1.48/ 2.22	1.872
RTRL 1.85V AUX	Enabled	10%	1.67/ 2.04	20%	1.48/ 2.22	1.865
RTRL 12V	Enabled	10%	10.80/ 13.20	20%	9.60/ 14.40	11.902
RTRL 1.85V	Enabled	10%	1.67/ 2.04	20%	1.48/ 2.22	1.834
RTRR 1.85V AUX	Enabled	10%	1.67/ 2.04	20%	1.48/ 2.22	1.859
RTRR 12V	Enabled	10%	10.80/ 13.20	20%	9.60/ 14.40	11.887
RTRR 1.85V	Enabled	10%	1.67/ 2.04	20%	1.48/ 2.22	1.837

Description	State	Warning RPM	Current RPM
FAN 0 FAN 0	Enabled	1620	1928
FAN 1 FAN 1	Enabled	1620	1881
FAN 2 FAN 2	Enabled	1620	1934
FAN 3 FAN 3	Enabled	1620	1850
FAN 4 FAN 4	Enabled	1620	1938
FAN 5 FAN 5	Enabled	1620	1906

Description	State	Advisory Temp	Critical Temp	Fault Temp	Current Temp
B1 0 BLADE 0	Enabled	[Autofan Control]		70C/158F	30C/ 86F
B1 1 BLADE 1	Enabled	[Autofan Control]		70C/158F	26C/ 78F
B1 2 SHUB	<not present>				

```

B1 3 CPU 0          <not present>
B1 4 CPU 1          <not present>
B1 5 CPU 0 POD     Enabled      [Autofan Control]   75C/167F   44C/111F
B1 6 CPU 1 POD     <not present>
B6 0 BLADE 0       Enabled      [Autofan Control]   70C/158F   29C/ 84F
B6 1 BLADE 1       Enabled      [Autofan Control]   70C/158F   24C/ 75F
B6 2 SHUB          Enabled      [Autofan Control]   92C/197F   52C/125F
B6 3 CPU 0         <not present>
B6 4 CPU 1         <not present>
B6 5 CPU 0 POD     Enabled      [Autofan Control]   75C/167F   45C/113F
B6 6 CPU 1 POD     <not present>
B7 0 BLADE 0       Enabled      [Autofan Control]   70C/158F   27C/ 80F
B7 1 BLADE 1       Enabled      [Autofan Control]   70C/158F   26C/ 78F
B7 2 N0 FPGA       Enabled      [Autofan Control]   85C/185F   32C/ 89F
B7 3 N1 FPGA       Enabled      [Autofan Control]   85C/185F   34C/ 93F
B7 4 Inlet         Enabled      [Autofan Control]   70C/158F   22C/ 71F
B7 5 Exhaust       Enabled      [Autofan Control]   70C/158F   28C/ 82F
RTRL 0 Monitor     Enabled      [Autofan Control]   70C/158F   27C/ 80F
RTRR 0 Monitor     Enabled      [Autofan Control]   70C/158F   27C/ 80F
0 FAN 0            Enabled      [Autofan Control]   60C/140F   24C/ 75F
1 FAN 1            Enabled      [Autofan Control]   60C/140F   25C/ 77F

```

Zone Name	State	Temp Sensors	Target	Current	Fan Index	Curr/Min Fan %
191010	Enabled		40C/104F	26C/ 78F	0,1,2,3,4,5	19%/ 19%
>>B1 BOARD		1	40C/104F	26C/ 78F	= Controlling Zone <<	
B1 SHUB		2	<disabled>			
B1 CPU POD		5,6	64C/147F	44C/111F	=	
B1 CPU		3,4	<disabled>			
B6 BOARD		1	40C/104F	24C/ 75F	=	
B6 SHUB		2	80C/176F	52C/125F	=	
B6 CPU POD		5,6	64C/147F	45C/113F	=	
B6 CPU		3,4	<disabled>			
B7 BOARD		4	45C/113F	22C/ 71F	=	
B7 N0 FPGA		2	65C/149F	32C/ 89F	=	
B7 N1 FPGA		3	65C/149F	34C/ 93F	=	
RTRL RTR		0	40C/104F	27C/ 80F	=	
RTRR RTR		0	40C/104F	27C/ 80F	=	
FAN CTRL		0,1	45C/113F	25C/ 77F	=	

```
001c01-L1>
```

Example 3-20 env altitude high Command Output

```
001c07-L1>env altitude high
001c07-L1>
```

Example 3-21 env reset Command Output

```
001c07-L1>env reset
001c07-L1>
```

fan

Use the `fan` command to determine whether the fans are on or off and to read the fan speeds. In the `fan` command output, the number in parentheses is the counter reading for the fan. The counter reading is a value provided by the fan tachometer. The system controller converts the counter reading into a revolutions-per-minute value.

Example 3-22 shows sample output from the `fan` command.

Example 3-22 fan Command Output

```
001c07-L1>fan
fan(s) are on.
fan 0 rpm 2465 (339)
fan 1 rpm 2423 (352)
fan 2 rpm 2430 (349)
```

When the temperature of the brick is below 30 °C, the fans run at 2400 rpm. If a fan fails and the speed of the fan drops below 2100 rpm, the system controller increases the fan speed for the fans to 4400 rpm. If any two fans drop below 2100 rpm, the system controller shuts down the brick.

When the temperature of the brick is between 30 °C and 40 °C, the fans run at 3400 rpm. If a fan fails and the speed of the fan drops below 3100 rpm, the system controller shuts down the brick.

flash

The following `flash` command set provides the status of and updates of the firmware images stored in flash memory.

- `flash status`

Shows the status of the two firmware images stored in flash memory. Example 3-23 shows sample output. Each image has a checksum value that indicates whether an image is valid.

- `flash status verbose`
Displays an expanded version of the flash status.
- `flash default <a|b>`
Sets firmware image A or firmware image B as the default image that the system controller uses when booting.
- `flash default current`
Sets the current image as the default flash image.
- `flash default new`
Sets the new image as the default flash image.
- `flash default old`
Sets the old image as the default flash image.
- `flash default reset`
Sets the firmware image with the latest time-stamp as the default image that the system controller uses when booting. If the selected firmware image is not valid, the flash default commands will return the following message: “cannot set default--image A (or B) is invalid!”.

Example 3-23 `flash status` Command Output

```
001c07-L1>flash status
Flash image A currently booted
```

Image	Status	Revision	Built
-----	-----	-----	-----
A	default	0.7.27	04/28/2000 13:06:43
B	valid	0.8.0	05/24/2000 10:50:23

help

The following help command set provides helpful information on the system controller commands.

- `help`

Generates a list of all of the system controller commands.

- `help <command>`

Displays more information on a single command. The variable `<command>` is the name of a command.

Example 3-24 shows output of the `help` command.

Example 3-24 `help` Command Output

```
001c01-L1>help
Commands are:
check          fru          deadlock     shcfg
shubcfg|shub  systemace|ace tiocfg       l2
daytona       abacus       athena       tio
hour          nicfg        nlcfg        rmmr|mmr_rd
wmmr|mmr_wr   tdr          pktgen       tune
lbi           psic         promice      promver|promversion
node          pic          war          shdbg
mbrick        ssn          !|shell     !!|shellout
ssi           pwm          isync        syscom
config|cfg    lldb        slit         psmi
error         blade       pci          *          autopower|apwr
syscom|junkbus|jb|bedrockbrick
partdb        cpu          b2b          config|cfg
debug         display|dsp  button|btn   env
fan           help|hlp    history|hist lldb
link          log         ioport|ioprt istat
l1            leds        margin|mgn   network
pimm          port|prt    power|pwr    quit
reset|rst     nmi         softreset|softrst select|sel
serial        sysstate    eeprom       uart
usb           router|rtr  service      date
nvram         security    flash        reboot_l1
version|ver   pbay        test|tst     scan
fru|pci|node
enter 'hlp <cmd>' for more help on a single command.
001c01-L1>
```

As stated in the `help` command output in Example 3-24, you can use `hlp <cmd>` for more information on a particular command.

history

Use the `history` command to display a history of the L1 command processor.

In the `history` command output, the first number in the history length is the number of commands stored in the history array. The second number in the history length is the maximum number of commands that can be stored in the history array. Example 3-25 shows sample output.

Note: For this set of commands, you can use `hist` in place of `history`.

Example 3-25 `history` Command Output

```
001c07-L1>history
History length: 3/20
2: fan speed 4
1: fan

0: env
```

l1

Use the `l1` command to engage the L1 controller command processor. Press `Ctrl+D` to disengage the command processor and return to console mode. Example 3-26 shows sample output.

Example 3-26 l1 Command Output

```
001c003-L1>l1
L1 command processor engaged, <CTRL-D> to exit.
```

leds

The `leds` command shows the current CPU state and is used on compute blades or bricks. The output of the `leds` command is used to diagnose system problems. Example 3-27 shows the `leds` command output on a blade-based Altix system.

Example 3-27 leds Command Output on a Blade-Based Altix System

```
001c01-L1>leds
B1 CPU  A: 0x00:   Kernel: CPU idle
           0x01:   Kernel: CPU idle
B1 CPU  B: < CPU not present >
B1 CPU  C: < CPU not present >
B1 CPU  D: < CPU not present >

B6 CPU  A: 0x00:   Kernel: CPU idle
           0x01:   Kernel: CPU idle
B6 CPU  B: < CPU not present >
B6 CPU  C: < CPU not present >
B6 CPU  D: < CPU not present >

001c01-L1>
```

Example 3-28 shows the `leds` command output on a non-blade Altix system.

Example 3-28 leds Command Output on a Non-Blade Altix System

```
001c01-L1>leds
CPU 0A: 0x01: Kernel: CPU idle
        0x00: Kernel: CPU idle
CPU 0C: 0x00: Kernel: CPU idle
        0x01: Kernel: CPU idle

CPU 1A: 0x00: Kernel: CPU idle
        0x01: Kernel: CPU idle
CPU 1C: 0x00: Kernel: CPU idle
        0x01: Kernel: CPU idle

CPU 2A: 0x00: Kernel: CPU idle
        0x01: Kernel: CPU idle
CPU 2C: 0x00: Kernel: CPU idle
        0x01: Kernel: CPU idle

CPU 3A: 0x01: Kernel: CPU idle
        0x00: Kernel: CPU idle
CPU 3C: 0x01: Kernel: CPU idle
        0x00: Kernel: CPU idle
001c01-L1>
```

link

Use the `link` command to show the error status for the network interface (NI) and I/O interface (II) connector links.

Use the `link verbose` command to show the complete status for the NI and II connector links.

Note: For this command, you can use `v` in the place of `verbose`.

log

The following `log` command set displays the contents of the log, clears or resets the log, and writes an entry into the log.

- `log`
Shows the contents of the log. Example 3-29 shows sample output. If the log is empty, the output from the `log` command is “log is empty.”
- `log clear|reset`
Use `log clear` or `log reset` to empty the log. Example 3-30 shows sample output.
- `log insert <entry>`
Use `log insert <entry>` to write a line in the log. The variable `<entry>` is text to enter into the log. Example 3-31 shows sample output from the `log Start the Test` command.

Example 3-29 `log` Command Output

```
001c07-L1>log
USB: registered as remote
USB-R: USB:device was reset
USB: unregistered
USB-R: IRouter:read failed - read error
USB: registered as remote
USB-R: USB:device was reset
USB: unregistered
USB-R: IRouter:read failed - read error
USB: registered as remote
SMP-R: UART:UART_NO_CONNECTION
L1 booting...
[L1-0] ALERT: eeprom.c line 367 ; eeprom 0 checksum error.
USB: registered as remote
L1 booting...
[L1-0] ALERT: eeprom.c line 367 ; eeprom 0 checksum error.
USB: registered as remote
CTI-R: UART:UART_BREAK_RECEIVED
CTI-R: IRouter:read failed - read error
USB-R: USB:device was reset
```

Example 3-30 `log reset` Command Output

```
001c003-L1>log reset
log reset.
```

Example 3-31 log insert Start the Test Command Output

```
001c003-L1>log insert Start the Test
log entry made.
```

network

The following `network` command set displays and sets the mode of the network communication interface. (For systems that include a C-brick, this command is valid only for C-bricks and only supported on Altix 3000 series systems.)

- `network`
Shows the current mode of the network communication interface. Example 3-32 shows sample output.
- `network usb`
Use `network usb` to set the network communication interface mode to Universal Serial Bus (USB) and disable autodetection.
- `network 422`
Use `network 422` to set the network communication interface mode to RS-422 protocol and disable autodetection. Example 3-33 shows sample output from the `network usb` command.
- `network autodetect|auto on`
Use `network autodetect|auto on` to turn on the autodetection.
- `network autodetect|auto off`
Use `network autodetect|auto off` to turn off the autodetection.

Example 3-32 network Command Output

```
001c07-L1>network
network interface communication is USB
```

Example 3-33 network usb Command Output

```
001c003-L1>network usb
nvram parameter changed, reset required to affect change.
```

nmi

Use the `nmi` command to issue a non-maskable interrupt (NMI) on server components that perform the compute function.

Example 3-34 `nmi` Command Output

```
001c07-L1>nmi
001c07-L1>
```

partdb

The following `partdb` command set displays, enables, and disables partitioning.

- `partdb`
Displays partition information from the database.
- `partdb enable`
Enables the partition for a system.
- `partdb disable`
Disables the partition for a system.

Note: For this command set, you can use `e` or `on` instead of `enable` and `d` or `off` instead of `disable`.

For more information on partitioning, see “System Partitioning” in the *Linux Configuration and Operations Guide*.

pbay

The following `pbay` command set displays information about the power bay, distributed power supplies (DPSs), DC output ports, and field replaceable units (FRUs) on Altix 3000 series systems:

- `pbay`
Shows the status of the power bay (see Example 3-35).

- `pbay version`
Shows the firmware version of the power bay (see Example 3-36).
- `pbay dps`
Shows the status of all distributed power supplies (DPS).
- `pbay dps <dps number 1 - 6>`
Shows the status of an individual distributed power supply. The distributed power supplies are identified by a number ranging from 1 to 6.
- `pbay dcport`
Shows the status of all DC output ports.
- `pbay dcport <DC port number 1 - 8>`
Shows the status for an individual DC port. The ports are identified by a number ranging from 1 to 8.
- `pbay fru`
Shows information about all the power bays and DPS FRUs.
- `pbay fru <0 for power bay, 1 - 6 for DPS>`
Shows information about the individual power bay FRUs, use the number 0. To view information about a distributed power supply FRU, use the numbers 1 through 6.
- `pbay reset`
Resets the power bay and the power supplies.
- `pbay env`
Shows the status of environmental monitoring.
- `pbay env on|off`
Enables or disables environmental monitoring.
- `pbay init`
Initializes communication with the power bay.
- `pbay serial`
Displays the power bay ICMB card serial number.

Example 3-35 `pbay` Command Output

```
001r28-L1>pbay
Total current: 16.5 Amps (+/-11%)
```

Example 3-36 `pbay version` Command Output

```
001r28-L1>pbay version
PBay FW Version 00.18, Built 09/29/2001 14:00:00
```

pci

The following `pci` command set displays the status of the PCI cards in an I/O brick, or a PCI expansion module, and powers up, powers down, and resets a PCI card. It is available on Altix 3000 series systems and on Altix 450 and Altix 4700 systems. Output on blade-based systems is slightly different than prior brick-based systems. The output has a column that specifies the blade slot.

- `pci`
Shows the value of the status register for each PCI card. The output of the `pci` command contains eight columns, as follows:
 - The Bus column lists the number of the bus for each PCI card.
 - The Slot column lists the slot value for each PCI card.
 - The Stat column lists the hexadecimal value of the status register for each PCI card.
 - The Card Type column lists the card type (7.5 W, 15 W, 25 W, or none) for each slot.
 - The Power column lists the value (error & off, error & on, okay & off, or okay & on) of the power OK and power on bits.
 - The Attn LED column lists the value (off or on) of the attention LED for the PCI card.
 - The Enable column lists the value (off or on) of the bus enable bit for the PCI card.
 - The Reset column lists the value (off or on) of the reset bit for the PCI card.

Example 3-37 shows sample output from the `pci` command.

- `pci <u|d>`

Use `pci <u|d>` to power up (u) or power down (d) all of the PCI cards in an I/O brick or PCI expansion module. Example 3-38 shows sample output.

- `pci <bus> u|d`

Use `pci <bus> <u|d>` to power up (u) or power down (d) all of the PCI cards on a bus in an I/O brick or PCI expansion module. The variable `<bus>` is the bus number. Example 3-39 shows sample output from the `pci 1 u` command.

- `pci <bus> <slot> u|d`

Use `pci <bus> <slot> <u|d>` to power up (u) or power down (d) an individual PCI card. The variable `<bus>` is the bus number and the variable `<slot>` is the slot number. Example 3-40 shows sample output from the `pci 1 2 u` command.

- `pci reset`

Use `pci reset` to reset all of the PCI cards in an I/O brick, Origin 300 server, or PCI expansion module. Example 3-41 shows sample output.

- `pci <bus> reset`

Use `pci <bus> reset` to reset all of the PCI cards on a bus in an I/O brick, Origin 300 server, or PCI expansion module. The variable `<bus>` is the bus number. Example 3-42 shows sample output from the `pci 1 reset` command.

- `pci <bus> <slot> reset`

Use `pci <bus> <slot> reset` to reset an individual PCI card. The variable `<bus>` is the bus number and the variable `<slot>` is the slot number. Example 3-43 shows sample output from the `pci 1 2 reset` command.

Note: For this command set, the mnemonic `rst` may replace the word `reset`.

Example 3-37 pci Command Output

```
001i21-L1>pci
Bus Slot Stat Card Type Power          Attn LED Enable Reset
-----
  1   1 0x91      15W  okay & on      off   on   off
  1   2 0x57      none okay & off    off   off  on
  1   3 0x91      15W  okay & on      off   on   off
  1   4 0xff  on board  N/A & N/A    N/A   N/A  off
  2   1 0x57      none okay & off    off   off  on
  2   2 0x57      none okay & off    off   off  on
```

Example 3-38 pci d Command Output

```
001i21-L1>pci d
001i21-L1>
```

Example 3-39 pci 1 u Command Output

```
001i21-L1>pci 1 u
001i21-L1>
```

Example 3-40 pci 1 2 u Command Output

```
001i21-L1>pci 1 2 u
001i21-L1>
```

Example 3-41 pci reset Command Output

```
001i21-L1>pci reset
001i21-L1>
```

Example 3-42 pci 1 reset Command Output

```
001i21-L1>pci 1 reset
001i21-L1>
```

Example 3-43 pci 1 2 reset Command Output

```
001i21-L1>pci 1 2 reset
001i21-L1>
```

port

Use the `port` command to view the value of the status register for each port on a brick. As shown in Example 3-44 and Example 3-45, the output of the `port` command contains six columns, as follows:

- The Port column lists the name of each port.
- The Stat column lists the hexadecimal value of the status register for each port.
- The Remote Pwr column lists the value, “okay” or “none,” of the remote power OK bit (bit 0) for each port.
- The Local Pwr column lists the value, “okay” or “none,” of the local power OK bit (bit 1) for each port.

- The Link LED column lists the value, “on” or “off,” of the link LED bit (bit 2) for each port.
- The SW LED column lists the value, “on” or “off,” of the software LED bit (bit 3) for each port.

Note: For this command, the mnemonic `prt` may replace the word `port`.

Example 3-44 `port` Command Output for an R-brick

```
001r19-L1>port
Port Stat Remote Pwr Local Pwr Link LED SW LED
-----
A 0x02 none okay off off
B 0x0f okay okay on on
C 0x0f okay okay on on
D 0x02 none okay off off
E 0x02 none okay off off
F 0x0b okay okay off on
G 0x0b okay okay off on
H 0x02 none okay off off
```

Example 3-45 `port` Command Output for a C-brick

```
001c07-L1>port
Port Stat Remote Pwr Local Pwr Link LED SW LED
-----
A 0x0f okay okay on on
B 0x0f okay okay on on
```

Example 3-46 shows the output of the `port v` command on a blade-based Altix system.

Example 3-46 `port v` Command Output for Blade-Based Altix System

```
001r19-L1>port v
Port Name Status Remote Pwr Local Pwr Link LED SW LED WP State
-----
B1 NI0 0x24 N/A N/A on N/A on POWER UP
B1 NI1 0x24 N/A N/A on N/A on POWER UP
B6 NI0 0x24 N/A N/A on N/A on POWER UP
B6 NI1 0x24 N/A N/A on N/A on POWER UP
B7 NI0 0x2c N/A N/A on N/A on POWER UP
B7 NI1 0x2c N/A N/A on N/A on POWER UP
B7 TIO 0 INT 0x24 N/A N/A on N/A on POWER UP
```

```

B7 TIO 1 INT 0x24      N/A      N/A      on      N/A on      POWER UP
RTRL RTRA A 0x24      N/A      N/A      on      N/A on      POWER UP
RTRL RTRA B <disabled>
RTRL RTRA C <disabled>
RTRL RTRA D <disabled>
RTRL RTRA E 0x24      N/A      N/A      on      N/A on      POWER UP
RTRL RTRA F <disabled>
RTRL RTRA G 0x2f      okay     okay     on      on on      LINK READY
RTRL RTRA H 0x2f      okay     okay     on      on on      LINK READY
RTRL RTRB A 0x24      N/A      N/A      on      N/A on      POWER UP
RTRL RTRB B <disabled>
RTRL RTRB C <disabled>
RTRL RTRB D 0x24      N/A      N/A      on      N/A on      POWER UP
RTRL RTRB E 0x24      N/A      N/A      on      N/A on      POWER UP
RTRL RTRB F <disabled>
RTRL RTRB G 0x2f      okay     okay     on      on on      LINK READY
RTRL RTRB H 0x2f      okay     okay     on      on on      LINK READY
RTRR RTRA A 0x24      N/A      N/A      on      N/A on      POWER UP
RTRR RTRA B <disabled>
RTRR RTRA C 0x24      N/A      N/A      on      N/A on      POWER UP
RTRR RTRA D 0x24      N/A      N/A      on      N/A on      POWER UP
RTRR RTRA E <disabled>
RTRR RTRA F <disabled>
RTRR RTRA G 0x2f      okay     okay     on      on on      LINK READY
RTRR RTRA H 0x2f      okay     okay     on      on on      LINK READY
RTRR RTRB A 0x24      N/A      N/A      on      N/A on      POWER UP
RTRR RTRB B <disabled>
RTRR RTRB C 0x24      N/A      N/A      on      N/A on      POWER UP
RTRR RTRB D <disabled>
RTRR RTRB E <disabled>
RTRR RTRB F <disabled>
RTRR RTRB G 0x2f      okay     okay     on      on on      LINK READY
RTRR RTRB H 0x2f      okay     okay     on      on on      LINK READY

```

001c01-L1>

Example 3-47 shows the output of the `port -v` command on a non-blade Altix system.

Example 3-47 `port -v` Command Output for non-Blade Altix Systems

```

001r19-L1>port -v
Port Name      Status Remote Pwr Local Pwr  Link LED SW LED WP  State
-----
T RTR F 0x02    none      okay     off      off off  POWER UP
T RTR G 0x2f    okay     okay     on       on on   LINK READY
T RTR H 0x02    none      okay     off      off off  POWER UP

```

```

T RTR A 0x2f      okay      okay      on      on  on  LINK READY
B RTR F 0x02      none      okay      off     off off POWER UP
B RTR G 0x2f      okay      okay      on      on  on  LINK READY
B RTR H 0x02      none      okay      off     off off POWER UP
B RTR A 0x2f      okay      okay      on      on  on  LINK READY
  N1 XIO 0x2f      okay      okay      on      on  on  LINK READY
  N3 XIO 0x02      none      okay      off     off off POWER UP
T RTR N0 NI0 0x2f  okay      okay      on      on  on  LINK READY
  N0 NI0 0x2f      okay      okay      on      on  on  LINK READY
B RTR N0 NI1 0x2f  okay      okay      on      on  on  LINK READY
  N0 NI1 0x2f      okay      okay      on      on  on  LINK READY
T RTR N1 NI0 0x2f  okay      okay      on      on  on  LINK READY
  N1 NI0 0x2f      okay      okay      on      on  on  LINK READY
B RTR N1 NI1 0x2f  okay      okay      on      on  on  LINK READY
  N1 NI1 0x2f      okay      okay      on      on  on  LINK READY
B RTR N2 NI0 0x2f  okay      okay      on      on  on  LINK READY
  N2 NI0 0x2f      okay      okay      on      on  on  LINK READY
T RTR N2 NI1 0x2f  okay      okay      on      on  on  LINK READY
  N2 NI1 0x2f      okay      okay      on      on  on  LINK READY
B RTR N3 NI0 0x2f  okay      okay      on      on  on  LINK READY
  N3 NI0 0x2f      okay      okay      on      on  on  LINK READY
T RTR N3 NI1 0x2f  okay      okay      on      on  on  LINK READY
  N3 NI1 0x2f      okay      okay      on      on  on  LINK READY

```

L1>

power

The following `power` command set displays the status of the supplies, and powers up and powers down the supplies.

- `power`
Shows the detailed current state of the power and margin values for the power supplies in a blade, a brick, or system. Example 3-48 shows sample power command output on a blade-based Altix system. Example 3-49 shows sample power command output on a non-blade Altix system.
- `power check`
Shows the summary of the current state of the power and margin values for the power supplies in a blade, a brick, or system. Example 3-50 shows sample output.
- `power vrm`

Shows output from voltage regulator module (VRM). One output indicates if the VRM is present (present), the other indicates if its outputting power (okay). See Example 3-52.

- `power <up|down>`

Power up or power down all of the power supplies. Example 3-51 shows sample output.

- `power up hold`

Note: For the power command set, the mnemonics `pwr`, `u`, and `d` may replace the words `power`, `up`, and `down`.

Example 3-48 `power` Command Output on Blade-Based Altix System

```
001c07-L1>power
Supply          State Voltage      Margin  Value
-----
B1 12V          on    11.871V        N/A
B1 1.85V AUX   NC     1.849V         N/A
B1 1.5V AUX    NC     1.494V         N/A
B1 1.85V       on     1.803V        normal    0
B1 1.2V        on     1.125V         low       1
B1 CPU0        on           N/A          N/A
B1 CPU1        <not present>
B6 12V          on    11.856V        N/A
B6 1.85V AUX   NC     1.868V         N/A
B6 1.5V AUX    NC     1.503V         N/A
B6 1.85V       on     1.797V         low       0
B6 1.2V        on     1.140V         low       1
B6 CPU0        on           N/A          N/A
B6 CPU1        <not present>
B7 12V          on    11.887V        N/A
B7 1.2V        on     1.195V        default    0
B7 2.5V        on     2.485V        default    0
B7 3.3V        on     3.314V         low       0
B7 1.2V        NC     1.204V         N/A
B7 1.85V       on     1.872V        default    0
RTRL 1.85V AUX NC     1.865V         N/A
RTRL 12V       on    11.918V        N/A
RTRL 1.85V     on     1.834V        normal    2
```


RTRR 1.85V AUX	NC	1.862V	N/A	
RTRR 12V	on	11.887V	N/A	
RTRR 1.85V	on	1.837V	normal	2
PS 0 [12V]	on	N/A	N/A	
PS 1 [12V]	on	N/A	N/A	
PS 2 [12V]	on	N/A	N/A	
PS 3 [12V]	on	N/A	N/A	

Example 3-49 power Command Output on Non-Blade Altix System

001c07-L1>power

Supply	State	Voltage	Margin	Value
-----	-----	-----	-----	-----
PWR T 48V	on	N/A	N/A	
PWR B 48V	on	N/A	N/A	
PWR 12V AUX	NC	12.340V	N/A	
PWR 3.3V AUX	NC	3.292V	N/A	
RTR T 1.85V	on	1.837V	normal	1
RTR T 5V AUX	NC	4.970V	N/A	
RTR T 3.3V AUX	NC	3.296V	N/A	
RTR B 1.85V	on	1.837V	normal	1
RTR B 3.3V AUX	NC	3.296V	N/A	
N0 12V	on	11.699V	N/A	
N0 2.5V	on	2.490V	normal	0
N0 1.85V	on	1.850V	normal	1
N0 1.2V	on	1.200V	normal	0
N0 1.25V	NC	1.248V	N/A	
N0 CPU A	on	N/A	N/A	
N0 CPU C	on	N/A	N/A	
N0 3.3V AUX	NC	3.287V	N/A	
N1 12V	on	11.731V	N/A	
N1 2.5V	on	2.493V	normal	0
N1 1.85V	on	1.852V	normal	1
N1 1.2V	on	1.204V	normal	0
N1 1.25V	NC	1.252V	N/A	
N1 CPU A	on	N/A	N/A	
N1 CPU C	on	N/A	N/A	
N1 3.3V AUX	NC	3.283V	N/A	
N2 12V	on	11.715V	N/A	
N2 2.5V	on	2.490V	normal	0
N2 1.85V	on	1.848V	normal	1
N2 1.2V	on	1.202V	normal	0
N2 1.25V	NC	1.246V	N/A	
N2 CPU A	on	N/A	N/A	
N2 CPU C	on	N/A	N/A	

N2	3.3V	AUX	NC	3.292V	N/A	
N3	12V		on	11.731V	N/A	
N3	2.5V		on	2.493V	normal	0
N3	1.85V		on	1.852V	normal	1
N3	1.2V		on	1.200V	normal	0
N3	1.25V		NC	1.248V	N/A	
N3	CPU	A	on	N/A	N/A	
N3	CPU	C	on	N/A	N/A	
N3	3.3V	AUX	NC	3.296V	N/A	

Example 3-50 power check Command Output

```
001c07-L1>power check
power appears on
```

Example 3-51 power up Command Output

```
001c07-L1>power up
001c07-L1>
```

Example 3-52 power vrm Command Output

```
001c07-L1>power vrm
VRM Type/Name      Present  Okay
-----
B1 1.85V           passed  passed
B1 CPU0            passed  passed
B1 CPU1            passed  passed
001c07-L1>
```

reboot_l1

The following `reboot_l1` command set reboots the L1 controller.

- `reboot_l1`
Reboots the L1 controller using the newest firmware image (firmware image A or firmware image B).
- `reboot_l1 <a|b>`
Reboots the L1 controller using the specified firmware image (A or B).
- `reboot_l1 <current|old|new|other>`

Reboot the L1 controller using either the current, older, newer, or other (non-current) flash image.

- `reboot_l1 addr <exp>`

Validates and boots the L1 controller flash image at the flash image address listed for *<exp>*.

- `reboot_l1 raw <exp>`

Boots the L1 controller flash image at the flash image address listed for *<exp>*.

reset

Use the `reset` command to perform a reset of the system. After the system controller receives a `reset` command, it sets various control and status signals back to their default values, and reboots the operating system. Example 3-53 shows sample output.

Example 3-53 `reset` Command Output

```
001c07-L1>reset
001c07-L1>
```

router

Use the following `router` command set to view and set router types, to set routers in different modes for reasons such as service, and to enable router ports. This command set is valid only for R-bricks and NUMALink modules:

- `router`

Shows the current router type. Example 3-53 shows sample output.

- `router meta`

Sets the router type to meta.

- `router repeater`

Sets the router type to repeater.

- `router ordinary`

Sets the router type to ordinary.

Note: The following commands are only used on Altix 3000 series systems with NUMALink 4.

- `router 6 port`
Enables all 6-port router ports.
- `router 8 port`
Enables all 8-port router ports.
- `router 8 port <str> <str> <str> <str>`
Enables all the 8-port router ports of the router specified with the IP address listed for the `<str> <str> <str> <str>` variable.
- `router service <str> <str> <str> <str>`
Sets into service mode, for repair, the 8-port router specified with the IP address listed for the `<str> <str> <str> <str>` variable.
- `router service off`
Disables the service mode after the repairs are completed.
- `router spare`
Configures the router for spare tear down.

Note: For the router command set, you can use `rtr` instead of `router`, `rep` instead of `repeater`, and `ord` instead of `ordinary`.

Example 3-54 router Command Output

```
001r19-L1>router
router type is ordinary
```

select

The following `select` command set displays and sets the device to serve as console I/O. (These commands are valid only for C-bricks.)

- `select`

Shows the current mode of console I/O. Example 3-55 shows sample output.

- `select <rack> <slot>`
Use `select <rack> <slot>` to set the rack and slot location of the device to be console I/O.
- `select subchannel console`
Use `select subchannel console` to set the current console subchannel as console I/O.
- `select subchannel <exp>`
Use `select subchannel <exp>` to set the mode of console I/O to the subchannel specified for `<exp>` (possible values for `<exp>` are 0, 1, 2, 3, or 4).
- `select subchannel <cpu>`
Use `select subchannel <a|b|c|d>` to set the mode of console I/O to a CPU subchannel (possible CPU subchannel values are a, b, c or d).
- `select filter <on|off>`
Use `select filter <on|off>` to enable (on) or disable (off) the console output filter.

Note: For this command set, you can use `sel` in the place of `select`, `sub` or `s` in the place of `subchannel`, and `con` in the place of `console`.

Example 3-55 `select` Command Output

```
001c07-L1>select
console input: 001c07 console
console output: not filtered.
```

serial

Use the following `serial` command set to view and set the system serial number (SSN) that is stored in each brick. On SGI Altix 4700 systems, you can use the `serial all` or `serial dimm` command to determine the DIMM source and manufacturer.

- `serial`
Shows the system serial number.

- `serial verify`
Shows the secure system serial number (SSN) settings stored in the NVRAM.
- `serial all`
Shows the brick serial number (BSN) and the SSN settings in NVRAM.
- `serial dimm`
Shows the dual-inline memory module (DIMM) part and serial number.
- `serial clear`
Clears the SSN.
- `serial <str> <str> <str> <str>`
Erases and reassigns the device's SSN. The variable `<str> <str> <str> <str>` is the value of a security key that is provided only to SGI employees.
- `serial security on`
Enables the system serial number (SSN) security. When this feature is enabled, it will not allow bricks or enclosures to power up if their SSN does not match those of neighboring bricks or enclosures. To change the SSN on a brick or enclosure, you need a key that can be obtained through a service request to SGI.

softreset

Use the `softreset` command to issue a software reset. In systems that include a C-brick, the `softreset` command is valid only for the C-brick.

Note: For this command, you can use the mnemonic `softrst` instead of the word `softreset`.

Example 3-56 `softreset` Command Output

```
001c07-L1>softreset
001c07-L1>
```

verbose

Use the `verbose` command to get a system maintenance port (SMP) prompt and get into character-echo mode.

version

Use the `version` command to view the version of the firmware that is currently running in the system controller. Example 3-57 shows sample output.

Example 3-57 `version` Command Output

```
001r19-L1>version
L1 0.7.27 (Image A), Built 04/28/2002 13:06:43 [P1 support]
```

L1 Controller Commands for Expert Users

This section describes commands that typically should only be used by highly experienced system administrators.

eeeprom

Use the following `eeeprom` command to view the raw eeeprom data.

- `eeeprom`
Shows brick eeeprom data.
- `eeeprom <eeeprom> <offset> <length>`
Shows brick eeeprom data at `<eeeprom> <offset> <length>`.

Use `eeeprom` to view brick eeeprom data. Example 3-58 shows sample output.

Example 3-58 `eeeprom` Command Output

```
001c01-L1>eeeprom
B0 BLADE ID (CH)
00 20 00 01 00 00 00 df
B0 BLADE ID (CIA), no data available (1)
B0 BLADE ID (BIA)
```

```
00 08 00 2d 58 52 c9 53 4f 4c 45 43 54 52 4f 4e
cc 42 41 53 45 49 4f 5f 42 4c 41 44 45 c6 4e 4e
50 37 39 35 cc 30 33 30 5f 32 30 34 33 5f 30 30
33 00 c2 5f 44 01 02 c2 30 30 c1 00 00 00 00 4e
B0 BLADE ID (PIA), no data available (1)
B0 BLADE ID (IUA)
00 01 30 01 00 03 00 00 00 00 08 00 01 00 0c
0b 45 01 03 00 20 1f 00 01 02 03 4a 00 03 4a 06
50 26 06 3f fc 06 3f b7 06 08 88 06 02 46 06 00
ec 18 00 6b 00 00 00 74
```

ioport

The `ioport` command set displays and sets the speeds and clock source of the I/O ports. This command set is valid only for bricks or enclosures with I/O ports (Altix 3000, Altix 3700, and Altix 3700 Bx2 systems).

- `ioport`
- `ioport 400|600`
- `ioport a 400|600`
- `ioport b 400|600`
- `ioport clksrc a|b`

Note: For the `ioport` command set, the mnemonic `ioprt` may replace the word `ioport`.

Example 3-59 shows output of the `ioport` command for compute brick.

Example 3-59 `ioport` Command Output for a Compute Brick

```
001x004-L1>ioport
001c01:
Compute I/O port B speed: 600 MHz (from attached I/O)
Compute I/O port D speed: 600 MHz (from attached I/O)
001x004-L1>
```

Example 3-60 shows output of the `ioport` command for the I/O brick (IX and PX).

Example 3-60 ioport Command Output for an I/O Brick

```
001x004-L1>ioport
PIC port A speed: 600 MHz (from NVRAM setting)
PIC port B speed: 600 MHz (from NVRAM setting)
PIC0 <-> PIC1 speed: 600 MHz
PIC0 <-> PIC2 speed: 600 MHz
PIC UST clock source: port B
001x004-L1>
```

istat

Use the following `istat` command set to view the status of the memory, queues, tasks, and memory allocation on SGI Altix 3000 series of systems. On all other Altix platforms, only the `istat queue` command is available.

- `istat memory`
Shows the status of the L1 controller memory. Example 3-61 shows sample output.
- `istat queues`
Shows the status of the L1 controller queues. Example 3-62 shows sample output.
- `istat tasks`
Shows the status of the L1 controller tasks. Example 3-63 shows sample output.
- `istat pmalloc`
Shows the status of the memory allocation. Example 3-64 shows sample output.

Example 3-61 `istat memory` Command Output

```
001c07-L1>istat memory
SYSMEM   [0x30005cf8]   Size: 41656 Avail: 26792
SMLPOOL  [0x30005cbc]   Size: 8704/ 272 Avail:    28
BIGPOOL  [0x30005c80]   Size: 16640/1040 Avail:    15
```

Example 3-62 `istat queues` Command Output

```
001c07-L1>istat queues
CMD_REQQ [0x30008a54]   Size: 10   Avail: 10   Msgs: 0
CMD_RSPQ [0x30008a9c]   Size: 10   Avail: 10   Msgs: 0
SMP_RQUE [0x3000c8a0]   Size: 10   Avail: 10   Msgs: 0
SMP_WQUE [0x3000c8e8]   Size: 10   Avail: 10   Msgs: 0
SMP_IQUE [0x3000c930]   Size: 10   Avail: 10   Msgs: 0
CTI_WQUE [0x3000f38c]   Size: 10   Avail: 10   Msgs: 0
USB_WQUE [0x3000ee7c]   Size: 10   Avail: 10   Msgs: 0
SCAN_QUE [0x30002440]   Size: 20   Avail: 20   Msgs: 0
FLASH_Q  [0x30009c64]   Size: 5    Avail: 5    Msgs: 0
BDR_WQUE [0x3000a0c8]   Size: 10   Avail: 10   Msgs: 0
```

Example 3-63 `istat tasks` Command Output

```
001c07-L1>istat tasks
MAIN_TSK [0x300019f8] (SLEEP      ) STACK: 2048 @ 0x300011f8 ( 904 used, 1144 free)
USB_CNTL [0x30007f38] (DRV_SUSP   ) STACK: 1536 @ 0x30007fe0 ( 288 used, 1248 free)
CMD_ITSK [0x300096e4] (READY     ) STACK: 3072 @ 0x30008ae4 (1936 used, 1136 free)
SMP_RTsk [0x3000e1c8] (EVT_SUSP   ) STACK: 2048 @ 0x3000c9c8 ( 800 used, 1248 free)
SMP_ITSK [0x3000e318] (QUEUE_SUSP) STACK: 2048 @ 0x3000d9c8 ( 768 used, 1280 free)
SMP_WTSK [0x3000e270] (QUEUE_SUSP) STACK: 2048 @ 0x3000d1c8 ( 416 used, 1632 free)
ENV_PITK [0x3000a9ec] (SEM_SUSP   ) STACK: 1024 @ 0x3000addc ( 276 used, 748 free)
ENV_FITK [0x3000aa94] (SEM_SUSP   ) STACK: 1024 @ 0x3000b1dc ( 260 used, 764 free)
ENV_TITK [0x3000ab3c] (SEM_SUSP   ) STACK: 1024 @ 0x3000b5dc ( 264 used, 760 free)
ENV_PMTK [0x3000abe4] (SEM_SUSP   ) STACK: 1024 @ 0x3000b9dc ( 260 used, 764 free)
ENV_FMTK [0x3000ac8c] (SEM_SUSP   ) STACK: 1024 @ 0x3000bddc ( 268 used, 756 free)
ENV_TMTK [0x3000ad34] (SEM_SUSP   ) STACK: 1024 @ 0x3000c1dc ( 324 used, 700 free)
BDR_RTsk [0x30009f30] (DRV_SUSP   ) STACK: 2048 @ 0x30016524 (1204 used, 844 free)
BDR_WTSK [0x3000a020] (QUEUE_SUSP) STACK: 1024 @ 0x30016d34 ( 532 used, 492 free)
CTI_RTsk [0x3000f23c] (DRV_SUSP   ) STACK: 2048 @ 0x3001717c ( 892 used, 1156 free)
CTI_WTSK [0x3000f2e4] (QUEUE_SUSP) STACK: 2048 @ 0x3001798c ( 572 used, 1476 free)
USB_RTsk [0x3000edd4] (READY     ) STACK: 1800 @ 0x300181d4 ( 732 used, 1068 free)
USB_WTSK [0x3000e0ec] (QUEUE_SUSP) STACK: 1500 @ 0x300188ec ( 480 used, 1020 free)
SCAN_TSK [0x30002398] (QUEUE_SUSP) STACK: 2048 @ 0x30001b98 ( 600 used, 1448 free)
I2C_HIGH [0x30007154] (SEM_SUSP   ) STACK: 1500 @ 0x300071fc ( 248 used, 1252 free)
I2C_LOW  [0x30006a2c] (SEM_SUSP   ) STACK: 1500 @ 0x30006ad4 ( 396 used, 1104 free)
FLASH_T  [0x30009bbc] (QUEUE_SUSP) STACK: 1024 @ 0x300097bc ( 316 used, 708 free)
```

Example 3-64 `istat pmalloc` Command Output

```
001c07-L1>istat pmalloc
```

```
small pool size:    272
small pool avail:   28
small pool used:    3
small pool max:     25
```

```
big pool size:     1040
big pool avail:    15
big pool used:     0
big pool max:      3
pmalloc calls:    0
prealloc calls:    3
```

```
history:
```

junkbus

The following `junkbus` command set provides the status of and sets the L1 controller-to-junkbus protocol setting. (Used on compute or router system components.)

- `junkbus`

Note: For the `junkbus` command set, you can use `jbus` instead of `junkbus`, `d` instead of `dump`, `o` for output, `i` for input, and `rst` for reset.

Use `junkbus` to determine the current settings for the L1-controller-to-junkbus protocol. Example 3-65 shows sample output for this command. PPP refers to point-to-point protocol. (Use the `l1dbg` command set to turn debugging on or off.)

Example 3-65 `junkbus` Command Output

```
001c003-L1>junkbus  
  
junkbus protocol is PPP
```

l1dbg and l2dbg

The following `l1dbg` command set displays and sets the state of communication debugging features. The `l2dbg` command is similar but affects L2 firmware. Both of the `l1dbg` and `l2dbg` commands can alter the behavior of the firmware, even across system controller power cycles as some values are stored in NVRAM. It is quite possible that you could lock up the firmware by changing selected values. There is **no** reset to default" feature.

Caution: The `l1dbg` and `l2dbg` commands select some internal debugging features of the L1 firmware. They should only be used by SGI engineers doing firmware development.

- `l1dbg meminfo|mem`
Shows system controller memory usage.
- `l1dbg menable`
Enables `malloc` function profiling.

- `l1dbg mdisable`
Disables `malloc` profiling.
- `l1dbg`
Gets L1 debugging settings.
- `l1dbg syscom <exp>`
Selects L1 <-> system debugging (mask).
- `l1dbg syscom <exp> <exp>`
Selects L1 <-> system debugging (slab & mask).
- `l1dbg irtr <exp>`
Sets L1 irouter debugging mode.
- `l1dbg env <exp>`
Turns L1 environmental debugging on (1=pwr, 2=fan, 4=tmp,8=setup).
- `l1dbg env off`
Turns L1 environmental debugging off.
- `l1dbg port on|off`
Turns L1 port interrupt debugging on/off.
- `l1dbg i2c on|off`
Turns L1 i2c interrupt debugging on/off.
- `l1dbg margin|mgn on|off`
Turns L1 voltage margin debugging on/off.
- `l1dbg console on|off`
Enables/disables console tracking messages.
- `l1dbg mempanic on|off`
Enables/disables L1 PANIC on `malloc` failures.
- `l1dbg pppdump on|off`
Enables/disables data dump on junkbus PPP errors.
- `l1dbg pppdelay <exp>`

Sets the length of JunkBus PPP read delay (debug).

- `l1dbg promreq|req on|off`
Turns PROM request debugging on/off.
- `l1dbg qsusp <exp> <exp>`
Sets queue suspend time for req/rsp and evt.
- `l1dbg printf on|off`
Enables/disables generating events on printf calls.
- `l1dbg cfg <exp>`
Sets debugging level for configuration exchange.
- `l1dbg nihi <exp>`
Sets NI and II port mask for debugging.
- `l1dbg b2b <exp>`
Enables B2B debugging.
- `l1dbg error`
Tests error message decoding.
- `l1dbg packets`
Shows outstanding packets.
- `l1dbg packets <exp>`
Sets packet malloc/free debugging.
- `l1dbg printf <str>`
Generates printf event.
- `l1dbg flashignore <exp> <exp>`
Ignores next <exp> flash pkts, test timeout handling in flashsc.

Use `l1dbg` to display the state of the communication debugging features. Example 3-66 shows sample output.

Example 3-66 l1dbg Command Output

```
001c07-L1>l1dbg
L1 irouter debugging is 0
L1 junkbus debugging is slab mask 0x0, mask 0x0
L1 environmental debugging is off (0x00)
L1 port interrupt debugging is off
L1 i2c interrupt debugging is on
L1 voltage margin debugging is off
L1 PROM request debugging is off
L1 irouter req/rsp suspend is 300, event suspend is 1000
L1 data dump on PPP error is off; junkbus PPP read delay 0
L1 console debugging is off
L1 malloc failure PANICs are disabled
L1 printf call events enabled
```

Use `l1dbg junkbus <on|off>` to turn on communication debugging for the L1 controller-to-junkbus-ASIC communication. Example 3-67 shows sample output from the `l1dbg junkbus on` command.

Example 3-67 l1dbg junkbus on Command Output

```
001c07-L1>l1dbg junkbus on
L1 irouter debugging is off
L1 junkbus communication debugging is on
L1 environmental debugging is off
L1 port interrupt debugging is off
L1 i2c interrupt debugging is off
L1 voltage margin debugging is off
```

margin

Use the `margin` command to show the margin status of all voltages or to set a margin value.

- `s|b<rng> margin|mgn`
All voltages margin status.
- `s|b<rng> margin|mgn default|d|low|l|norm|n|high|h`
Sets all voltages to ROM defaults.

- `s|b<rng> margin|mgn <exp> default|d|low|l|norm|n|high|h`
Sets supply index *<exp>* to specified margin ('mgn' command shows indexes).
- `s|b<rng> margin|mgn <exp> <exp>`
Sets margin on supply index *<exp1>* to value *<exp2>* (*mgn* command shows indexes).
- `s|b<rng> margin|mgn dimm v|voltage <str>`
Sets DIMM margin based on specified target voltage *<exp>*.

Use the `margin` to display the state of the communication debugging features. Example 3-66 shows sample output.

Example 3-68 margin Command Output

```
001c07-L1>margin
Supply          State Voltage      Margin  Value Index
-----
B0 5V           on    5.019V    normal    0    0
B0 3.3V         on    3.280V    normal    0    1
B0 1.85V        on    1.849V    normal    0    2
B0 1.2V         on    1.193V    normal    0    3
B2 1.85V        on    1.849V    normal    2    0
B2 1.2V         on    1.203V    normal    0    1
B4 1.85V        on    1.849V    normal    2    0
B4 1.2V         on    1.203V    normal    0    1
RTRL 1.85V      on    1.834V    normal    2    0
RTRR 1.85V     on    1.834V    normal    2    0
001c01-L1>
```

scan

Use the scan command to perform a JTAG boundary scan hard reset or soft reset.

- scan reset
Performs hard and soft JTAG reset.
- scan reset hard
Performs hard JTAG reset (via TRSTN).
- scan reset soft
Performs soft JTAG reset (via TMS).
- scan reset both
Performs hard and soft JTAG reset.
- scan sel <exp> <exp>
Select SIC: <addr> <CER>.
- scan sel <exp> <exp> <exp>
Selects SIC: <addr> <CER> <MR>.
- scan sel <exp> <exp> <exp> <exp>
Selects SIC: <addr> <CER> <MR> <IOR> .

- `scan ids <exp>`
Reads and display IDCODE registers.
- `scan set trst 0`
Directs control of JTAG TRST signal.
- `scan set trst 1`
Directs control of JTAG TRST signal.
- `scan set tck 0`
Directs control of JTAG TCK signal.
- `scan set tck 1`
Directs control of JTAG TCK signal.
- `scan set tms 0`
Directs control of JTAG TMS signal.
- `scan set tms 1`
Directs control of JTAG TMS signal.
- `scan set tdi 0`
Directs control of JTAG TDI signal.
- `scan set tdi 1`
Direct control of JTAG TDI signal.
- `scan set psi 0`
Directs control of JTAG PSI signal.
- `scan set psi 1`
Directs control of JTAG PSI signal.
- `scan get`
Displays state of JTAG TAP signals.
- `scan debug <exp>`
Sets scan debug message level.
- `scan debug`

Display scan debug message level.

- `scan info`

Displays scan information.

- `scan count`

Counts length of IR and BYPASS registers.

- `scan count <exp>`

Counts length of IR and BYPASS registers.

Example 3-66 shows sample `scan chip` output.

Example 3-69 scan chip Command Output

```
001c07-L1>scan chip
Valid target chips are: B0:TIO0 B2:SHUB0 B2:LBI0 B4:SHUB0 B4:LBI0
RTRL:PSIC0 RTRL:RTR0 RTRL:RTR1 RTRR:PSIC0 RTRR:RTR0 RTRR:RTR1
001c01-L1>
```

rmmr

Use the following `rmmr` command to read a memory-mapped register.

- `wmnr | mnr_wr <str> <exp> <exp> <exp>`
Writes chip `<str>` MMR at address `<exp>` with value `<datahi> <datalo>`

tdr

Use the following `tdr` command to read test data registers (TDRs).

- `tdr get user <str>`
Gets all user-override TDR field values: `tdr set user <chip_name>`.
- `tdr get cfg <str>`
Gets all config override TDR field values: `tdr get cfg <chip_name>`.
- `tdr get o|over|overrides <str>`
Get all override TDR field values: `tdr get over <chip_name>`.
- `tdr get <str>`
Gets list of valid TDRs: `tdr set <chip_name>`.
- `tdr get <str> <str>`
Gets all TDR field values: `tdr set <chip_name> <tdr_name>`.
- `tdr get <str> <str> <str>`
Gets TDR field value: `tdr set <chip_name> <tdr_name> <field_name>`.
- `tdr set <str> <str> <str> <exp>`
Sets TDR field value: `tdr set <chip_name> <tdr_name> <field_name> <data>`.
- `tdr set <str> <str> <str> <exp> <exp>`

Sets TDR field value: `tdr set <chip_name> <tdr_name> <field_name> <data_hi> <data_lo>`.

- `tdr dmp|dump <str> <str>`

Dumps TDR structure contents: `<chip_name> <tdr_name>`.

- `tdr rst|reset <str>`

Resets TDR fields to default values: `<chip_name> <tdr_name>`.

- `tdr rst|reset <str> <str>`

Resets TDR fields to default values: `<chip_name> <tdr_name>`.

test

Use the following `test` command set to test various components in the L1 controller.

- `test|tst i2c`

Runs i2c test one pass.

- `test|tst i2c <exp>`

Runs i2c test `<exp>` passes.

- `s|b<rng> test|tst ioexp set <exp> <exp>`

At ioexp index `<exp1>`, sets to value `<exp2>`.

- `s|b<rng> test|tst ioexp get <exp>`

Gets value of ioexp at index `<exp>`.

- `s|b<rng> test|tst ioexp get all`

Gets value of all ioexpanders.

- `test|tst intr`

Gets I2C interrupt counts.

- `test|tst display`

Resets LCD display (self-test).

- `test|tst exception|exc mem`

Tests memory fault handling.

- `test|tst exception|exc stack`
Tests stack corruption error handling.
- `test|tst exception|exc wdog`
Tests software watchdog reset.
- `test|tst exception|exc fatal`
Tests system panic.

Note: For this command set, you can use the mnemonic `tst` and `exc` in place of the words `test` and `exception`.

Use the commands in this set as follows:

Example 3-70 `test intr` Command Output

```
001r19-L1>test intr
I2C interrupts, High: 25 Low: 406
DS1780 0, 1: 0 (0x00) 2: 0 (0x00)
IOExp 0: 197
IOExp 1: 0
IOExp 2: 0
IOExp 3: 0
IOExp 4: 137
IOExp 5: 0
IOExp 6: 11
IOExp 7: 0
```

uart

The following `uart` command set displays the status of the following UARTs (universal asynchronous receivers/transmitters): CTI, CTC, SMP, and Junkbus.

- `uart`
Sets displays the status of the following UARTs (universal asynchronous receivers/transmitters): CTI, CTC, SMP, and Junkbus.
- `uart junkbus reset` (You can use `r` in place of `reset` for this command)
Resets the UART of the Junkbus ASIC.

Example 3-71 shows sample output from the `uart` command.

Example 3-71 `uart` Command Output

```
001c20-L1>uart
      Baud   Read   Read   Read   Read   Read   Write  Write  Write
UART  Rate   State  Status Timeouts Breaks Errors State  Status Timeouts
-----
CTI   107142  Discon Ready  0       0       0       Discon Ready  0
CTC   107142  Discon Ready  0       0       0       Discon Ready  0
SMP   37500   Discon Ready  0       0       1       Discon Ready  0
BED   57692   Connect Suspend 0       0       13      Connect Ready  0
```

usb

The `usb` command displays status information for the universal serial bus, (USB) L1 port. SSEs use this port to access L1 controllers in systems that do not have L2 controllers. This command is only used on Altix 3000 series systems with NUMALink 3. Example 3-72 shows sample output.

Example 3-72 `usb` Command Output

```
001c20-L1>usb
Device: 0 Disconnects: 2 Bus Resets: 13
Endpoint State      Status      Stalls Errors Timeouts
-----
Control Active   Suspended  43     0     0
Read   Active   Ready     0     0     0
Write  Active   Ready     0     0     0
```

L2 Controller Commands

The following subsections describe the L2 controller commands and command sets, which are listed alphabetically. Examples of output are included where applicable.

autopower

The following `autopower` command set enables, disables, aborts, and shows the current auto power status.

- `autopower`

Shows the current auto power-up setting.

- `autopower on`

Enables the auto power on.

- `autopower off`

Disables the auto power on.

- `autopower abort`

Aborts the auto power on.

Note: For these commands, the mnemonic `apwr` may replace the word `autopower`.

Use the `autopower` command to view the current auto power-up setting. Example 3-73 shows the sample output.

Example 3-73 `autopower` Command Output

```
L2>autopower
auto power up appears disabled
L2>
```

config

The following `config` command set displays configuration information.

- `config|cfg`

Shows configuration information (normal).

- `config|cfg summary|s`

Shows configuration information (summary).

- `config|cfg verbose|v`

Shows configuration information (verbose).

- `config|cfg l2`

Shows L2 configuration information (normal).

- `config|cfg l2 verbose|v`

- Shows L2 configuration information (verbose).
- `config|cfg l|list`
Shows configuration information (list).
- `config|cfg t|times`
Shows various L2 process information.
- `config|cfg snap`
Snaps the current system configuration to `/work/default.scf`.
- `config|cfg snap <str>`
Snaps the current system configuration to file specified by `<str>`.
- `config|cfg diff v|verbose`
Shows changes since last snap to file `/work/default.scf`.
- `config|cfg diff v|verbose <str>`
Shows changes since last snap to file specified by `<str>`.
- `config|cfg diff v|verbose <str> <exp>`
Shows changes with file `<str>`; increment racks.
- `config|cfg diff v|verbose <str> <str>`
Shows changes between two system configuration files.
- `config|cfg diff`
Shows changes since last snap to file `/work/default.scf`.
- `config|cfg diff <str>`
Shows changes since last snap to file specified by `<str>`.
- `config|cfg diff <str> <exp>`
Shows changes with file `<str>`; increment racks.
- `config|cfg diff <str> <str>`
Shows changes between two system configuration files.
- `config|cfg dump`
Shows the contents of system configuration file `/work/default.scf`.

- `config|cfg dump <str>`

Shows the contents of system configuration file specified by *<str>*.

Note: For these commands, the mnemonic `cfg` may replace the word `config`, and `s`, `v`, and `re` may replace the words `summary`, `verbose`, and `rescan`.

Use the `config` command to view configuration information for all the L1 controllers connected to all the L2 controllers in the server system (see Example 3-74).

Example 3-74 `config` Command output

```
L2>config
L2 10.25.4.193: - 0241
L1 10.25.4.193:0:0      - 001c01
L1 10.25.4.193:0:1      - 001c11
L1 10.25.4.193:0:2      - 001c21
L1 10.25.4.193:0:3      - 001c31
L1 10.25.4.193:0:4      - 001r41
L1 10.25.4.193:0:5      - 002c01
L1 10.25.4.193:0:6      - 002c11
L1 10.25.4.193:0:7      - 002c21
L1 10.25.4.193:0:8      - 002c31
L1 10.25.4.193:0:9      - 002r41
L1 10.25.4.193:0:10     - 004r51
L1 10.25.4.193:0:11     - 006r51
L2 10.25.4.195: - 0341 (LOCAL)
L1 10.25.4.195:0:0      - 003c01
L1 10.25.4.195:0:1      - 003c11
L1 10.25.4.195:0:2      - 003c21
L1 10.25.4.195:0:3      - 003c31
L1 10.25.4.195:0:4      - 003r41
L1 10.25.4.195:0:5      - 003r51
L1 10.25.4.195:0:6      - 004c01
L1 10.25.4.195:0:7      - 004c11
L1 10.25.4.195:0:8      - 004c21
L1 10.25.4.195:0:9      - 004c31
L1 10.25.4.195:0:10     - 004r41
L1 10.25.4.195:0:12     - 005r51
```

The number that follows the port number, after the second colon, is the L1 index.

Bricks are referenced by their racks and slot or bay locations. These values are stored in non-volatile memory on the L1. Virtually all system controller communications require that each brick have a valid and unique rack and slot. For information how to determine the L1 index value for a brick, see TBD.

The number that follows the L1 index, after the dash, is the brick identification number (for example, 001c07). The first three digits of the brick identification number indicate the rack in which the brick resides. The fourth digit indicates the type of brick (see Table 3-3). The last two digits indicate the slot position in which the brick resides.

Table 3-3 Valid Brick Types

Type	Description
c	C-brick
i	I-brick
p	P-brick
r	R-brick
x	X-brick
?	Unknown brick type

Use the `config summary` command to view a summary of configuration information (see Example 3-75).

Example 3-75 `config summary` Command output

```
L2>config summary
qs115-3-341-L2>cfg summary
L2s: 3
L1s: 32
  R Bricks: 10
  C Bricks: 22
L2>
```

The `config list` commands can be used to list all of the bricks in a system (see Example 3-76).

Example 3-76 config list Command output

```
L2>config list
006r51    006r41    006c31    006c21    006c11    006c01
005r51    005r41    005c11    005c01
004r51    004r41    004c31    004c21    004c11    004c01
003r51    003r41    003c31    003c21    003c11    003c01
002r41    002c31    002c21    002c11    002c01
001r41    001c31    001c21    001c11    001c01
32 bricks
L2>
```

The `config snap` command can be used to take a snapshot of the current system configuration from the perspective of the system controllers. The file is written to the L2's memory file system (the L2 has a limited amount of flash memory dedicated to a persistent file system).

All bricks as well as NUMALink connections are recorded (see Example 3-77).

Example 3-77 config snap Command output

```
L2>config snap
Writing system configuration file: /work/default.scf
```

The `config diff` command can be used to quickly identify a missing piece of hardware or disconnected cable (see Example 3-78).

Example 3-78 config diff Command output

```
L2>config diff
HARDWARE and FILE (/work/default.scf) = 0 differences
L2>
```

date

Use the following `date` command set to view and set the current date and time used by the L2 controller.

- `date`
Shows the current date and time value used by the L2 controller. Example 3-79 shows sample output.
- `date <str>`

Sets the date and time value used by the L2 controller. The variable *<str>* is a time value in the form *yyyymmddHHMMSS* (where *yyyy* is the four-digit year, *mm* is a two-digit month, *dd* is a two-digit day, *HH* is a two-digit hour, *MM* is a two-digit minute, and *SS* is a two-digit second).

- `date tz`

Shows the time zone offset used by the L2 controller.

- `date tz <str>`

Sets the time zone offset used by the L2 controller. The variable *<str>* is a maximum of +12 (for 12 hours ahead of GMT) and a minimum of -12 (for 12 hours behind GMT).

Note: The time does not automatically update for daylight savings time. You need to manually set the time using the `date <str>` command.

Example 3-79 `date` Command Output

```
L2>date
09/14/4655 22:47:07
```

destination

The following `destination` command set displays the brick identification numbers of bricks that are the destinations of L1 commands, and sets the bricks that are the destinations for L1 commands.

- `destination`

Shows the brick identification numbers of the bricks that are the destinations of L1 commands. Example 3-81 shows sample output.

- `rack <rng> slot <rng> destination`

Sets individual bricks that are the destinations for L1 commands. The variable *<rng>* specifies a rack number(s) and slot number(s).

Note: For the *<rng>* variable, you can enter a single number, or two numbers separated by a hyphen to indicate a range, or numbers separated by commas to indicate separate items. For *<rng>*, you can also enter an asterisk or the word “all” to select all the items available.

- destination reset

Resets the destination of L2 commands to all bricks in all racks and all slots. Example 3-80 shows sample output.

Note: For this command, the mnemonic *rst* may replace the word *reset*.

Example 3-80 destination reset Command Output

```
L2>destination reset
default destination reset to all bricks and slots
```

Note: On large systems that are partitioned, the L2 *select partition* command will implicitly set the correct destination based on the partition selected.

Example 3-81 destination Command Output

```
L2>destination
all racks, all slots
```

Example 3-82 rack *<rng>* slot *<rng>* destination Command Output

```
L2>r 1 s 7 destination
1 default destination(s) set
L2> destination
001c07 (127.0.0.1:1:0)
```

Note: For these commands, the mnemonic *dest* may replace the word *destination*. Also, the mnemonics *r* and *s* may replace the words *rack* and *slot*.

dhcpserver

The following `dhcpserver` command set displays the setting for the current DHCP server (see Example 3-83) and enables and disables the DHCP server.

- `dhcpserver`
Displays the setting for the current DHCP server (see Example 3-83).
- `dhcpserver on|enable`
Enables the DHCP server.
- `dhcpserver off|disable`
Disables the DHCP server.

Note: For this command, the mnemonic `dhcps` may replace the word `dhcpserver`.

Example 3-83 `dhcpserver` Command Output

```
L2>dhcpserver
DHCP server startup enabled (if no other DHCP server is found).
```

env

The following `env` command set provides an environmental status summary.

- `env summary`
Generates an environmental summary of an entire system. Example 3-84 shows sample output.

Note: For this command, `r` may replace the word `rack`; `s` may replace the word `slot` and may also replace the word `summary`.

Example 3-84 `env summary` Command Output

```
L2>env summary
all environmental conditions appear normal
L2>
```

help

Generates a list of all of the system controller commands. Use the `help <command>` command to display more information on a single command. The variable `<command>` is the name of a command.

Note: For this command, `hlp` may replace the word `help`.

ip

Use the following `ip` command set to set, clear, and show the L2 controller static IP address configuration settings.

- `ip`
Shows the L2 controller static IP address settings. Example 3-85 shows sample output.
- `ip <addr> <netmask>`
Sets the address and netmask L2 static IP configuration parameters.
- `ip <addr> <netmask> <broadcast>`
Sets the address, netmask, and broadcast L2 static IP configuration parameters.
- `ip clear|reset`
Clears the L2 static IP address settings.
- `ip gateway|gw <addr>`
Shows the L2 controller static IP address settings.

Example 3-85 `ip` Command Output

```
essc1-001-L2>ip
addr: 137.38.88.197 netmask: 255.255.255.0 broadcast addr: 137.38.88.255
essc1-001-L2>
```

l1

The following `l1` command set enters L1 mode or sends an L1 controller command to a specified L1 controller.

- `l1`
Engages the default L1 command processor or enter the l1 mode.
- `rack <rng> slot <rng> l1`
Engages the L1 command processor of a brick in a specified rack and slot. The `<rng>` variable following `rack` is one or more rack numbers, and the `<rng>` variable following `slot` is one or more slot numbers. Example 3-86 shows sample output.
- `<rack>.<slot> l1`
Engages the L1 command processor for a specified rack and slot.
- `rack <rng> slot <rng> l1 <command>`
Sends a command to a destination brick without changing the default destination value. The `<rng>` variable following `rack` is one or more rack numbers, the variable `<rng>` following `slot`, is one or more slot numbers, and the variable `<command>` is an L1 controller command such as `config`. Example 3-87 shows sample output.
- `<rack>.<slot> l1 <command>`
Sends a command to a destination brick without changing the default destination value.
- `l1 <command>`
Sends a specified L1 controller command to a default destination.

Note: For the `<rng>` variable, you can enter a single number, or two numbers separated by a hyphen to indicate a range, or numbers separated by commas to indicate separate items. For `<rng>`, you can also enter an asterisk or the word “all” to select all the items available.

Example 3-86 `rack <rng> slot <rng> l1` Command Output

```
L2>r 1 s 19 l1
entering L1 mode 001r19, <CTRL-T> to escape to L2

001r19-L1>
```

Example 3-87 `rack <rng> slot <rng> l1 config` Command Output

```
L2>r 1 s 7 l1 config
:0 - 001c07
```

```
:1 - 001i21  
L2>
```

Note: For these commands, the mnemonics *r* and *s* may replace the words *rack* and *slot*.

12

Use the following 12 command set to engage and lock a specified L2 command processor or to send a command to specified L2 command processor(s).

- 12
Engages and locks the L2 command processor. Example 3-88 shows sample output.
- rack <rng> slot <rng> 12 <command>
Sends a specified command to a specified L2 controller(s).
- <ip> 12 <command>
Sends a specified command to an L2 controller specified with its IP address.

Note: For these commands, the mnemonics *r* and *s* may replace the word *rack* and *slot*.

Example 3-88 12 Command Output

```
L2>12  
L2 command processor engaged, <CTRL-D> for console mode.
```

Note: For the <rng> variable, you can enter a single number, or two numbers separated by a hyphen to indicate a range, or numbers separated by commas to indicate separate items. For <rng>, you can also enter an asterisk or the word “all” to select all the items available.

l2find

The `l2find` command lists all of the L2 controllers that are connected together on the same subnet (see Example 3-89). Note that the `l2find` command does not show the L2 to which you are connected.

Example 3-89 l2find Command Output

```
L2>l2find
13 L2's discovered:
```

IP	SSN	NAME	RACK	FIRMWARE

[L2's with System Serial Number NOT set]				
137.38.82.101			000	L3 controlle
137.38.82.102			000	L3 controlle
[L2's with different System Serial Numbers]				
137.38.82.156	L1000625		001	1.8.0
137.38.82.51	N1000405	itsys1	111	1.7.7
137.38.82.58	L0000002	klsys2	002	1.7.7
137.38.82.159	L0000138	klsys4	004	1.7.9
137.38.82.162	L0000005	klsys5	005	1.7.7
137.38.82.57	L0000007	klsys7	007	1.8.0
137.38.82.55	L0000123	perch	009	1.7.7
137.38.82.158	L0000018	shrimp	018	1.7.7
137.38.82.50	N0000001	sn2-dbg2	022	1.7.7
137.38.82.52	M0000114	snapper	015	1.7.7
137.38.82.157	L0000119	whale	008	1.7.7

log

The following `log` command set displays the contents of the log, resets the log, and writes an entry into the log.

- `log`
Shows the contents of the log. If the log is empty, the output from the `log` command is `log is empty`.
- `log clear|reset`
Empties the log.
- `log insert <entry>`

The variable *<entry>* is text to enter in the log.

Example 3-90 shows sample output where the text for *<entry>* is “Start the Test.”

Example 3-90 log insert Start the Test Command Output

```
L2>log insert Start the Test
log entry made.
```

multisys

Use the following `multisys` command set to enable, disable, and show the settings for multiple system network sharing.

- `multisys`
Shows the current setting for the L2 controller multiple system network sharing.
Note: Example 3-91 shows sample output.
- `multisys on`
Enables multiple L2 controller system network sharing. Network sharing enables L2s to communicate only with other L2s that have the same serial number.
- `multisys off`
Disables multiple L2 controller system network sharing. When L2 controller system network sharing is disabled, L2s communicate with all L2s.

Note: For these commands, `msys` can replace the word `multisys`.

Example 3-91 multisys Command Output

```
L2>multisys
L2 multiple system network support enabled.
L2's will only connect to L2s with same system SN.
L2>
```

nvramp reset

The `nvramp reset` command returns the NVRAM settings of the L2 controller to the factory default settings.

password

The `password` command can be used to display, set, change, or clear an L2 controller password. There is no password length minimum. The password is truncated at 15 characters.

- `password pw`
Displays L2 password status.
- `password|pw set <str>`
Sets/changes the password required to access this L2.
- `password|pw clear`
Clears the L2 password (only allowed from L2 Console/Modem serial ports).
- `password|pw clear <str>`
Clears the L2 password (<str> is current L2 password).

pbay

Use `pbay config` to report the bricks connected to each power bay . Use `pbay <command>` to send a specified command to one L1 controller connected to each power bay.

ping

Use the following `ping` command set to set the ping function between L2 controller and L2 controllers or L2 controller and L1 controllers.

Note: This command is only used for debugging purposes.

- `ping`
Displays the L2 controller to L2 controller or L2 controller to L1 controller ping configurations.
- `ping min <frequency in seconds>`
Sets the L2 controller to L2 controller ping response to a minimum specified in <frequency in seconds>.

- `ping max <frequency in seconds>`
Sets the L2 controller to L2 controller ping response to a maximum specified in *<frequency in seconds>*.
- `ping l2 <frequency in seconds>`
Sets the L2 controller to L2 controller ping frequency.
- `ping sender <1 or 0>`
Use `ping sender <1 or 0>` to enable (1) or disable (0) sender ping as a valid response.
- `ping l1 <frequency in seconds>`
Sets `ping l1<frequency in seconds>` to set L2 controller to L1 controller ping frequency in seconds.
- `ping reset`
Resets L2 controller to L2 controller and L2 controller to L1 controller ping configuration to default.

power

The following `power` command set displays the power status of the bricks and powers up and powers down the bricks.

- `power`
Shows the power status of each brick identified by the destination set of commands. Example 3-92 shows sample output.
- `power up`
Powers up each brick identified by the destination set of commands.
- `power down`
Powers down each brick identified by the destination set of commands.
- `power summary`
Displays a power status summary. Example 3-93 shows sample output.

Note: For the power command set, the mnemonics `pwr`, `u`, `d`, and `s` may replace the words `power`, `up`, `down`, and `summary`. The mnemonics `r` and `s` may replace the words `rack` and `slot`.

Example 3-92 power Command Output

```
L2>power
001i21:
power appears on
001r19:
power appears on
001c07:
power appears on
001c10:
power appears on
```

Example 3-93 power summary Command Output

```
L2>power summary
all 2 bricks appear powered off
margin: 1 default, 0 low, 0 normal, 0 high, 0 custom, 0 other
L2>
```

quit

Use the `quit` command to exit the L2 command processor.

rackid

Use the following `rackid` command set to display or set the L2 controller rack ID.

Note: This command only applies to non-blade Altix systems.

- `rackid`
Displays the rack ID of an L2 controller. Example 3-94 shows sample output.
- `rackid <decimal exp>`

Sets the rack ID for an L2 controller, where *<decimal exp>* is the rack ID number for the L2 controller

Example 3-94 rackid Command Output

```
essc1-001-L2>rackid
rack ID is 001
essc1-001-L2>
```

reboot_l2

The `reboot_l2` command reboots the L2 controller using the newest firmware image. The `reboot_l2 force` command reboots the L2 controller even if the firmware image is invalid.

reset

Use the following `reset` command set to reset brick(s).

- `reset`
Resets the individual default brick or bricks.

Note: For the `reset` command set, you can use `rst` in the place `reset`, `r` in the place of `rack`, and `s` in the place of `slot`.

Caution: If you enter a `reset` command at the L2 prompt, it will reset all the bricks in your server system, which can create problems for your operating system if the operating system is running when you enter the `reset` command. So make sure your operating system is shut down before using the `reset` command.

select

The following `select` command set displays and sets the brick ID number of the compute brick and subchannel that receives console input.

- `select`

Shows the brick ID number and subchannel that receives console input.

Example 3-95 shows sample output. In this example, the brick ID is a C-brick in rack 2, slot 1 (002c01) and the subchannel is the console channel.

- `select terse`

Shows a less detailed display of information identifying the brick ID number and the subchannel that receives the console input.

- `select subchannel console <blade|node cpu>`

Selects the current console as the subchannel of the blade or node that receives the console input. Console 0, console 1, console 2, and so on. 0, 1, 2 refers to the node or blade.

- `select <subchannel>`

Selects subchannel which is <blade|node><cpu>. <cpu> is a|b|c|d. So 0a is blade|node 0, cpu A. 3c is blade|node 3, cpuC. Example 3-96 shows sample output.

- `select subchannel <exp>`

Selects the subchannel <exp> CPU of the brick that receives that console input.

- `select <exp> <exp>`

Use `select <exp> <exp>` to enter the rack and slot of the brick that receives the console input.

- `select <rack>.<slot>`

Selects the brick (brick ID) that receives console input. The variable <rack> is a rack number and the variable <slot> is a slot number. In Example 3-97, the brick ID is a C-brick in rack 3, slot 1 (003c01).

- `select reset`

Resets the selection of the brick to receive the console input to the default setting.

- `select partition <exp>`

Use `select partition <exp>` to select the partition and system console to receive the console input.

- `select filter on`

Turns on the console output filter.

- `select filter off`

Turns off the console output filter. Example 3-98 shows sample output.

Note: For this command set, the mnemonic sel, t, sub (or s), con, part (or p), res may replace the words select, terse, subchannel, console, partition, and reset.

Example 3-95 select Command Output

```
L2>select
console input: 002c01 console
console output: not filtered
console detection: L2 detected
```

Example 3-96 select subchannel <a|b|c|d> Command Output

```
L2>select subchannel c
console input: 003c01 CPUc
console output: not filtered

console detection: L2 detected
```

Example 3-97 select <rack>.<slot> Command Output

```
L2>select 3.1
console input: 003c01 console
console output: not filtered
console detection: L2 detected
```

Example 3-98 select filter on Command Output

```
L2>select filter on
console input: 003c01 console
console output: filtered

console detection: L2 detected
```

serial

Use the following `serial` command set to view and set the L2 controller system serial number.

- `serial`
Shows the L2 controller system serial number. Example 3-99 shows sample output.
- `serial set <str>`
Sets the L2 controller system serial number. The variable `<str>` is the L2 system serial number. Example 3-100 shows sample output.

Example 3-99 `serial` Command Output

```
L2>serial
L2 system serial number: not set.
```

Use `serial set <str>` to

Example 3-100 `serial set <str>` Command Output

```
L2>serial set L0000010
L2 system serial number set to L0000010.
L2> serial
L2 system serial number: L0000010.
```

shell

Use the `shell` command to escape to the L2 operating system. Example 3-101 shows sample output.

Example 3-101 `shell` Command Output

```
L2>shell
bash$ rm core
bash$ exit
exit
L2>
```

Note: For this command, you may use the character `!` instead of the word `shell`.

smp

The following `smp` command set displays the status of the system maintenance port (SMP) network connection. These commands are also used to join a specified SMP group or leave a current SMP group.

Note: This command is for expert users.

- `smp`
Shows the status of the SMP network connection. Example 3-102 shows a sample output.
- `smp verbose`
Displays an expanded status of the SMP network connection.
- `smp join <exp>`
Joins a specified SMP group identified by the `<exp>` variable.
- `smp leave`
Exits the current SMP group.

Note: For this command set, the mnemonic `v` may replace the word `verbose`.

Example 3-102 `smp` Command Output

```
essc1-001-L2>smp
Session   Who           Group  Mode  Console
-----
>>>  6    network port    6     L2    038c01 console (default)
      1    modem port      1     L2    038c01 console (default)
      1    modem port      1     L2    038c01 console (default)

essc1-001-L2>
```

sysname

Use the following `sysname` command set to display or set the system name (name of the L2 controller).

- `sysname`

Use the `sysname` command to display the name of the system. Example 3-103 shows a sample output.

- `sysname <str>`

Use `sysname <str>` to set a system name where the variable `<str>` is the system name. Example 3-104 shows sample output.

Example 3-103 `sysname` Command Output

```
essc1-001-L2>sysname
L2 system name : essc1.
essc1-001-L2>
```

Example 3-104 `sysname <str>` Command Output

```
essc1-001-L2>sysname essc2
L2 system name : essc2.

essc2-001-L2>
```

version

Use the following `version` command set to view information about the L1 and L2 controller firmware versions running in the system controller.

- `version`

Displays the L2 controller firmware version currently running in the system controller, as shown in Example 3-105.

- `l1 version`

Use `l1 version` to view the L1 controller firmware version currently running in the system controller.

- `version verbose`

Displays the verbose view of the L2 controller firmware version currently running in the system controller.

- `version summary`
Displays a summary of the L1 and L2 controller firmware versions currently running in the system controller.
- `version check`
Displays any L1 and L2 controller firmware version mismatches in the system.
- `rack <rng> slot <rng> version summary`
Use `rack <rng> slot <rng> version summary` to display the L1 and L2 controller firmware versions currently running in the rack(s) and slot(s) that you specify with the `<rng>` variable following rack and the `<rng>` variable following slot.

Note: For the `<rng>` variable, you can enter a single number, or two numbers separated by a hyphen to indicate a range, or numbers separated by commas to indicate separate items. For `<rng>`, you can also enter an asterisk or the word “all” to select all the items available.

- `rack <rng> slot <rng> version check`
Use `rack <rng> slot <rng> version summary` to display the L1 and L2 controller firmware versions mismatches in the rack(s) and slot(s) that you specify with the `<rng>` variable following rack and the `<rng>` variable following slot.

Example 3-105 `version` Command Output

```
L2>version
L2 version 0.4.0
```

Index

Symbols

*(asterisk character) command, 53

A

autopower command, 54, 111

B

blades

- identifying, 40
- types, 40

brick command, 55

brick identification, 39

C

command

- *(asterisk character), 53
- autopower, 54, 111
- brick, 55
- config, 59, 112
- cpu, 61
- date, 62, 116
- debug, 63
- destination, 117
- dhcpserver, 119
- display, 65
- eeprom, 95

env, 66, 119

fan, 72

flash, 72

help, 73, 120

history, 75

ioport, 96

ip, 120

istat, 98

junkbus, 100

l1, 76, 120

L1 controller commands, 52

l1dbg, 100

l2, 122

L2 controller commands, 111

l2find, 123

leds, 76

link, 77

log, 77, 123

margin, 103

multisys, 124

network, 79

nmi, 80

nvramp reset, 124

partdb, 80

password, 125

pbay, 80, 125

pci, 82

ping, 125

port, 84

power, 87, 126

quit, 127

rackid, 127

reboot_l1, 90

- reboot_l2, 128
- reset, 91, 128
- router, 91
- scan, 105
- select, 92, 128
- serial, 93, 131
- shell, 131
- smp, 132
- softreset, 94
- sysname, 133
- test, 109
- uart, 110
- usb, 111
- verbose, 95
- version, 95, 133
- command targeting
 - L1 mode, 27, 32
 - L2 mode, 17
- config command, 59, 112
- console communications, 21
 - subchannels, 21
- console mode
 - from L1 mode, 28, 33
 - from L2 emulator mode, 20
- console selection, 21, 29
- cpu command, 61

D

- date command, 62, 116
- debug command, 63
- Dense router, 41
- destination command, 117
- dhcpserver command, 119
- display command, 65

E

- eeprom command, 95
- env command, 66, 119

F

- fan command, 72
- firmware upgrade
 - L1 and L2, 35
- flash command, 72

H

- help command, 73, 120
- history command, 75

I

- identifying blades, 40
- identifying bricks, 39
- individual rack unit (IRU), 41
- information, warning, and error message
 - viewing in L1 mode, 27
- information, warning, and error messages
 - viewing in L2 mode, 19, 35
- ioport command, 96
- ip command, 120
- istat command, 98

J

- junkbus command, 100

L

l1 command, 76, 120

L1 console selection, 34

L1 controller

 commands, 52

 commands, devices supported, 52

 messages, 46

 on Altix 4700 systems, 23

 online list, 51

L1 controller command

 *(asterisk character), 53

 autopower, 54

 blade, 54

 brick, 55

 config, 59

 cpu, 61

 date, 62

 debug, 63

 display, 65

 eeprom, 95

 env, 66

 fan, 72

 flash, 72

 help, 73

 history, 75

 ioport, 96

 istat, 98

 junkbus, 100

 l1, 76

 l1dbg, 100

 leds, 76

 link, 77

 log, 77

 margin, 103

 network, 79

 nmi, 80

 partdb, 80

 pbay, 80

 pci, 82

 port, 84

 power, 87

 reboot_l1, 90

 reset, 91

 router, 91

 scan, 105

 select, 92

 serial, 93

 softreset, 94

 test, 109

 uart, 110

 usb, 111

 verbose, 95

 version, 95

L1 firmware

 upgrading, 35

L1 mode

 command targeting, 27, 32

 from L1 software, 24, 31

 from L2 emulation mode, 23

 from L2 mode, 12

 powering off, 28

 powering on, 28

 resetting the brick, 28

 to console mode, 28, 33

 viewing information, warning, and error
 messages, 27, 35

 viewing system configuration, 24, 32

L1 software

 to L1 mode, 24, 31

l1dbg command, 100

l2 command, 122

L2 controller

 application on Altix 4700 system, 4

 commands, 111

 messages, 48

 on Altix 4700 systems, 13

 online list, 51

L2 controller command

 autopower, 111

- config, 112
 - date, 116
 - destination, 117
 - dhcpserver, 119
 - env, 119
 - help, 120
 - ip, 120
 - l1, 120
 - l2, 122
 - l2find, 123
 - log, 123
 - multisys, 124
 - nvrn reset, 124
 - password display, set, change or clear, 125
 - pbay, 125
 - ping, 125
 - power, 126
 - quit, 127
 - rackid, 127
 - reboot_l2, 128
 - reset, 128
 - select, 128
 - serial, 131
 - shell, 131
 - smp, 132
 - sysname, 133
 - version, 133
 - L2 emulation mode
 - to L1 mode, 23
 - viewing system configuration, 16
 - L2 emulator mode
 - powering off, 19
 - powering on, 19
 - resetting system, 19
 - setting command targeting, 17
 - to console mode, 20
 - viewing information, warning, and error messages, 19
 - L2 firmware
 - upgrading, 35
 - L2 mode
 - to L1 mode, 12
 - l2find command, 123
 - leds command, 76
 - link command, 77
 - log command, 77, 123
- M**
- margin command, 103
 - messages
 - L1 controller, 46
 - L2 controller, 48
 - multisys command, 124
- N**
- network command, 79
 - nmi command, 80
 - nvrn reset command, 124
- P**
- partdb command, 80
 - password command, 125
 - pbay command, 80, 125
 - pci command, 82
 - ping command, 125
 - port command, 84
 - power command, 87, 126
 - powering off
 - L1 mode, 28
 - L2 mode, 19
 - powering on
 - L1 mode, 28

L2 emulator mode, 19

Q

quit command, 127

R

rackid command, 127

reboot_l1 command, 90

reboot_l2 command, 128

reset command, 91, 128

resetting

 L1 mode, 28

resetting system

 L2 mode, 19

router command, 91

S

scan command, 105

select command, 92, 128

serial command, 93, 131

shell command, 131

smp command, 132

softreset command, 94

subchannels

 associated with console communications, 21

sysname command, 133

system control on Altix 4700 systems, 4

T

test command, 109

U

uart command, 110

upgrading

 L1 and L2 firmware, 35

usb command, 111

V

verbose command, 95

version command, 95, 133

viewing system configuration

 L1 mode, 24, 32

W

warning and error messages

 viewing in L1 mode, 27

warning messages

 viewing in L2 mode, 19, 35

