Sun™ Datacenter Switch 3456 Reference Manual

Sun Microsystems, Inc.
www.sun.com

Part No. 820-4734-10
August 2008, Revision A

Submit comments about this document at: `http://www.sun.com/hwdocs/feedback`

Please
Recycle

Contents

Preface

This reference manual provides higher level information about the Sun Datacenter Switch 3456 switch.

This document is written for system administrators, network administrators, and users who have advanced knowledge and experience managing an InfiniBand network. Topics include theory of operation, syslog errors, and administrative commands.

Using UNIX Commands

This document might not contain information about basic UNIX® commands and procedures such as shutting down the system, booting the system, and configuring devices. Refer to the following for this information:

- Software documentation that you received with your system
- Solaris™ Operating System documentation, which is at:

  `http://docs.sun.com`

Shell Prompts

Shell	Prompt
C shell	*machine-name*%
C shell superuser	*machine-name*#
Bourne shell and Korn shell	$
Bourne shell and Korn shell superuser	#

Typographic Conventions

Typeface*	Meaning	Examples
AaBbCc123	The names of commands, files, and directories; on-screen computer output	Edit your .login file. Use ls -a to list all files. % You have mail.
AaBbCc123	What you type, when contrasted with on-screen computer output	% **su** Password:
AaBbCc123	Book titles, new words or terms, words to be emphasized. Replace command-line variables with real names or values.	Read Chapter 6 in the *User's Guide*. These are called *class* options. You *must* be superuser to do this. To delete a file, type rm *filename*.

* The settings on your browser might differ from these settings.

Related Documentation

The documents listed as online are available at:

http://docs.sun.com/app/docs/prod/switch.3456

Application	Title	Part Number	Format	Location
Product Notes	*Sun Datacenter Switch 3456 Product Notes*	820-4727-10	PDF	Online
Unpacking	*Sun Datacenter Switch 3456 Unpacking Guide*	820-4736-10	PDF Printed	Shipping crate Online
Site Planning	*Sun Datacenter Switch 3456 Site Planning Guide*	820-4728-10	PDF	Online
Installation	*Sun Datacenter Switch 3456 Installation Guide*	820-4730-10	PDF Printed	Shipping kit Online
Administration	*Sun Datacenter Switch 3456 Administration Guide*	820-4731-10	PDF	Online
Service	*Sun Datacenter Switch 3456 Service Manual*	820-4733-10	PDF	Online
Reference	*Sun Datacenter Switch 3456 Reference Manual*	820-4734-10	PDF	Online
Regulatory	*Sun Datacenter Switch 3456 Safety and Compliance Guide*	820-4735-10	PDF	Online

Documentation, Support, and Training

Sun Function	URL
Documentation	http://www.sun.com/documentation/
Support	http://www.sun.com/support/
Training	http://www.sun.com/training/

Third-Party Web Sites

Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites

or resources. Sun is not responsible or liable for any actual or alleged damage or loss caused by or in connection with the use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. You can submit your comments by going to:

`http://www.sun.com/hwdocs/feedback`

Please include the title and part number of your document with your feedback:

Sun Datacenter Switch 3456 Reference Manual, part number 820-4734-10.

CHAPTER 1

Network Operation

This chapter explains how the Sun Datacenter Switch 3456 interacts with the InfiniBand network and components. Topics include:

Initial Fabric Startup

Sun Datacenter Switch 3456 Startup

1. Sun Datacenter Switch 3456 startup commences when standby power is enabled for the Chassis Management Controllers (CMCs). Standby power is applied when the power supplies are energized.
2. The redundant CMCs perform self-tests and determine if the other instance is present and operational. The CMCs then negotiate active and standby roles.
3. The active CMC becomes available on the management network for the fabric director, Sun N1 System Manager (N1SM), and the direct operation command-line interface.
4. The CMC checks that the required number of power supplies are present and operational in each power domain, and then activates full power mode for each power supply.
5. If the power supply population is functional, commands to the CMC enable standby power for each fabric card and line card. Standby power is provided to each card sequentially to avoid overloading the power supplies.

6. The CMC performs further discovery and checks of the fabric cards and line cards using Intelligent Platform Management interface (IPMI) communication with the Intelligent Platform Management controllers (IPMC)s in the fabric cards and line cards.

Note – Conversion from standby to full-power mode can be an automatic or explicitly controlled operation.

---

7. The CMC determines if the available power is sufficient for the number of fabric cards and line cards in each power domain, and then commands to the CMC can initiate poweron sequencing. All the fabric cards are powered on first, followed by the line cards.
8. After poweron of each fabric card, the CMC checks that a sufficient number of fans on the fabric card are fully operational. If too few fans are operational, the fabric card is shut down.
9. Completing the fabric card poweron, the CMC checks that both the card and supplied power status is okay.
10. When all fabric cards have been powered on and checked, the CMC re-evaluates the cooling capacity to ensure that line cards are only powered on if sufficient cooling capacity is available.
11. The line cards are then powered on, as directed by commands to the CMC.
12. When a fabric card or line card is instructed to go to full power, the card's onboard IPMC verifies that the local power sensors indicate okay status, and then enables power for the switch chips.
13. The IPMC performs basic card self-test functions that include being able to read sensible data from the key I^2C slave devices on the card. The switch chips also have I^2C slave devices.
14. The IPMC constantly monitors the various sensors and signals on the card, including cable and link status.
15. The IPMC enables all IB ports and links on the fabric cards, and all internal ports and links on the line cards. The iPASS connectors on the line cards are not enabled until the IPMC is instructed to do so by the CMC.
16. After power sequencing and basic tests have been completed, the IPMC reports the card as operational to the CMC.
17. When the CMC has received operational status confirmation from all cards, the CMC correlates IB link state information to verify that all relevant internal links have become operational.
18. Links that fail to become operational are reported, and might become blacklisted.

19. When the Sun Datacenter Switch 3456 internal state has become stable, the fabric director proxy (FDP) within the CMC informs the fabric director that the Sun Datacenter Switch 3456 is fully operational.
20. The CMC then instructs the IPMCs on the line cards to enable external links through the iPASS connectors on the line cards.

Standalone Server Startup

1. When standby power is available, each standalone server is initially available on the management network through its service processor for the fabric director, N1SM, and the direct operation command-line interface.
2. The standalone servers (service processors) enable full power and boot the operating system (OS) when instructed or by a default policy setting. The host OS also becomes available on the management network.
3. After the OS and the basic InfiniBand (IB) stack have been initialized, the InfiniBand Link Diagnostic Agent (LDA) components and associated FDP modules are activated.
4. The LDA immediately starts monitoring the local IB links and keeps track of basic link status. The LDA probes the remote end for connectivity info as well as to generate test traffic when the links are up.
5. The service processor and the host FDP might be contacted by the fabric director at any point in time. When a FD to FDP session has been established, the host FDP reports state change events to the fabric director. The service processor is used in poll mode so that the existence of the server can be determined. This information could otherwise come from the N1SM.

Sun Blade 6048 Modular System With IB NEMs and Blade Startup

1. Each shelf in a Sun Blade 6048 Modular System chassis is completely independent. The Chassis Management Module (CMM) in each shelf or blade service processor is available on the management network for the fabric director, N1SM, and the direct operation command-line interface when standby power is available in the chassis.
2. The InfiniBand Network Express Module (IB NEM) is powered up whenever full power is available in the shelf.

3. The FDP in the IB NEM (for the CMM) begin monitoring IB Host Channel Adapter (HCA), switch, link, and cable state on the IB NEM whenever full power has been enabled.
4. The service processors in the blade servers enable full power and boot the OS when instructed or by a default policy setting.
5. After the OS and the basic InfiniBand stack have been initialized, the InfiniBand LDA components and associated FDP modules are activated.

Note – OS boot or provisioning through IB is a special case.

---

6. The FDP within the IB NEMs keeps track of basic HCA and host driver states associated with the HCA over the I^2C bus.
7. If a blade LDA is present, then the associated FDP forwards basic link monitoring status to the IB NEM FDP through the HCA.
8. The IB NEM enables both external and internal IB links, but requires explicit instructions to enable the external links from the FDP.
9. The IB NEM FDP instructs the LDA on one of the blades to perform checking of external IB switch links.
10. All blade LDAs check external IB switch links, however at a reduced packet rate to avoid VL15 overflow.
11. The blade FDP forwards status info to the IB NEM FDP or to the fabric director directly through IP based protocols on the management (or data) network.

Fabric-Level Continuity Check

1. For each discovered and monitored fabric component, the fabric director keeps track of connected iPASS cable IDs and types.
2. For each cable with standard iPASS connectors on both ends, the fabric director can determine connectivity by looking up which fabric components have the same cable ID or serial number. Two components with the same cable ID are connected by that cable. Additionally, the standard iPASS cables enable the fabric director to keep the corresponding IB links or ports disabled until correct connectivity has been verified.
3. For splitter cables, only the 12X end has a cable ID or serial number. The fabric director always enables Sun Datacenter Switch 3456 external links associated with a split cable as soon as possible. Connectivity can only be determined by probing the Global Unit IDs (GUID)s associated with the cable end connectors.

4. During fabric construction, the operator can use the fabric director Graphical User Interface (GUI) to keep track of on-going cabling work. As a side effect, the initial discovery and checking phase could possibly last for weeks.
5. The criteria for completing the connectivity checking phase are based on a combination of:
   - Exhaustive discovery of components in the component list
   - Rules and policies for required components or component numbers
   - Explicit administrator input

Note – Any discovered fabric component must complete the basic startup and connectivity check before the fabric director has completed the discovery and checking phase.

---

6. After the criteria have been met, the fabric director instructs the subnet manager to initialize the subnet through a specific IB port. Both master and standby subnet manager instances are initiated for each IB subnet.

Fabric Initialization and Data Traffic Enabling

1. The subnet manager starts discovery and initialization based on the following configuration states input from the fabric director:
   - The fabric configuration is not according to any known physical configuration blueprint, and complete discovery and routing is required. This state is the default mode if no fabric director input or check is used.
   - The fabric configuration is according to a specific known physical configuration blueprint and the corresponding preestablished logical configuration blueprint can be initialized. There is no need for complete rediscovery or rerouting. However, the present configuration might only be a subset of the blueprint configuration.

     In this state, the subnet manager performs an optimized subnet discovery procedure to record missing components and then initializes the present components based on predefined Local IDs (LID)s and forwarding table contents.

     The subnet manager does not assume that the same GUIDs are present. Only that equivalent components are interconnected in the exact same topology and with the same port numbers.
   - The fabric configuration is exactly the same as a previously recorded configuration. In this state, no discovery is required at all.

Note – This state is highly unlikely, when subnet manager configurations are redundant and rolling subnet manager upgrades are supported. The exception is a complete restart after a power outage.

2. If one or more links or switch elements are not present or operational, then they cannot affect the contents of forwarding tables entries for LIDs associated with HCA ports. However, the DR paths the subnet manager uses to access switches, HCAs, and the LID routes to switches, must be adapted to compensate for missing switches and links.
3. After completing the subnet initialization, each HCA port has LIDs assigned for each defined rail, and the corresponding multipath routes are set up. Internet Protocol over InfiniBand (IPoIB) multicast groups are also set up for each rail. If a rail is degraded, due to missing or nonoperational components, communication through that rail might not be possible for one or more nodes.
4. The hosts can verify that the subnet manager has completed the initialization by the local port link state transitioning to LinkActive. At this time, the default host behavior is to start setting up data traffic.
5. If a Sun LDA is enabled on the host, data traffic enabling can be differed until the LDA has performed additional link tests in the LinkActive state.
6. The subnet manager permits each activated HCA port or link a grace period to become visible to subnet administrator access. If the LDA-based tests provoke link problems, the link can be disabled without mis-information communicated to peer nodes.
7. If the grace period expires and the subnet manager sees the link state as LinkActive, the link is considered stable, and the corresponding HCA port and any associated paths are visible to subnet administrator.
8. The host can enable normal data traffic with available peer nodes.

Adding Fabric Components

Adding a Fabric Card

Adding a fabric card is normally a repair action or expansion of a less than fully configured Sun Datacenter Switch 3456.

1. The administrator informs the CMC identifying the slot where the fabric card is to be installed. The CMC verifies that the slot is empty and writes a log event.
2. The CMC acknowledges the new card has been inserted by observing the corresponding presence sensors.
3. Through administrator commands, the CMC enables standby power for the new fabric card and verifies that the IPMC on the card is fully operational.
4. The CMC checks that the available power supplies in the corresponding power domain can accommodate another fabric card powering on. The administrator commands the CMC to power on the card, and waits for the IPMC to report the card as fully operational.
5. If installing the fabric card compliments the chassis cooling budget, the CMC powers on any previously disabled components and informs the fabric manager.
6. The CMC checks that all IB links connected to operational line cards have trained correctly, and that no excessive error counts exist for the fabric card ports and corresponding line card ports.
7. The CMC reports the fabric card as operational to the fabric manager.
8. When the fabric manager is told of the operational fabric card, the manager instructs one or more available LDA instances to perform SMP-based acceptance testing of the new fabric card.
9. If the card passes the acceptance tests, the fabric manager informs the subnet manager to perform activation of the new fabric components. The activation information can either be a blanko instruction, or contain the exact list of switch GUIDs that are allowed to be activated. Because the subnet manager already has received Simple Management Protocol (SMP) traps from the line card switch chips that experienced link training, the subnet manager has acknowledged the newly added fabric card.
10. Upon permission to activate the switch chips on the fabric card, the subnet manager verifies that the connectivity is in accordance with the recorded topology information. The subnet manager updates the forwarding tables for the new switches with the existing routing information. The forwarding tables in operational switches already contain correct routing information, since the subnet manager configured them with a complete logical configuration blueprint.
11. The subnet manager transitions the link state of the relevant ports to LinkActive and starts the grace period for each port.
12. The subnet manager reports events about grace period start for each activated port.

13. The fabric manager observes the grace period start events and can instruct one or more LDA instances to perform grace period testing of the relevant switch ports and links. If the testing fails, the link's state is updated.
14. When the grace period ends, the subnet manager verifies that all involved ports still are in a LinkActive state and considers the links as online.
15. The subnet manager again provides path records for paths routed through the links now marked as online.
16. The LDAs on various hosts observe that paths are now online, and verify with probe operations to remote peer LDA instances through the rail numbers.
17. The LDA instance on each host reports that the corresponding fabric rail is operational. Depending on how the LDA is integrated with the host runtime system, reporting can include the local IB stack, Upper Layer Protocols (ULP)s, and N1GE agents.

Adding a Line Card

Adding a line card is normally a repair action or expansion of a less than fully configured Sun Datacenter Switch 3456.

1. The administrator informs the CMC identifying the slot where the line card is to be installed. The CMC verifies that the slot is empty and writes a log event.
2. The CMC acknowledges the new card has been inserted by observing the corresponding presence sensors.
3. Through administrator commands, the CMC enables standby power for the new line card and verifies that the IPMC on the card is fully operational.
4. The CMC checks that the available power supplies in the corresponding power domain can accommodate another line card powering on. The aministrator commands the CMC to power on the card, and waits for the IPMC to report the card as fully operational.
5. The CMC checks that all IB links connected to operational fabric cards have trained correctly, and that no excessive error counts exists for the line card ports and corresponding fabric card ports.
6. The CMC records the current iPASS cable status for the line card.
7. All the external links on the line card remain disabled.
8. The CMC reports the line card as operational to the fabric manager.

9. When the fabric manager is told of the operational line card, the manager instructs one or more available LDA instances to perform SMP-based acceptance testing of the new line card.

10. If the card passes the acceptance tests, the fabric manager informs the subnet manager to perform activation of the new fabric components. The activation information can either be a blanko instruction, or contain the exact list of switch GUIDs that are allowed to be activated. Because the subnet manager already has received SMP traps from the fabric card switch chips that experienced link-training, it has acknowledged the newly added line card.

11. Upon permission to activate the switch chips on the line card, the subnet manager verifies that the connectivity is in accordance with the recorded topology information. The subnet manager updates the forwarding tables for the new switches with the existing routing information. The forwarding tables in operational switches already contain correct routing information for the switch chips on the new line card, since the subnet manager configured them with a complete logical configuration blueprint.

12. The subnet manager transitions the link state of the ports on connected fabric cards as well as all the internal link ports on the line card to LinkActive and starts the grace period for each port.

13. The subnet manager reports events about grace period start for each activated port.

14. The fabric manager observes the grace period start events and can instruct one or more LDA instances to perform grace period testing of the relevant switch ports and links. If the testing fails, the link's state is updated.

15. When the grace period ends, the subnet manager verifies that all involved ports still are in a LinkActive state and considers the links as online.

16. At this time, the only new routes or paths available are the trivial ones for the switch management ports on the new line card. The subnet manager can now provide path records for those trivial paths, but those paths are never involved with any data traffic.

17. Any cable connected to the line card at this time is reported to the fabric manager and handled according to cable type.

Adding a Leaf-Switch

Adding a leaf-switch can be a repair operation for an IB NEM, or part of a system expansion.

Note – Future stand-alone leaf-switches are not discussed in this document.

1. The FDP on the Sun Blade 6048 Modular System shelf CMM (if present) acknowledges the insertion of an IB NEM and reports its existence to the fabric manager.
2. The IB NEM FDP:
   - Verifies it can communicate with the FPGA on the IB NEM, and if auto power on is not enabled, instructs the FPGA to power up the NEM.
   - Verifies it can communicate with each switch instance on the IB NEM through the I^2C link.
   - Checks the state of each blade slot of the Sun Blade 6048 Modular System shelf
   - Verifies that it can communicate through I^2C with each HCA instance on the IB NEM that are not kept in reset by the corresponding blade.
3. For all operational HCA instances, the IB NEM FDP verifies that IB links between the HCA ports and the corresponding switch ports have trained correctly, and that no excessive error counts are registered.
4. Any cable connected to the IB NEM at this time is reported to the fabric manager and handled according to cable type.

Consider when one leaf-switch within the IB NEM is to be connected to a Sun Datacenter Switch 3456 based IB subnet. The other switch within the IB NEM does not have any external connectivity. Instead, the other switch provides an additional shelf local, fully connected, IB subnet for the second set of HCA ports within the IB NEM. In this situation, each Sun Blade 6048 Modular System shelf needs a redundant subnet manager instance, to operate on the blades from behind the local HCA ports. The basic logical blueprint definition for this single-switch subnet configuration is very trivial and can be fixed for all shelf instances. However, to accommodate future semi-dynamic subnet expansion, the blueprint includes a specific LID value range supplied by the fabric director, and used for the particular shelf instance.

Adding a Cable to a Leaf-Switch

Adding a cable to a leaf-switch or IB NEM can occur when installing the leaf-switch, or as a repair action for a failed cable or connector.

1. The administrator starts the operation by plugging in the cable or by asking the fabric director for instructions of what cable and connection to use.

Asking for the fabric director's assistance is requested from the fabric director GUI (FD GUI) on a laptop or PDA. Providing the cable's serial number to the FD GUI can be achieved with a bar code reader or cable ID reader.

Cables must be laid out according to the floor plan, and the fabric director must be provided a detailed physical configuration blueprint. The cable connectors are labeled with site-specific information that identifies where the cable is to be attached (for example, rack, shelf, module, and connector numbers).

2. When an iPASS cable connector is plugged into a Sun Datacenter Switch 3456 line card connector or IB NEM connector, the corresponding FDP instance acknowledges the action. The FDP instance also reports the cable type and ID as well as the card position and connector number to the fabric director.

3. If the administrator has initiated a complete cabling transaction with the fabric director, the FD GUI displays the location of each end of the cable with a visualization of the involved components, and illuminates the respective component's Locator LEDs.

4. The fabric director can react on cabling that is not in accordance with the blueprint and alert the administrator to move the offending connector. An example is noncompliance to rail numbering or NEM internal switch numbers.

5. When the cable is connected correctly at both ends, the fabric director instructs both the Sun Datacenter Switch 3456 and the IB NEM FDP to enable the external links associated with the cable.

6. The fabric director instructs an available LDA instance to perform SMP based acceptance testing of the links associated with the cable. Typically, the LDA associated with a blade in the shelf is used.

7. If the cable passes the acceptance testing, the fabric director records the cable and the associated links as okay.

8. If the subnet manager is active at this time, the fabric director instructs it to perform activation of the new fabric components. The activation information can either be a blanko instruction, or contain the exact list of switch and HCA GUIDs and associated port numbers that are permitted to be activated. Because the subnet manager already has received SMP traps from the line card switch chips that experienced link training, it has acknowledged the newly added cable.

9. Upon permission to activate the switch chips on the line card, the subnet manager verifies that the connectivity is in accordance with the recorded topology information. The subnet manager updates the forwarding tables for the switches with the existing route information. Since the connection to the leaf-switch and associated end nodes are not identical to those indicated in the logical blueprint, the subnet manager routing information and forwarding table entries for the switch chips in the Sun Datacenter Switch 3456 might not correspond to the LID values already associated with routes through the newly connected external ports.

When the subnet manager determines that there is a LID value conflict, the subnet manager updates the route information and the relevant forwarding table entries by associating the correct LID values with the relevant routes.

Note – The operation described in the previous paragraph is not a situation of rerouting, but rather renaming. The routes to reach end nodes behind the line card external ports were already determined as part of the construction process for the logical configuration blueprint. So, when the LID values associated with these routes change, there is no need to repeat the original routing algorithm, or impact the routes for any other LID values that are already operational.

---

10. The subnet manager transitions the link state of the relevant ports to LinkActive and starts the grace period for each port.
11. The subnet manager reports events about grace period start for each activated port.
12. The fabric manager observes the grace period start events and can instruct one or more LDA instances to perform grace period testing of the relevant switch and HCA ports and links. If the testing fails, the link's state is updated.
13. When the grace period ends, the subnet manager verifies that all involved ports still are in a LinkActive state and considers the links as online.
14. The subnet manager again provides path records for paths routed through the links now marked as on-line.
15. The LDAs on various hosts observe that paths are now online, and verify with probe operations to remote peer LDA instances through the rail numbers.
16. The LDA instance on each host or blade attached to the leaf-switch reports that the corresponding fabric rail is operational. Depending on how the LDA is integrated with the host run-time system, reporting can include the local IB stack, ULPs, and N1GE agents.

Adding a Standalone Server

Adding a standalone server can be a repair action or represent a fabric expansion operation.

1. The new standalone server must be added to the cluster configuration in a way that allows it to be discovered by N1SM and the fabric director. Registration includes assigning an IP address for the service processor as well as for the host of the server.

2. The new standalone server is also provisioned with a system image that reflects its role in the cluster configuration. Provisioning can be done manually through N1SM, or the system image can be defined in a portion of the ROCKS definition of the cluster.

3. After the registration and basic provisioning is completed, the subsequent steps are as described in "Standalone Server Startup" on page 3.

Adding a Cable to a Standalone Server

Adding a cable to a standalone server can take place as part of the installation of the standalone host, or as the repair action for a failed HCA card, cable, or Sun Datacenter Switch 3456 line card.

Note – Connectivity of the split end of a splitter cable cannot be determined unless both cable ends are connected to an operational IB port. The following process originates when the iPASS cable connection is detected at the Sun Datacenter Switch 3456 end.

---

1. The Sun Datacenter Switch 3456 CMC detects the new iPASS connector and reports the cable type, ID, card position, and connector number to the fabric director.

2. Since the cable is a splitter type, the fabric director then instructs the CMC to enable the ports and links associated with the connector. The links would not be enabled if doing so would violate the blueprint rules.

3. When the split end of the cable is connected to an operational HCA port on the standalone server, or when the standalone server and HCA become operational, the IB link trains and the corresponding IB link LED is lit.

4. The administrator can initiate a cabling transaction with the fabric director. The FD GUI can provide visual guidance and show status updates.

5. When the LDA on the standalone server detects the trained link on the HCA port, the LDA generates test traffic and probes the remote end for connectivity information.

6. The FDP on the standalone server reports the connectivity information (node type, GUID, and port number of the remote end of the link) to the fabric director.

7. The fabric director verifies that the connection follows the blueprint rules.

8. The fabric director then instructs the subnet manager to bring up the link to the HCA port.

9. If the subnet manager knows the GUID of the HCA port (such as a repair operation), the subnet manager assigns the old LID values. If the GUID is unknown, a value in accordance with the logical configuration blueprint is assigned.
10. After LIDs are assigned, the subnet manager updates the routing information and switch forwarding tables in the same way as for "Adding a Cable to a Leaf-Switch" on page 10.
11. The subnet manager transitions the link state of the HCA and switch ports to LinkActive and starts the grace period for each port.
12. The subnet manager reports events about grace period start for both activated ports.
13. The fabric manager observes the grace period start events. The fabric manager instructs standalone server LDA instances to perform grace period testing of the link between the local HCA ports and the Sun Datacenter Switch 3456 port. If the testing fails, the link's state is updated.
14. When the grace period ends, the subnet manager verifies that both ports still are in a LinkActive state and considers the links as online.
15. The subnet manager again provides path records associated with the HCA port.
16. The LDAs on various hosts observe that paths are now online, and verify with probe operations to remote peer LDA instances on the standalone server.
17. The LDA instance on the standalone server again reports that the corresponding host rail is operational. Depending on how the LDA is integrated with the host runtime system, reporting can include the local IB stack, ULPs, and N1GE agents.

Adding a Blade

Adding a blade can be a repair action or represent a fabric expansion operation.

1. The new blade must be added to the cluster configuration in a way that enables the blade to be discovered by N1SM and the fabric director. Registration is automatic because the Sun Blade 6048 Modular System shelf is already registered. A pool of IP addresses is available to blade service processors as well as to the host of the blades.
2. The new blade is also provisioned with a system image that reflects its role as a compute server or as an I/O blade. Provisioning can be done manually through N1SM, or the system image can be defined in a portion of the ROCKS definition of the cluster.

3. If the blade OS is provisioned through IB, the DHCP client ID for the blade must be defined and registered with the Dynamic Host Control Protocol (DHCP) server. The boot server must also have knowledge of the blade IP address.

4. Because the OS is written to the FLASH disk of the blade after Preboot eXecution Environment (PXE) boot and install, OS provisioning of the blade is based on this flashed image.

5. After registration and basic provisioning, the subsequent steps are as described in "Sun Blade 6048 Modular System With IB NEMs and Blade Startup" on page 3.

Adding a Complete Sun Datacenter Switch 3456

Adding a Sun Datacenter Switch 3456 to an operational IB subnet configuration is semidestructive if the operation requires moving cables with operational links. However, if a system is initially constructed with under-provisioned up-link bandwidth, then a Sun Datacenter Switch 3456 can be added without impacting existing communication or changing any blueprint or routing information.

Adding a second Sun Datacenter Switch 3456 is nonintrusive if a new subnet is created for an existing cluster of Sun Blade 6048 Modular Systems, where the unused HCA and switch ports in the IB NEMs are connected to the new Sun Datacenter Switch 3456.

The following bullets elaborate on the different scenarios:

- Moving from a fully connected, nonblocking single Sun Datacenter Switch 3456 configuration to a dual Sun Datacenter Switch 3456, single subnet configuration means that the physical and logical blueprint definitions are changed.

1. The existing subnet and subnet manager instances can continue to operate while the reconfiguration is taking place as long as half of the cable connectivity to the existing Sun Datacenter Switch 3456 is in compliance with the new blueprint definitions.

2. The fabric director receives the updated physical blueprint information along with the instructions that the current subnet and subnet manager instances are to remain operational.

3. With support from the FDP and LDA instances in the system, the fabric director assists in recabling so that the proper half of the existing Sun Datacenter Switch 3456 cable connectivity is moved to the new Sun Datacenter Switch 3456.

4. The current subnet manager instance acknowledges connectivity being lost and reestablished, but does not initialize the newly established connectivity.

5. After the new physical configuration has been established, the fabric director gets confirmation from the administrator and shuts down the subnet manager instances. The fabric director then starts new subnet manager instances with the new logical blueprint information.

6. In this scenario, the cluster experiences a relatively small down-time during subnet manager reinitialization. See "Fabric Initialization and Data Traffic Enabling" on page 5.

- Moving from a reduced connectivity, blocking, single Sun Datacenter Switch 3456 configuration to a full connectivity, nonblocking, dual Sun Datacenter Switch 3456 configuration means that neither the physical nor the logical blueprint definitions are changed.

  Adding connectivity to the new Sun Datacenter Switch 3456 is considered a series of add cable operations. The difference is that the subnet manager has not initialized the switches and internal links before the first Sun Datacenter Switch 3456 is connected to the subnet. This behavior is also a side effect of initially connecting a leaf-switch to the new Sun Datacenter Switch 3456.

- Adding a separate subnet from scratch means that the physical blueprint definition has expanded, and defines a new logical blueprint for the second subnet. This situation is a subset of the two previously discussed scenarios.

  The existing subnet operation is not affected. Incorrect cabling is denied by the fabric manager and subsequently, the illegal connectivity is not used by the subnet manager.

- Interconnecting a set of operational, independent, per-NEM subnets means that the physical blueprint is expanded to include two fully connected subnets. The logical blueprint definitions for the per-NEM subnets are either redefined or expanded.

  In the situation where the logical blueprints are completely redefined, the scenario is almost identical to that of adding the second subnet from scratch. The difference is that the existing per-NEM subnets are kept operational while their subnet manager instances are stopped, and a new global subnet manager is initialized. The transition changes LID values, and all existing communication contexts must be reestablished.

  The fabric director already has assigned a specific LID range for each Sun Blade 6048 Modular System shelf when the logical blueprints for the per-NEM subnets are expanded. The fabric director also ensures that the new global subnet manager uses the existing LID assignments. This way, communication between the blades in each shelf can continue uninterrupted.

CHAPTER 2

System Log Error Messages

This chapter describes error messages reported from the IB network. Topics include:

Event Message Overview

If the `SUNWsibs9p` or FME packages are installed and set up for InfiniBand network, event messages from the Sun Datacenter Switch 3456 occur on the host.

The InfiniBand (IB) management software is responsible for setting up and controlling all the IB devices that are connected together. The events reported by this software can therefore be for some other device than the Sun Datacenter Switch 3456 switch where the software is running. In InfiniBand a port on a IB device (switch or channel adapter card) is uniquely identified by a value called PortGUID. This value is often displayed in the IB related `syslog` messages. This value is also used by the `showib` command when listing the connections in the IB topology.

Many of the messages also contain IB-specific parameters. Refer to the *InfiniBand Specification*, Volume 1 for an exact explanation. All the IB related `syslog` messages contain a prefix of the type `IBSRM event` *event-number.* This part of the message is omitted from TABLE 2-1 to make it more readable.

Syslog Messages

TABLE 2-1 Syslog Messages

Message	Severity	Description
IBSRM_EVENT_PORT_UNKNOWN_MKEY at port with GUID = *port-GUID*	Warning	The IB management software is unable to control (set up and initialize) the port identified by *port-GUID* due to an unknown key setting. If this device is expected to be connected to the switch, set up the identified device with `MKEY` set to 0 and reboot that device. If it is an unexpected connection, remove the cable.
IBSRM_EVENT_PORT_GID_ONLINE, port with GUID = *port-GUID*	Information	The IB port identified by *port-GUID* has become operational. This is an expected event for all IB ports connected to the IB subnet. This port should now be visible in the topology list displayed by the `showib` CLI.
IBSRM_EVENT_PORT_GID_OFFLINE, port with GUID = *port-GUID*	Information	The IB port identified by *port-GUID* is no longer operational (it cannot participate in any data transfers). The usual cause of this event is that the cable is unplugged. This event can also happen if the cable is not properly connected or if the cable or connector is bad.
IBSRM_EVENT_PORT_STATE_CHANGED, port with GUID = *port-GUID*, `ib port =` *IB-port-number*, `connector =` *connector-number*, `new_state =` *new-state-value*, `old_state =` *old- state-value*	Information	The IB port identified by *port-GUID*, *port-number* and *connector-number* has changed the IB port state from *old-state-value* to *new-state-value*. The state values can be decoded as follows: `1` = Down `2` = Initialize `3` = Armed `4` = Active This message is often seen together with the `_ONLINE` and `_OFFLINE` messages.
IBSRM_EVENT_PORT_BAD_DIAG_CODE, port_GUID = *port-GUID*, `diag_code =` *diag-code-value*	Error	The IB port identified by *port-GUID* has reported a bad diagnostic code *diag-code-value*. The port will not be become operational. Reboot the device to see if it recovers.
IBSRM_EVENT_PORT_LINK_INTEG_REACHED, port_GUID = *port-GUID*	Error	This error event causes the IB port identified by *port-GUID* to have poor performance. Check that the cables are properly inserted. If cables are properly inserted, try rebooting the device.

TABLE 2-1 Syslog Messages *(Continued) (Continued)*

Message	Severity	Description
IBSRM_EVENT_PORT_EXCESS_BUFF_OVERRUN, port_GUID = *port-GUID*	Error	This error event causes the IB port identified by *port-GUID* to have poor performance. Check that the cables are properly inserted. If the cables are properly inserted, try rebooting the device.
IBSRM_EVENT_PORT_FLOW_CNTL_TIMER_EXPD, port_GUID = *port-GUID*	Error	This error event causes the IB port identified by *port-GUID* to have poor performance. Check that the cables are properly inserted. If the cables are properly inserted, try rebooting the device.
IBSRM_EVENT_PORT_CAP_MASK_CHANGED, port_GUID = *port-GUID*, `cap_mask =` *capability-mask*	Information	The IB port identified by *port-GUID* has had a change in IB capabilities. The new port capabilities are identified by the value of the *capability-mask*.
IBSRM_EVENT_PORT_SYS_IMG_GUID_CHANGED, port_GUID = *port-GUID*, `sys_im_GUID =` *system-image-GUID*	Information	The system image GUID of IB port identified by *port-GUID* has changed to *system-image-GUID*
IBSRM_EVENT_PORT_BAD_MKEY, src GUID = *source-port-GUID*, `dest GUID =` *destination-port-GUID*, `attr_ID =` *attribute-ID*, `method =` *method*	Warning	The IB port *destination-port-GUID* has received a package from *source-port-GUID* that has a bad `MKEY` value. *attribute-ID* and *method* describe the type of packet received. The packet has been dropped. Check to ensure that no unexpected IB devices are connected to this IB network.
IBSRM_EVENT_PORT_BAD_PKEY, src GUID = *source-port-GUID*, `dest GUID =` *destination-port-GUID*, `PKey =` *PKey-value*, `src qp =` *source-QP-number*, `dest qp =` *destination-QP-number*	Warning	The IB port *destination-port-GUID* has received a package from *source-port-GUID* that has a bad partition key *PKey-value*. The packet is dropped. *source-QP-number* and *destination-QP-number* further describe this packet. The packet has been dropped. Check to ensure that no unexpected IB devices are connected to this IB network.
IBSRM_EVENT_PORT_BAD_QKEY, src GUID = *source-port-GUID*, `dest GUID =` *destination-port-GUID*, `qkey =` *QKey-value*, `src qp =` *source-QP-number*, `dest qp =` *destination-QP-number*	Warning	The IB port *destination-port-GUID* has received a package from *source-port-GUID* that has a bad QKey value *QKey-value*.The packet is dropped. *source-QP-number* and *destination-QP-number* further describe this packet. The packet has been dropped. Check to ensure that no unexpected IB devices are connected to this IB network.
IBSRM_EVENT_PORT_PKEY_TABLE_FULL, `port_GUID =` *port-GUID*, `PKey =` *PKey-value*	Warning	The PKey *PKey-value* could not be written into the PKey table of the IB port identified by *port-GUID* because the table was already full. The port can not participate in transmissions involving this PKey. This should not happen unless there are many simultaneously running IB services in the IB subnet.

TABLE 2-1 Syslog Messages *(Continued) (Continued)*

Message	Severity	Description
IBSRM_EVENT_NODE_DISC_FAILED, port_GUID =*port-GUID*	Warning	Even after several attempts, one or more devices attached to the node identified by *port-GUID* could not be discovered. The device is not able to participate in any data transfers. Check cabling and connections. If cabling seems to be OK, reboot the device.
IBSRM_EVENT_NODE_PROG_FAILED, port_GUID =*port-GUID*, `IB port number port/connector number` *connector*	Warning	Even after several attempts, the node or port *port-GUID* could not be programmed (set up for proper IB usage). Check cabling and connections. If cabling seems to be OK, reboot the device.
IBSRM_EVENT_PORT_NO_MORE_LIDS, port_GUID = *port-GUID*	Error	The port identified by *port-GUID* could not be owned or managed because the management software ran out of LIDs it is allowed to use. This situation might happen if the Sun Datacenter Switch 3456 switch is part of an extremely large IB network.
IBSRM_EVENT_TOO_SMALL_LFT, switch GUID *switch-GUID*, `max lids supported` *max-supported-lids*, `max lid in use` *max-lid*	Error	The switch identified by *switch-GUID* cannot be programmed with wanted LID value *max-lid* because this LID is beyond the size of the switch forward table *max-supported-lids*. Any packets addressed for *max-lid* fail to transfer through the indicated IB switch.This situation might happen if the switch is part of an extremely large IB network.
IBSRM_EVENT_PORT_DUPLICATE_GUID, `duplicate GUID =` *GUID-value*, `original peer switch GUID` *switch-GUID-1*, `duplicate peer switch GUID` *switch-GUID-2*, `original peer IB port` *port1*, `duplicate peer IB port` *port2*	Error	Duplicate ports in the IB topology have the same GUID value *GUID-value*. One instance is connected to *switch-GUID-1 port1* while the other is connected to *switch-GUID-2 port2*. The IB subnet performs badly since unique GUID values are central for correct IB management. The only way to correct this situation is to exchange one of the indicated devices with another device having a proper GUID.
IBSRM_EVENT_MCG_CREATED, mclid = *mc-lid-value*	Information	The multicast group identified by multicast LID *mc-LID-value* has been created.
IBSRM_EVENT_MCG_DELETED, mclid = *mc-lid-value*	Information	The multicast group identified by multicast LID *mc-LID-value* has been deleted.
IBSRM_EVENT_SM_BECAME_MASTER, sm_GUID = *GUID-value*	Information	The master subnet manager in the IB subnet is running behind the port identified by *GUID-value*.
IBSRM_EVENT_MISC_EMPTY_SUBNET, local GUID *GUID-value*	Information	The IB subnet that the local switch *GUID-value* is part of is currently empty. (No hosts are connected or yet done booting.)

TABLE 2-1 Syslog Messages *(Continued) (Continued)*

Message	Severity	Description
IBSRM_EVENT_MISC_BAILOUT, reason = *reason*, `file` *source-file*, `line` *source-line*	Error	The IB management software running on the Sun Datacenter Switch 3456 switch is bailing out. The bailout point in the source is identified by *source-file* and *source-line*. The reason for the bailout is indicated by *reason*, where 1 is lack of memory, 2 is package transfer errors, 3 is other. The IB switch tries to automatically restart the IB management software when this event occurs. If the event recurs after the restart, contact service.
`IBSRM will not write to journal`	Information	This message is expected on each IB management start or restart on the Sun Datacenter Switch 3456 switch and does not indicate any error. This message is an indication that the switch does not support stable storage to save IB management policy and configuration information.
IBSRM_EVENT_MISC_PORT_ADD_DDP_FAILED, port_GUID = *port-GUID*	Warning	An attempt to auto-add the port *port-GUID* to the default data partition (DDP) failed. This causes failures in data transfers to and from this port, which uses the default data partition.
IBSRM_EVENT_MISC_PORT_ADD_DDP_DONE, port_GUID = *port-GUID*	Information	The port *port-GUID* has been auto-added to the default data partition. This port had previously failed to be added to the DDP because of a full PKey table. However, the removal of the port from some other partition has freed up an entry in the PKey table which allowed the auto-add to the DDP.

For more information about the *InfiniBand Specification* go to:

`http://www.infinibandta.org`

You must register before you can download specifications.

CHAPTER 3

Switch-Specific Commands

The Sun Datacenter Switch 3456 has a collection of switch-specific commands that can be used to administrate and monitor the switch. These commands compliment the CLIA commands, are much simpler to type, and return succinct user-friendly output.

Topics in this chapter include:

Command Summary

The switch-specific commands are a simplified way to direct the Pigeon Point Shelf Manager, internal to the Sun Datacenter Switch 3456, to perform more complex tasks. Only the `root` user of the shelf manager can run the switch-specific commands. The format of the switch-specific commands is as follows:

`#` *command* `[`*arguments*`][`*arguments*`]...`

TABLE 3-1 lists the switch-specific commands alphabetically, with cross-references to information about the commands.

TABLE 3-1 Switch-Specific Commands

Command Syntax	Cross-Reference
`activate lc|fc` *slot*	"`activate`" on page 25
`checkfans`	"`checkfans`" on page 34
`checklinks`	"`checklinks`" on page 35

TABLE 3-1 Switch-Specific Commands *(Continued)*

Command Syntax	Cross-Reference
`checkpowers lc|fc` *slot*	"checkpowers" on page 36
`checkpwrfault`	"checkpwrfault" on page 37
`checkswitches`	"checkswitches" on page 37
`checkvoltages`	"checkvoltages" on page 38
`deactivate lc|fc` *slot*	"deactivate" on page 26
`disableboard lc|fc` *slot*	"disableboard" on page 26
`disableipmb lc|fc` *slot bus*	"disableipmb" on page 27
`disablepsu` *slot*	"disablepsu" on page 28
`disablestby lc|fc` *slot*	"disablestby" on page 28
`disableswitchport lc|fc` *slot switch-chip port*	"disableswitchport" on page 29
`enableboard lc|fc` *slot*	"enableboard" on page 29
`enablehotinsert lc|fc` *slot*	"enablehotinsert" on page 30
`enableipmb lc|fc` *slot bus*	"enableipmb" on page 30
`enablepsu` *slot*	"enablepsu" on page 31
`enablestby lc|fc` *slot*	"enablestby" on page 31
`enableswitchport lc|fc` *slot switch-chip port*	"enableswitchport" on page 32
`getbaseguid lc|fc` *slot*	"getbaseguid" on page 39
`getfan` *slot*	"getfan" on page 39
`getfwversion lc|fc` *slot*	"getfwversion" on page 40
`i2ctest`	"i2ctest" on page 40
`i3prog` *filename* `lc|fc` *slot switch-chip*	"i3prog" on page 32
`mcmversion`	"mcmversion" on page 41
`psustatus` *slot*	"psustatus" on page 41
`resetswitch lc|fc` *slot switch-chip state*	"resetswitch" on page 33
`showlogs`	"showlogs" on page 42
`showpresent`	"showpresent" on page 43
`showtemps`	"showtemps" on page 43
`showvoltages`	"showvoltages" on page 44

Control Commands

This section describes commands that control or configure the Sun Datacenter Switch 3456:

activate

Syntax

```
activate lc|fc slot
```

where *slot* is the number of the line card (`0 - 23`) or fabric card (`0 - 17`).

Purpose

This command brings a line card or fabric card from standby state to a fully powered on state. Always activate all fabric cards, before activating line cards.

Note – Activating a fabric card or line card can take up to 3 minutes for all of the I3 switch chips to boot. During the I3 switch chip boot process, the card might be identified as being in an active (M4) state.

Example

Activate line card 20:

```
# activate lc 20
Activating LC 20
Pigeon Point Shelf Manager Command Line Interpreter
    Command issued via IPMB, status = 213 (0xd5)
#
```

deactivate

Syntax

`deactivate lc|fc` *slot*

where *slot* is the number of the line card (`0 - 23`) or fabric card (`0 - 17`).

Purpose

This command brings a line card or fabric card from a fully powered on state to a standby state. In the standby state, the IPMB bus is still active. You must deactivate a line card or fabric card if you want to run the `checkpowers` command for that component.

Example

Deactivate line card 20

```
# deactivate lc 20
Deactivating LC 20
Pigeon Point Shelf Manager Command Line Interpreter
    Command issued via IPMB, status = 0 (0x0)
    Command executed successfully
#
```

disableboard

Syntax

`disableboard lc|fc` *slot*

where *slot* is the number of the line card (`0 - 23`) or fabric card (`0 - 17`).

Purpose

This command brings a line card or fabric card from a standby state to a fully powered off state. You can disable a component from a fully powered state. If you do so, you are asked if you want to do so. To avoid this message, first deactivate the component, and then disable it.

Example

Disable line card 20, which is still active:

```
# disableboard lc 20
lc 20 is active do you want to continue removing this board (y/n)? n
#
```

Disable line card 20, which has been deactivated:

```
# disableboard lc 20
#
```

disableipmb

Syntax

`disableipmb lc|fc` *slot bus*

where:

- *slot* is the number of the line card (`0 - 23`) or fabric card (`0 - 17`).
- *bus* is either `a` or `b`.

Purpose

This command disables the IPMB bus and controller for a line card or fabric card.

Caution – Disabling the IPMB bus for an active line card or fabric card effectively deactivates the card.

Example

Disable line card 20 IPMB bus A:

```
# disableipmb lc 20 a
#
```

disablepsu

Syntax

`disablepsu` *slot*

where *slot* is the number of the power supply (`0 - 15`).

Purpose

This command brings a power supply from a fully powered on state to a standby state.

Example

Bring power supply 0 to a standby state:

```
# disablepsu 0
Using psu i2c addr 0x5d
PSU 0, 12 V is off
#
```

disablestby

Syntax

`disablestby lc|fc` *slot*

where *slot* is the number of the line card (`0 - 23`) or fabric card (`0 - 17`).

Purpose

This command disables the standby power on a line card or fabric card.

Example

Disable the standby power for line card 20:

```
# disablestby lc 20
#
```

disableswitchport

Syntax

`disableswitchport lc|fc` *slot switch-chip port*

where:

- *slot* is number of the line card (`0 - 23`) or fabric card (`0 - 17`).
- *switch-chip* is the number of the chip on the line card (`0 - 23`) or fabric card (`0 - 7`).
- *port* is the number of the port (`1 - 24`).

Purpose

This command disables a port on an I3 switch chip for a specific line card or fabric card.

Example

Disable port 6 on I3 switch chip 1 of fabric card 14:

```
# disableswitchport fc 14 1 6
```

enableboard

Syntax

`enableboard lc|fc` *slot*

where *slot* is the number of the line card (`0 - 23`) or fabric card (`0 - 17`).

Purpose

This command brings a line card or fabric card from a fully powered off state to a standby state. The IPMB bus becomes active.

Example

Enable line card 20:

```
# enableboard lc 20
lc 20 is now enabled
#
```

enablehotinsert

Syntax

`enablehotinsert lc|fc` *slot*

where *slot* is the number of the line card (`0 - 23`) or fabric card (`0 - 17`).

Purpose

This command readies the IPMB bus and standby power for a line card or fabric card slot before insertion, so that bus and power are automatically enabled upon line card or fabric card insertion into that slot. This command is similar to the `enableboard` command or the `enableipmb` and `enablestby` commands, but is executed before the card is inserted.

Example

Ready line card slot 20 for hot insertion:

```
# enablehotinsert lc 20
lc 20 is now enabled
#
```

enableipmb

Syntax

`enableipmb lc|fc` *slot bus*

where:

- *slot* is the number of the line card (`0 - 23`) or fabric card (`0 - 17`)
- *bus* is either `a` or `b`

Purpose

This command enables the IPMB bus and controller for a line card or fabric card.

Example

Enable line card 20 IPMB bus A:

```
# enableipmb lc 20 a
#
```

enablepsu

Syntax

`enablepsu` *slot*

where *slot* is the number of the power supply (`0 - 15`).

Purpose

This command brings a power supply from a standby state to a fully powered on state.

Example

Bring power supply 0 to a fully powered on state:

```
# enablepsu 0
Using psu i2c addr 0x5d
Turning on 12V ...
PSU 0, 12 V on
#
```

enablestby

Syntax

`enablestby lc|fc` *slot*

where *slot* is the number of the line card (`0 - 23`) or fabric card (`0 - 17`).

Purpose

This command enables the standby power on a line card or fabric card.

Example

Enable the standby power for line card 20:

```
# enablestby lc 20
#
```

enableswitchport

Syntax

`enableswitchport lc|fc` *slot switch-chip port*

where:

- *slot* is number of the line card (`0 - 23`) or fabric card (`0 - 17`).
- *switch-chip* is the number of the chip on the line card (`0 - 23`) or fabric card (`0 - 7`).
- *port* is the number of the port (`1 - 24`).

Purpose

This command enables a port on an I3 switch chip for a specific line card or fabric card.

Example

Enable port 6 on I3 switch chip 1 of fabric card 14:

```
# enableswitchport fc 14 1 6
```

i3prog

Syntax

`i3prog` *filename* `lc|fc` *slot switch-chip*

where:

- *filename* is the firmware filename.
- *slot* is number of the line card (`0 - 23`) or fabric card (`0 - 17`).
- *switch-chip* is the number of the chip on the line card (`0 - 23`) or fabric card (`0 - 7`).

Purpose

This command programs I3 switch chips with new firmware. This command can also program the GUID. Two binaries are sent by FTP to the CMC's file system. The firmware filenames have this format:

`m24_i3fw_`*identifier*`_v`*version*`_52.bin`

`m24_i3fw_`*identifier*`_v`*version*`_53.bin`

When the I3 switch chip is programmed, the `_52.bin` and `_53.bin` extensions are removed. The `i3prog` command looks for the extensions automatically.

Note – When you upgrade the firmware of one I3 switch chip, all switch chips should have the same firmware.

Example

Program I3 switch chip 1 on line card 13 with `m24_i3fw_101007_v1.0.4_52.bin` and `m24_i3fw_101007_v1.0.4_53.bin`:

```
# i3prog m24_i3fw_101007_v1.0.4 lc 13 1
```

resetswitch

Syntax

`resetswitch lc|fc` *slot switch-chip state*

where:

- *slot* is number of the line card (`0 - 23`) or fabric card (`0 - 17`).
- *switch-chip* is the number of the chip on the line card (`0 - 23`) or fabric card (`0 - 7`). Using a value of `255` affects all switch chips on the card.
- *state* is `0` to reset once, and `1` to hold in reset.

Purpose

This command resets an I3 switch chip on a line card or fabric card. Reset can be one time, or reset can be held indefinitely until the switch is reset again.

Example

Reset I3 switch chip 17 or line card 13 once:

```
# resetswitch lc 13 17 0
```

Monitoring Commands

This section describes commands that enable you to monitor the status or condition of components within the Sun Datacenter Switch 3456:

checkfans

Syntax

```
checkfans
```

Purpose

This command displays fan status for fabric cards 0 through 17. Each fabric card has eight fans, 0 through 7. The output contains fan speeds or a warning message of fan failure.

Example

```
# checkfans
Checking M24 fans...
FC 0 Fan 0 RPM = 24356.000000
Warning : FC 0 Fan 1 stopped
FC 0 Fan 2 RPM = 24070.000000
FC 0 Fan 3 RPM = 24356.000000
FC 0 Fan 4 RPM = 24070.000000
FC 0 Fan 5 RPM = 24356.000000
FC 0 Fan 6 RPM = 24356.000000
FC 0 Fan 7 RPM = 24356.000000
```

Note – The output seen in the example is just a portion of the full output.

checklinks

Syntax

```
checklinks [-c][-e]
```

Purpose

This command verifies that links are up for active line cards and fabric cards. Output is either a simple return of `OK`, or a comprehensive list of ports and respective I3 switch chips on the component that are down. The `-c` option will include ports to line card connectors, stating which connector (for example 3A) on the respective line card is down. The `-e` option attempts to enable links that are down. When this option is used, re-run the `checklinks` command to verify that links have been enabled.

Example

A favorable `checklinks` output:

```
# checklinks -c
LC 0 Active, checking links.....OK
LC 1 Active, checking links.....OK
LC 2 Active, checking links.....OK
LC 3 Active, checking links.....OK
LC 4 Active, checking links.....OK
LC 5 Active, checking links.....OK
```

A `checklinks` output where cable ports for several I3 switch chips on line card 13 are down:

```
LC 13 Active, checking links.....
Port 16 on I3 01 LC 13 is down          (Cable 3 A )
Port 17 on I3 01 LC 13 is down          (Cable 3 A )
Port 18 on I3 01 LC 13 is down          (Cable 3 A )
Port 22 on I3 03 LC 13 is down          (Cable 6 A )
Port 23 on I3 03 LC 13 is down          (Cable 6 A )
Port 24 on I3 03 LC 13 is down          (Cable 6 A )
```

Note – The output seen in the examples are just a portion of the full output.

checkpowers

Syntax

`checkpowers lc|fc` *slot*

where *slot* is the number of the line card (`0 - 23`) or fabric card (`0 - 17`).

Purpose

This command performs a pass-fail test for the internal power levels of a line card or fabric card.

Note – The component must be deactivated before it is checked.

Example

Check the internal power for line card 20:

```
# checkpowers lc 20
Power check for LC 20 PASSED
#
```

Note – The output seen in the example is just a portion of the full output.

checkpwrfault

Syntax

```
checkpwrfault
```

Purpose

This command verifies internal power conditions are nominal for line cards and fabric cards. Output is a simplified `OK`.

Example

```
# checkpwrfault
Checking power fault in M24...
FC 0 Power fault sensor = 0x000000     OK
FC 1 Power fault sensor = 0x000000     OK
FC 2 Power fault sensor = 0x000000     OK
FC 3 Power fault sensor = 0x000000     OK
FC 4 Power fault sensor = 0x000000     OK
```

Note – The output seen in the example is just a portion of the full output.

checkswitches

Syntax

```
checkswitches
```

Purpose

This command verifies switch boot status and the presence of fatal errors. Output is a simplified `OK`. Should an I3 switch chip fail, the output indicates so.

Example

A favorable `checkswitches` output:

```
# checkswitches
Checking booted switches in M24...
FC 0 Active, checking switches ....OK
FC 1 Active, checking switches ....OK
FC 2 Active, checking switches ....OK
```

```
FC 3 Active, checking switches ....OK
FC 4 Active, checking switches ....OK
FC 5 Active, checking switches ....OK
```

A `checkswitches` output where I3 switch chips 17 and 21 on line card 13 did not boot properly:

```
LC 12 Active, checking switches ....OK
LC 13 Active, checking switches ....
I3 no 17 on LC 13 did not boot correct. Boot syndrome = 0x00000000
I3 no 21 on LC 13 did not boot correct. Boot syndrome = 0x00000000
LC 14 Active, checking switches ....OK
```

Note – The output seen in the examples are just a portion of the full output.

checkvoltages

Syntax

`checkvoltages`

Purpose

This command verifies that voltages are at the nominal values for line cards and fabric cards. Output is a simplified `OK`.

Example

```
# checkvoltages
Reading M24 voltages...
Checking FC 0 ...
FC 0 OK
Checking FC 1 ...
FC 1 OK
Checking FC 2 ...
FC 2 OK
```

Note – The output seen in the example is just a portion of the full output.

getbaseguid

Syntax

`getbaseguid lc|fc` *slot*

where *slot* is the number of the line card (`0 - 23`) or fabric card (`0 - 17`).

Purpose

This command displays the base GUID for a line card or fabric card. The GUID is a 64-bit value.

Example

Display the base GUID for fabric card 9:

```
# getbaseguid fc 9
Base GUID = 0x00144f0000a39316
#
```

getfan

Syntax

`getfan` *slot*

where *slot* is the number of the fabric card (`0 - 17`).

Purpose

This command verifies that the fan speeds for a specific fabric card are nominal. A positive verification is the return of a value of 176.

Example

Verify the fans on fabric card 14:

```
# getfan 14
IPMB 0xce PWM = 176
#
```

getfwversion

Syntax

```
getfwversion lc|fc slot
```

where *slot* is the number of the line card (`0 - 23`) or fabric card (`0 - 17`).

Purpose

This command displays firmware information for line cards and fabric cards, including I3 switch chip firmware versions. Output identifies firmware type and version number.

Example

Display the firmware versions for fabric card 9:

```
# getfwversion fc 9
FW versions for FC 9
 H8 version          : 0.12
 FPGA version        : 43
 I3 FW image version : 1.0.4
 I3 FW build ID      : 0x311f
#
```

i2ctest

Syntax

```
i2ctest
```

Purpose

This command conducts a pass-fail test of the I^2C devices on the active CMC. Output identifies device, bus, address, and result of test.

Example

Perform the I^2C test:

```
# i2ctest
CMC I2C access test started
I2C test of ADM1026 at bus 0 addr 2e -PASSED
I2C test of PCA9506 at bus 0 addr 20 -PASSED
```

```
I2C test of PCA9506 at bus 0 addr 21 -PASSED
I2C test of PCA9506 at bus 0 addr 22 -PASSED
I2C test of LM75 at bus 0 addr 48 -PASSED
I2C test of PCA9545 at bus 0 addr 70 -PASSED
I2C test of PCA9506 at bus 4 addr 21 -PASSED
I2C test of PCA9506 at bus 4 addr 22 -PASSED
I2C test of PCA9506 at bus 4 addr 24 -PASSED
I2C test of PCA9548 at bus 4 addr 71 -PASSED
I2C test of PCA9548 at bus 4 addr 72 -PASSED
I2C test of PCA9555 at bus 4 addr 20 -PASSED
CMC I2C access test PASSED
#
```

mcmversion

Syntax

`mcmversion`

Purpose

This command displays the version of the Magnum Chassis Manager (MCM).

Example

Display the version of the MCM:

```
# mcmversion

MCM version 1.0.3
Build time: May 19 2008 14:22:00
#
```

psustatus

Syntax

`psustatus` *slot*

where *slot* is the number of the power supply (`0 - 15`).

Purpose

This command displays the on-off status for a power supply. If a power supply has been taken to a standby state, it is reported as `off`.

Example

Display the status of power supply 0:

```
# psustatus 0
Using psu i2c addr 0x5d
PSU 0, 12 V on
#
```

showlogs

Syntax

```
showlogs
```

Purpose

This command displays a log of switch events, including timestamp, location, sensor, and type of event.

Example

```
# showlogs
Mar 13 03:10:04 2008; from:(FC 14); sensor:(FAN_PRESENT); event:asserted: 0x05
Mar 13 03:10:04 2008; from:(FC 14); sensor:(FAN_PRESENT); event:asserted: 0x06
Mar 13 03:10:04 2008; from:(FC 14); sensor:(FAN_PRESENT); event:asserted: 0x07
Mar 13 03:10:04 2008; from:(FC 14); sensor:(I3_BOOTED); event:asserted: 0x00
Mar 13 03:10:04 2008; from:(FC 14); sensor:(I3_BOOTED); event:asserted: 0x01
Mar 13 03:10:04 2008; from:(FC 14); sensor:(I3_BOOTED); event:asserted: 0x02
```

Note – The output seen in the example is just a portion of the full output.

showpresent

Syntax

```
showpresent
```

Purpose

This command displays physically installed power supplies, line cards, and fabric cards. If a line card or fabric card is present, its activity is also displayed.

Example

A favorable `showpresent` output:

```
# showpresent
PSU 00 present, state unknown
PSU 01 present, state unknown
PSU 02 present, state unknown
PSU 03 present, state unknown
PSU 04 present, state unknown
PSU 05 present, state unknown
```

A `showpresent` output where fabric card 16 is not active:

```
FC 14 present and active (state = M4)
FC 15 present and active (state = M4)
FC 16 present but not active
FC 17 present and active (state = M4)
LC 00 present and active (state = M4)
LC 01 present and active (state = M4)
```

Note – The output seen in the examples are just a portion of the full output.

showtemps

Syntax

```
showtemps
```

Purpose

This command displays internal temperatures for all line cards and fabric cards.

Example

```
# showtemps
Temperature on FC 00, LM75 min = 26.00 ºC,  LM75 max = 31.00 ºC, at adm1026 =
27.00 ºC
Temperature on FC 01, LM75 min = 23.00 ºC,  LM75 max = 31.00 ºC, at adm1026 =
27.00 ºC
Temperature on FC 02, LM75 min = 26.00 ºC,  LM75 max = 25.00 ºC, at adm1026 =
30.00 ºC
Temperature on FC 03, LM75 min = 26.00 ºC,  LM75 max = 31.00 ºC, at adm1026 =
27.00 ºC
Temperature on FC 04, LM75 min = 23.00 ºC,  LM75 max = 25.00 ºC, at adm1026 =
30.00 ºC
Temperature on FC 05, LM75 min = 26.00 ºC,  LM75 max = 31.00 ºC, at adm1026 =
27.00 ºC
```

Note – The output seen in the example is just a portion of the full output.

showvoltages

Syntax

```
showvoltages
```

Purpose

This command displays the internal voltages for the line cards and fabric cards. The left column displays the expected voltage, the right column displays the measured voltage.

Example

```
# showvoltages
Reading M24 voltages...
FC 0 readings
 12V       = 12.16
 3.3V      = 3.27
 3.3V STBY = 3.34
 1.8V      = 1.75
 1.6V_0    = 1.57
 1.6V_1    = 1.60
 1.2V_0    = 1.19
 1.2V_1    = 1.20
```

Note – The output seen in the example is just a portion of the full output.

CHAPTER 4

Command-Line Interface Agent Commands

This appendix provides a reference of command-line interface agent (CLIA) commands that can be used when administrating the Sun Datacenter Switch 3456. Topics include:

Command Summary

The `clia` command is an intermediary interface to route directives to the Pigeon Point Shelf Manager, internal to the Sun Datacenter Switch 3456. The shelf manager is similar to the system controller used in Sun servers. Only the `root` user of the shelf manager can run the `clia` command and its subcommands. The format of the CLIA commands is as follows:

```
# clia subcommand [arguments][arguments]...
```

TABLE 4-1 lists the subcommands and arguments alphabetically, with cross-references to information about the commands.

TABLE 4-1 Command-Line Interface Agent Subcommands

Subcommand Syntax	Cross-reference
`activate` *IPMB-address FRU-ID*	"`activate`" on page 57
`alarm [clear|minor|major|critical]`	"`alarm` " on page 82
`deactivate` *IPMB-address FRU-ID*	"`deactivate`" on page 57
`exit`	"`exit` | `quit`" on page 82
`quit`	"`exit` | `quit`" on page 82
`fru [-v] [`*addr* `[id=`*FRU-ID* | `type=`*site-type*`]] | [type=`*site-type* `[/`*site-number*`]]`	"`fru`" on page 58
`frucontrol` *IPMB-address FRU-ID option*	"`frucontrol`" on page 59
`fruinfo [-v] [-x]` *addr FRU-ID*	"`fruinfo`" on page 59
`getfruledstate [-v] [`*IPMB-addr* `state [`*FRU-ID* `[`*LED-ID* | `ALL]]]`	"`getfruledstate`" on page 60
`getipmbstate` *IPMB-address* `[`*link*`]` (in radial IPMB-0 environment)	"`getipmbstate`" on page 68
`getlanconfig` *channel* `[`*parameter-name* `[`*additional-parameters*`]]`	"`getlanconfig`" on page 76
`help [`*command* `[`*subcommand*`]]`	"`help`" on page 82
`ipmc [-v] [`*IPMB-address*`]`	"`ipmc`" on page 69
`localaddress`	"`localaddress`" on page 70
`poll`	"`poll`" on page 71
`sel [-v] [`*IPMB-address* `[`*record-count* `[`*starting-entry*`]]]`	"`sel`" on page 71
`sensordata [`*IPMB-address* `[`*sensor-name* | `[`*lun*`:]`*sensor-number*`]]`	"`sensordata`" on page 83
`session`	"`session`" on page 72
`setextracted` *IPMB-address FRU-ID*	"`setextracted`" on page 61
`setfruledstate` *IPMB-address FRU-ID LedId*`|ALL` *LedOp* `[`*LedColor*`]` *LedOp* = `ON | OFF | LOCAL | BLINK` *onTime offTime* `| TEST` *onTime* *LedColor* = `BLUE | RED | GREEN | AMBER | ORANGE | WHITE | NONE |` *number*	"`setfruledstate`" on page 62
`setipmbstate` *IPMB-address* `A|B [`*link*`] 1|0` (in radial IPMB-0 environment)	"`setipmbstate`" on page 73
`setlanconfig` *channel parameter-name additional-parameters*	"`setlanconfig`" on page 79
`setlocked` *IPMB-address FRU-ID* `0 | 1`	"`setlocked`" on page 63

TABLE 4-1 Command-Line Interface Agent Subcommands *(Continued)*

Subcommand Syntax	Cross-reference
`shelf` *subcommand* `shelf [cooling_state | fans_state | address_table | power_distribution | power_management | pci_connectivity | ha_connectivity | h110_connectivity | point-to-point_connectivity]`	"shelf" on page 64
`shelfaddress [`*up-to-20-characters-of-the-shelf-address*`]`	"shelfaddress" on page 74
`shmstatus`	"shmstatus" on page 75
`showunhealthy`	"showunhealthy" on page 67
`switchover`	"switchover" on page 75
`terminate`	"terminate" on page 76
`user [`*subcommand*`]`	"user" on page 50
`user [-v] [`*userid*`]`	"user [-v]" on page 50
`user add` *userid user-name channel-access-flags privilege-level password*	"user add" on page 51
`user delete userid`	"user delete" on page 52
`user enable` *userid* `1 | 0`	"user enable" on page 52
`user name` *userid user-name*	"user name" on page 53
`user passwd` *userid password*	"user passwd" on page 54
`user channel` *userid channel-number channel-access-flags privilege-level*	"user channel" on page 55
`version`	"version" on page 84

User Management Commands

This section describes commands for user management:

user

Syntax

`user [`*subcommand*`]`

The following subcommands are supported:

- `add`
- `delete`
- `enable`
- `name`
- `passwd`
- `channel`

Purpose

The `user` command shows information about the RMCP user accounts on the shelf manager. This command also provides a simple way to add, delete, and modify the user accounts.

The following sections describe the syntax of the `user` command for different applications of this command.

user [-v]

Syntax

`user [-v] [`*userid*`]`

Purpose

This command shows information about users. When you type the command with a `-v` option, it also shows information about disabled users. If you specify the optional user ID (*userid*), only information about the user with that ID is shown.

The following items of information are shown:

- User ID
- User name
- Channel access information for each IPMI channel – The maximum privilege level of that user on that channel and channel access flags.

  If the channel access information is the same for several channels, the output is coalesced and the range of channels is shown.

Example

```
# clia user -v
Pigeon Point Shelf Manager Command Line Interpreter
    1: ""
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
    2: "openhpi"
        Channels 0-15 Privilege level: "OEM Proprietary"
            Flags: "IPMI Messaging"
#
```

user add

Syntax

`user add` *userid user-name channel-access-flags privilege-level password*

Purpose

This command adds a new user to the system. Command parameters have the following meaning:

- *userid* – A valid user ID.
- *user-name* – The user name (it is truncated to 16 characters without any notice).
- *channel-access-flag* – The first byte of the SetUserInfo commands (only bits 4,5, and 6 are meaningful).
  - Bit 6 – IPMI messaging enabled.
  - Bit 5 – Link authentication enabled.
  - Bit 4 – Restricted to callback.
- *privilege-level* – The user privilege level.
- *password* – The user password (it is truncated to 16 characters without any notice).

Example

Add user 3 with the name `test_user`, administrator privilege level, and password `passwurd`.

```
# clia user add 3 "test_user" 0x40 4 "passwurd"
Pigeon Point Shelf Manager Command Line Interpreter
    User 3 added successfuly
# clia user
Pigeon Point Shelf Manager Command Line Interpreter
```

```
    1: ""
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
    2: "openhpi"
        Channels 0-15 Privilege level: "OEM Proprietary"
            Flags: "IPMI Messaging"
    3: "test_user"
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
#
```

user delete

Syntax

`user delete` *userid*

Purpose

This command deletes the user specified by the *userid*.

Example

Delete the user with userid 3.

```
# clia user delete 3
Pigeon Point Shelf Manager Command Line Interpreter
    User 3 deleted successfuly
# clia user
Pigeon Point Shelf Manager Command Line Interpreter
    1: ""
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
    2: "openhpi"
        Channels 0-15 Privilege level: "OEM Proprietary"
            Flags: "IPMI Messaging"
#
```

user enable

Syntax

`user enable` *userid* `1 | 0`

Purpose

This command enables or disables a user by *userid*. The last command parameter specifies the requested action, as follows:

- `0` – Disables the specified user.
- nonzero – Enables the specified user.

Example

Disable and enable user with userid 3.

```
# clia user enable 3 0
Pigeon Point Shelf Manager Command Line Interpreter
    User 3 disabled successfuly
# clia user
Pigeon Point Shelf Manager Command Line Interpreter
    1: ""
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
    2: "openhpi"
        Channels 0-15 Privilege level: "OEM Proprietary"
            Flags: "IPMI Messaging"
# clia user enable 3 1
Pigeon Point Shelf Manager Command Line Interpreter
    User 3 enabled successfuly
# clia user
Pigeon Point Shelf Manager Command Line Interpreter
    1: ""
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
    2: "openhpi"
        Channels 0-15 Privilege level: "OEM Proprietary"
            Flags: "IPMI Messaging"
    3: "test_user"
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
#
```

user name

Syntax

`user name` *userid user-name*

Purpose

This command is used to modify the user name for the specified user. The command parameters have the following meanings:

- *userid* – The valid user ID.
- *user-name* – The user name (which is truncated to 16 characters without any notice).

Example

Change the name of user 3 to `administrator`.

```
# clia user
Pigeon Point Shelf Manager Command Line Interpreter
    1: ""
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
    2: "openhpi"
        Channels 0-15 Privilege level: "OEM Proprietary"
            Flags: "IPMI Messaging"
    3: "test_user"
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
# clia user name 3 administrator
Pigeon Point Shelf Manager Command Line Interpreter
    User 3, name changed successfuly
# clia user
Pigeon Point Shelf Manager Command Line Interpreter
    1: ""
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
    2: "openhpi"
        Channels 0-15 Privilege level: "OEM Proprietary"
            Flags: "IPMI Messaging"
    3: "administrator"
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
#
```

user passwd

Syntax

`user passwd` *userid password*

Purpose

This command is used to modify the password for the specified user. The command parameters have the following meanings:

- *userid* – The valid user ID.
- *password* – The user password (which is truncated to 16 characters without any notice).

Example

Change the password of user ID 3 to `root`.

```
# clia user passwd 3 root
Pigeon Point Shelf Manager Command Line Interpreter
    User 3, password changed successfuly
#
```

user channel

Syntax

`user channel` *userid channel-number channel-access-flags privilege-level*

Purpose

This command is used to modify the channel access setting for a specified channel and user. (The user is specified by the user ID.) The command parameters have the following meanings:

- *userid* – The valid user ID.
- *channel-number* – The channel number.
- *channel-access-flags* – The first byte of the SetUserInfo commands (only bits 4, 5, and 6 are meaningful).
  - Bit 6 – IPMI messaging enabled.
  - Bit 5 – Link authentication enabled.
  - Bit 4 – Restricted to callback.
- *privilege-level* – The user privilege level.

Example

Change the maximum privilege level for user 3 on channel 5 to `User`.

```
# clia user 3
Pigeon Point Shelf Manager Command Line Interpreter
    3: "administrator"
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
# clia user channel 3 5 0x60 2
Pigeon Point Shelf Manager Command Line Interpreter
    User 3, channel 5 access updated successfuly
# clia user 3
Pigeon Point Shelf Manager Command Line Interpreter
    3: "administrator"
        Channels 0-4 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
        Channel 5 Privilege level: "User"
            Flags: "Link Authentication" "IPMI Messaging"
        Channels 6-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
#
```

FRU Management Commands

This section describes the commands for FRU management.

activate

Syntax

`activate` *IPMB-address FRU-ID*

Purpose

This command sends the IPMI command `Set FRU Activation (Activate FRU)` to the specified FRU. The FRU is specified using the IPMB address of the owning IPM controller and the FRU device ID.

Example

Activate the IPM controller proper at address AA.

```
# clia activate aa 0
Pigeon Point Shelf Manager Command Line Interpreter
    Command issued via IPMB, status = 0 (0x0)
    Command executed successfully
#
```

deactivate

Syntax

`deactivate` *IPMB-address FRU-ID*

Purpose

This command sends the IPMI command `Set FRU Activation (Deactivate FRU)` to the specified FRU.

Example

Deactivate the IPM controller proper at address AA.

```
# clia deactivate aa 0
Pigeon Point Shelf Manager Command Line Interpreter
    Command issued via IPMB, status = 0 (0x0)
    Command executed successfully
#
```

fru

Syntax

`fru [-v] [`*addr* `[id=`*FRU-ID* `| type=`*site-type*`]] | [type=`*site-type* `[/`*site-number*`]]`

Purpose

This command shows information about a specific FRU.

The following information is shown for the FRU in standard mode:

- IPMB address and the FRU device ID.
- Entity ID, entity instance.
- Site type and number (if known).
- Current hot-swap state, previous hot-swap state, and cause of the last state change for the FRU. The hot-swap states M0–M7 are defined in the PICMG 3.0 specification as follows:
  - M0 – Not installed
  - M1 – Inactive
  - M2 – Activation request
  - M3 – Activation in progress
  - M4 – FRU active
  - M5 – Deactivation request
  - M6 – Deactivation in progress
  - M7 – Communication lost

The following information is shown for the FRU in verbose mode only:

- The FRU device type, device type modifier (only for `FRU-device-ID != 0`). This information is taken from the FRU Sensor Data Record (SDR) and conforms to section 37.12 of the IPMI specification.
- Device ID string from the FRU SDR.
- Current FRU power level and maximum FRU power level. Current assigned power consumption in Watts.

Example

Get standard FRU information about line card 20 at IPMB address AA.

```
# clia fru aa 0
Pigeon Point Shelf Manager Command Line Interpreter
aa: FRU # 0
```

```
    Entity: (0xa0, 0x60)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: "magnum_lc"
#
```

frucontrol

Syntax

`frucontrol` *IPMB-address FRU-ID option*

Purpose

This command sends the FRU Control command to the specified FRU, performing the specified operation on the FRU payload.

The parameter *option* specifies the option of the FRU Control command to be used:

- `cold_reset` (abbreviated as `cr`) – Performs a cold reset of the FRU payload.
- `warm_reset` (abbreviated as `wr`) – Performs a warm reset of the FRU payload.
- `graceful_reboot` (abbreviated as `gr`) – Performs a graceful reboot of the FRU payload.
- `diagnostic_interrupt` (abbreviated as `di`) – Issues the diagnostic interrupt.

Example

Issue a cold reset command to FRU 0 at IPMB address 9C.

```
# clia frucontrol 9c 0 cr
Pigeon Point Shelf Manager Command Line Interpreter
    FRU Control: Controller 0x9c, FRU ID # 0, command 0x00, status
0(0x0)
    Command executed successfully
#
```

fruinfo

Syntax

`fruinfo [-v]` *addr FRU-ID*

Purpose

This command shows FRU information in a user-friendly format.

Example

Display FRU information for line card 20 at IPMB address AA:

```
# clia fruinfo aa 0
Pigeon Point Shelf Manager Command Line Interpreter
aa: FRU # 0, FRU Info
Common Header:    Format Version = 1
Board Info Area:
    Version     = 1
    Language Code            = 25
    Mfg Date/Time           = Oct 29 18:14:00 2007 (6220454 minutes since 1996)
    Board Manufacturer       = Sun Microsystems, Inc.
    Board Product Name       = DCS 3456 - Line Card
    Board Serial Number      = SUG10224107043
    Board Part Number        = SUG SLR1022
    FRU Programmer File ID   = Fruid_LC.txt
Product Info Area:
    Version     = 1
    Language Code            = 25
    Manufacturer Name        = Sun Microsystems, Inc.
    Product Name             = DCS 3456 - Line Card
    Product Part / Model#    = SUG SLR1022
    Product Version          = Rev 2.8ECO_02
    Product Serial Number    = SUG10224107043
    Asset Tag                =
    FRU Programmer File ID   = Fruid_LC.txt
#
```

getfruledstate

Syntax

`getfruledstate [-v] [`*IPMB-addr* `state [`*FRU-ID* `[`*LED-ID*`|ALL]]]`

Purpose

This command shows the current FRU LED state on all levels of control that are enabled for the LEDs. In verbose mode, information about the colors supported by the LEDs is also shown.

Example

Show LED state for line card 20 at IPMB address AA.

```
# clia getfruledstate -v aa
Pigeon Point Shelf Manager Command Line Interpreter
aa: FRU # 0, Led # 0 ("BLUE LED"):
    Local Control LED State: LED OFF
    LED's color capabilities:
        Colors supported(0x02): BLUE
        Default LED Color in Local Control State(0x01): BLUE
        Default LED Color in Override State(0x01): BLUE
aa: FRU # 0, Led # 1 ("LED 1"):
    Local Control LED State: LED OFF
    LED's color capabilities:
        Colors supported(0x10): AMBER
        Default LED Color in Local Control State(0x04): AMBER
        Default LED Color in Override State(0x04): AMBER
aa: FRU # 0, Led # 2 ("LED 2"):
    Local Control LED State: LED ON, color: GREEN
    LED's color capabilities:
        Colors supported(0x08): GREEN
        Default LED Color in Local Control State(0x03): GREEN
        Default LED Color in Override State(0x03): GREEN
aa: FRU # 0, Led # 3 ("LED 3"):
    Override LED State (current state): LED OFF
    Local Control LED State: LED OFF
    LED's color capabilities:
        Colors supported(0x40): WHITE
        Default LED Color in Local Control State(0x06): WHITE
        Default LED Color in Override State(0x06): WHITE
#
```

setextracted

Syntax

`setextracted` *IPMB-address FRU-ID*

Purpose

This command notifies the shelf manager that the specified FRU has been physically extracted from the shelf.

Example

```
# clia setextracted 9c 0
Pigeon Point Shelf Manager Command Line Interpreter
Set FRU extracted state successfully
#
```

setfruledstate

Syntax

`setfruledstate` *IPMB-address FRU-ID LedId*|`ALL` *LedOp* [*LedColo*r]*LedOp* = `ON` | `OFF`|`LOCAL`|`BLINK` *onTime offTime*|`TEST` *onTime LedColor* = `BLUE`|`RED`|`GREEN`|`AMBER`|`ORANGE`|`WHITE NONE`| *number*

Purpose

This command enables you to set the state of a specific LED or all LEDs for the given FRU.

The first argument, *IPMB-address,* is the IPMB address of an IPM controller. The second argument, *FRU-ID,* is the FRU device ID. The third argument can be either an LED ID (a numerical value) or `ALL`. In the latter case, the specified operation applies to all LEDs.

The argument *LedOp* specifies the operation applied to the FRUs, based on the PICMG 3.0 specification. The operations are defined as follows:

- `ON` – Turns on the LED.
- `OFF` – Turns off the LED.
- `LOCAL` – Reverts to the local control of the LED.
- `BLINK` – Causes the LED to blink, repeatedly turning it on for *onTime* milliseconds and then turning it off for *offTime* milliseconds.
- `TEST` – Runs a lamp test for *onTime* milliseconds.

For the `TEST` operation, *onTime* must be less then 12800 ms (12.8 sec). For the `BLINK` operation, both *onTime* and *offTime* values must be within a 10 – 2500 ms range.

The optional parameter *LedColor* designates a color, either with a symbolic name or a decimal value. Symbolic names of colors correspond to decimal values in accordance with the PICMG 3.0 specification, as noted in the following list. (If the parameter is not specified, the default LED color is used.)

- BLUE = 1
- RED = 2

- GREEN = 3
- AMBER = 4
- ORANGE = 6
- NONE = 14 (don't change color)

Example

Enable blinking on LED 1 of FRU 0 of IPM controller at IPMB address 10h. The blinking is in the default color. The on duration is 100 ms. The off duration is 200 ms.

```
# clia setfruledstate 10 0 0 BLINK 100 200
Pigeon Point Shelf Manager Command Line Interpreter
    Setting FRU's led state completed successfully, status = 0x0
#
```

setlocked

Syntax

`setlocked` *IPMB-address FRU-ID* `0|1`

Purpose

This command sets the locked bit for the specified FRU to the specified state (`0` for unlock or `1` for lock).

The locked bit controls whether the FRU is allowed to autonomously progress from state M1 (inactive) to state M2 (activation request). If the locked bit is set, this transition is not allowed.

This command can be used to reactivate a previously manually deactivated FRU by clearing the locked bit for it.

Example

Clear the locked bit for the IPM controller proper at address 9C, thus allowing it to reactivate.

```
# clia setlocked 9c 0 0
Pigeon Point Shelf Manager Command Line Interpreter
Lock set successfully to 0x0
#
```

shelf

Syntax

`shelf` *subcommand*

The following subcommands are supported.

- `address_table`
- `cooling_state`
- `fans_state`
- `power_distribution`
- `power_management`
- `pci_connectivity`
- `ha_connectivity`
- `h110_connectivity`
- `point-to-point_connectivity`
- `MaxCurrent [feed]` *Amps*
- `MinVoltage [feed]` *Volts*
- `Activation` *addr FRU-ID* `1|0`
- `Deactivation` *addr FRU-ID* `1|0`
- `PwrCapability` *addr FRU-ID Watts*
- `PwrDelay` *addr FRU-ID 10ths-of-second*
- `Allowance` *seconds*
- `PwrReorder` *addr1 FRU-ID1* `before|after` *addr2 FRU-ID2*
- `info_refresh`
- `info_force_update`

Purpose

The syntax of the command `shelf` shows key Shelf FRU information, plus selected current operating data for the shelf. TABLE 4-2 lists the parameters supported by the `shelf` command:

TABLE 4-2 Parameter for `shelf` Command

Command Parameter	Provided Information
`cooling_state` (can be abbreviated to `cs`)	Shows the current cooling state of the shelf: • Normal – All temperature sensors show normal operating temperature. • Minor Alert – At least one temperature sensor is in minor alert state. None of the sensors is in major or critical alert state. • Major Alert – At least one temperature sensor is in major alert state. None of the sensors is in critical alert state. • Critical Alert – At least one temperature sensor is in critical alert state.
`fans_state` (can be abbreviated to `fs`)	Shows the current state of the fan tachometers in the shelf: • Normal – All fan tachometer sensors show normal operating speed. • Minor Alert – At least one fan tachometer sensor is in minor alert state. None of the sensors is in major or critical alert state. • Major Alert – At least one fan tachometer sensor is in major alert state. None of the sensors is in critical alert state. • Critical Alert – At least one fan tachometer sensor is in critical alert state.
`address_table` (can be abbreviated to `at`)	Shows the Address Table record in the Shelf FRU Info. The following information is provided: • Shelf Address (shown according to its type). • List of address table entries, showing Hardware Address, Site Number, and Site Type for each of them.
`power_distribution` (can be abbreviated to `pd`)	The following information is provided for each of the power feeds (mostly from the Shelf Power Distribution record of the Shelf FRU Information): • Maximum External Available Current. • Maximum Internal Current. • Minimum Expected Operating Voltage. • Actual Power Available. • Currently Used Power. • List of FRUs connected to the feed, showing Hardware Address and FRU Device ID for each FRU.

TABLE 4-2 Parameter for `shelf` Command *(Continued) (Continued)*

Command Parameter	Provided Information
`power_management` (can be abbreviated to `pm`)	The Shelf Power Management record in the Shelf FRU Info. This record contains a list of FRU Power Descriptors. For each descriptor the following information is provided: • Hardware Address. • FRU Device ID. • Maximum FRU Power Capability. • Shelf Manager Controlled Activation. • Delay Before Next Power On.
`pci_connectivity` (can be abbreviated to `pcic`)	The Shelf PCI Connectivity record in the Shelf FRU Info. The following information is provided: • PCI Slot Descriptor. • DSEL Connection. • Segment ID. • Extended PCI Slot Descriptor. • Geographic Address. • Interface Number. • System Slot Capable.
`ha_connectivity` (can be abbreviated to `ha`)	The Shelf HA Connectivity record in the Shelf FRU Info. The following information is provided: • Radial Connectivity Support.
`h110_connectivity` (can be abbreviated to `h110c`)	The Shelf H110 Connectivity record in the Shelf FRU Info. The following information is provided: • Geographic Address. • Segment ID.
`point-to-point_connectivity` (can be abbreviated to `ppc`)	The Shelf Point-to-Point Connectivity record in the Shelf FRU Info. The following information is provided: • Channel Type. • Channel Count. • Slot/Hw Address. • Channel Descriptor.

For the command parameters `cooling_state` and `fans_state`, the verbosity option `-v` is available. Enter the option before the command parameter (for example, `clia shelf -v cooling_state`). The command displays the list of sensors (temperature or fan tachometers) that contribute to the current state. Each sensor is shown as a tuple (*IPMB-address*, *sensor-number*).

Example

Get shelf cooling status.

```
# clia shelf cooling_state
Pigeon Point Shelf Manager Command Line Interpreter
Cooling state: "Normal"
#
```

showunhealthy

Syntax

```
showunhealthy
```

Purpose

This command shows the list of FRUs that appear to have a problem. The FRUs might be identified as:

- `Communication Lost.`
- `Communication lost due to local failure.`
- `Unexpected deactivation.`

For each FRU, the following information is shown – IPMB address and FRU device ID, current hot-swap state, previous hot-swap state, and cause of the last state change.

Example

Show the list of unhealthy components in the system.

```
# clia showunhealthy
Pigeon Point Shelf Manager Command Line Interpreter
There are no unhealthy components in the shelf.
#
```

Shelf Manager Commands

This section describes commands used to manage the Shelf Manager:

- "getipmbstate" on page 68

getipmbstate

Syntax

`getipmbstate` *IPMB-address* `[`*link*`]` (in radial IPMB-0 environment)

`getipmbstate` *IPMB-address* (in bused IPMB-0 environment)

Purpose

This command shows the current state of IPMB-0 on the target IPM controller.

In a bused environment, or in a radial environment if the target IPMC is not an IPMB hub, the argument *link* is not used.

In the radial environment, if the target IPM controller is an IPMB hub, the command works as follows:

- If *link* is omitted, the command prints information about the state of all radial IPMB links.
- If the *link* is present, the command prints information about the specific radial IPMB link (1 to 95).

In both cases, information about the state of both IPMB-A and IPMB-B is shown.

Example

Show the current state of IPMB-0 on the IPMC at IPMB address 10h.

```
# clia getipmbstate 10
Pigeon Point Shelf Manager Command Line Interpreter
10: Link: 0, LUN: 0, Sensor # 1 ("IPMB LINK")
```

```
    Bus Status: 0x8  (IPMB-A Enabled, IPMB-B Enabled)
    IPMB A State: 0x08 (LocalControl, No failure)
    IPMB B State: 0x08 (LocalControl, No failure)
#
```

ipmc

Syntax

`ipmc [-v] [`*IPMB-address*`]`

Purpose

This command shows information about the IPM controller at the specified address, or about all IPM controllers known to the Shelf Manager, if *IPMB-address* is omitted.

The following information is shown for the IPM controller in standard mode:

- IPMB address of the controller, as two hexadecimal digits.
- Entity ID and entity instance for the IPM controller.
- Maximum possible FRU device ID for the IPM controller.
- PICMG extension version. This version should be 2.0 for PICMG 3.0-compliant IPM controllers.
- Current hot-swap state, previous hot-swap state, and cause of the last state change for FRU device 0 of the IPM controller (which represents the IPM controller itself). The hot-swap states M0–M7 are defined in the PICMG 3.0 specification as follows:
  - M0 – Not installed
  - M1 – Inactive
  - M2 – Activation request
  - M3 – Activation in progress
  - M4 – FRU active
  - M5 – Deactivation request
  - M6 – Deactivation in progress
  - M7 – Communication lost

Example

Get information about the IPM controller at address 10.

```
# clia ipmc -v 10
Pigeon Point Shelf Manager Command Line Interpreter
10: Entity: (0xf0, 0x60) Maximum FRU device ID: 0x08
    PICMG Version 2.2
   Hot Swap State: M4 (Active), Previous: M3 (Activation In Process), Last State
Change Cause: Normal State Change (0x0)
    Device ID: 0x00, Revision: 0, Firmware: 2.51, IPMI ver 1.5
    Manufacturer ID: 00400a, Product ID: 0000, Auxiliary Rev: 00000000
    Device ID String: "ShMM-1500"
    Global Initialization: 0x0, Power State Notification: 0x0, Device
Capabilities: 0x29
    Controller provides Device SDRs
    Supported features: 0x29
        "Sensor Device" "FRU Inventory Device" "IPMB Event Generator"
    10: Base Interface (0x00), Channel: 1
        Link: Disabled Ports: 1
    10: Base Interface (0x00), Channel: 2
        Link: Disabled Ports: 1
#
```

localaddress

Syntax

```
localaddress
```

Purpose

This command shows the IPMB address of the current shelf manager, based on its hardware address (as opposed to its generic BMC address 0x20).

Example

```
# clia localaddress
Pigeon Point Shelf Manager Command Line Interpreter
Local IPMB Address = 0x10
#
```

poll

Syntax

`poll`

Purpose

This command initiates rediscovery of IPM controllers on IPMB-0 by sending the `Get Device ID` command to all IPMB addresses.

Example

```
# clia poll
Pigeon Point Shelf Manager Command Line Interpreter
IPMB polling thread started
#
```

sel

Syntax

`sel [-v] [`*IPMB-address* `[`*record-count* `[`*starting-entry*`]]]`

`sel clear [`*IPMB-address*`]`

`sel info [`*IPMB-address*`]`

Purpose

This command shows the contents of the system event log (SEL) on the specified IPM controller (at IPMB address 20h by default). The optional parameter *record-count* indicates how many records from the record number *starting-entry* in the SEL are shown. The optional parameter *starting-entry* is the entry number of the first SEL record to print, relative to the beginning of the SEL.

For each SEL record, the following information fields are shown:

- Record ID
- Record type (currently only events are supported, for which the word `Event` is shown)
- Time stamp (for timestamped records)
- Source address parameters – IPMB address, LUN, and channel number
- Type and number of the sensor that generated the event
- Event or reading type code
- Three bytes of event data, in raw and processed (if available) formats

The command `sel clear` clears the SEL on the specified IPM controller (at IPMB address 20h by default).

The `-v` option makes the SEL entries output more user-friendly.

Example

Read the SEL on the shelf manager.

```
# clia sel info
Pigeon Point Shelf Manager Command Line Interpreter
20: SEL version: 1.5
    Number of log entries: 525
    Free space: 7968 bytes
    Last addition timestamp: Jul 22 05:48:55 1939
    Last erase timestamp: May  8 22:20:55 2003
    Supported operations: 0x0f
#
```

session

Syntax

`session`

Purpose

This command shows information about active RMCP sessions. The information includes the following items:

- The maximum possible number of sessions and the number of currently active sessions.
- For each currently active session:
  - Session handle

- The user ID and name used during session activation
- Maximum session privilege level
- The IPMI channel number and type
- For LAN sessions, peer IP address and port number

Example

```
# clia session
Pigeon Point Shelf Manager Command Line Interpreter
32 sessions possible, 2 sessions currently active
Session: 1
   User: ID 1, Name: ""; Privilege Level: "Administrator"
   Channel: 1 ("LAN_802_3"); Peer IP address: 172.16.2.203, Port: 1764
Session: 2
   User: ID 1, Name: ""; Privilege Level: "Administrator"
   Channel: 1 ("LAN_802_3"); Peer IP address: 172.16.2.203, Port: 1765
#
```

setipmbstate

Syntax

`setipmbstate` *IPMB-address* `A|B [`*link*`] 1|0` (in radial IPMB-0 environment)

`setipmbstate` *IPMB-address* `A|B 1|0` (in bused IPMB-0 environment)

Purpose

This command enables or disables an IPMB link on the target IPM controller. The second argument defines the bus (IPMB-A or IPMB-B) to be enabled or disabled. The last argument defines the operation to be performed – `1` to enable link, `0` to disable link.

In a bused environment, and in a radial environment for target IPM controllers other than an IPMB hub, argument *link* is not used. For an IPMB hub controller in a radial environment, the argument *link* is optional.

If *link* is present, the command enables or disables the specific radial IPMB link (1 to 95). If *link* is omitted, the command enables or disables all the links on the IPMB hub in the radial system.

Example

Disable IPMB-A link on line card 20 at IPMB address AA.

```
# clia setipmbstate aa A 0
Pigeon Point Shelf Manager Command Line Interpreter
     Command executed successfully
#
```

shelfaddress

Syntax

`shelfaddress [`*up-to-20-characters-of-the-shelf-address*`]`

`shelfaddress -x` *byte1 ..... byteN*

Purpose

This command gets or sets the Shelf Address field of the Address Table within shelf FRU information.

Without the option `-x`, the new shelf address is specified by a double-quoted string that can contain any ASCII characters and can be as long as 20 characters.

If the option `-x` is specified, the new shelf address is specified as a sequence of hexadecimal bytes separated with spaces. Up to 20 bytes can be specified, each byte is represented with two hexadecimal digits (the `0x` prefix is optional).

Example

```
# clia shelfaddress
Pigeon Point Shelf Manager Command Line Interpreter
    Shelf Address Info:
# clia shelfaddress "New Shelf Address"
Pigeon Point Shelf Manager Command Line Interpreter
    Shelf Address Info set successfully
# clia shelfaddress
Pigeon Point Shelf Manager Command Line Interpreter
    Shelf Address Info: "New Shelf Address"
#
```

shmstatus

Syntax

```
shmstatus
```

Purpose

This command returns the shelf manager status in redundant configurations: `Active` or `Backup`. The ready for operation flag is a parameter that shows as `Yes`.

Example

```
# clia shmstatus -v
Pigeon Point Shelf Manager Command Line Interpreter
Host: "Active"
Ready For Operation: Yes
Detailed State Flags: "Shelf FRU Found" "Backup Healthy" "Initial Update Sent"
"RMCP Up"
#
```

switchover

Syntax

```
switchover
```

Purpose

This command initiates switchover of the redundant shelf manager instances.

Example

Initiate the switchover from either the active or backup instance.

```
# clia switchover
 This Shelf Manager is now active, but is shutting down to trigger
a switchover.
#
```

terminate

Syntax

```
terminate
```

Purpose

This command terminates the Shelf Manager without rebooting the shelf management card.

Example

Terminate a Shelf Manager from either the active or backup instance.

```
# clia terminate
#
```

Network Management Commands

This section describes commands for network management. The commands are:

getlanconfig

Syntax

`getlanconfig` *channel* [*parameter-name* [*additional-parameters*]]

`getlanconfig` *channel* [*parameter-number* [*additional-parameters*]]

Purpose

This command shows the value of the specified LAN configuration parameter on the specified channel. If no configuration parameter name or number is specified, all configuration parameters for the specified channel are shown.

TABLE 4-3 lists names and numbers of LAN configuration parameters supported by the `getlanconfig` command.

TABLE 4-3 LAN Configuration Parameters for `getlanconfig`

Parameter Name	Number	Description
`auth_support`	1	An 8-bit value that contains authentication type support flags for the LAN channel.
`auth_enables`	2	Five 8-bit values that contain authentication type enable flags for Callback, User, Operator, Administrator, and OEM privilege levels for the LAN channel.
`ip`	3	A string value that contains the IP address assigned to the LAN channel in dotted decimal notation (for example, `192.168.0.15`).
`ip_source`	4	A value that encodes the source of the assigned IP address.
`mac`	5	A string value that contains the MAC address assigned to the LAN channel as six hexadecimal byte values delimited by `:` symbols (for example, `00:A0:24:C6:18:2F`).
`subnet_mask`	6	A string value that contains the subnet mask assigned to the LAN channel in dotted decimal notation (for example, `255.255.255.0)`.
`ipv4_hdr_param`	7	Three 8-bit values that contain various IPv4 header parameters for sending RMCP packets: • Time-to-live • IP header flags (bits [7:5]) • Precedence (bits [7:5]) and type of service (bits [4:1])
`pri_rmcp_port`	8	A 16-bit value that contains the primary RMCP port number (the port used for regular RMCP communication).
`sec_rmcp_port`	9	A 16-bit value that contains the secondary RMCP port number (the port used for secure RMCP communication).
`arp_control`	10	Two flags that control Address Resolution Protocol (ARP) behavior on the LAN channel: • Enable responding to ARP requests • Enable sending gratuitous ARPs
`arp_interval`	11	The gratuitous ARP interval in seconds, in fixed-point format (potentially including a fractional part).
`dft_gw_ip`	12	A string value that contains the IP address of the default gateway in dotted decimal notation.

TABLE 4-3 LAN Configuration Parameters for `getlanconfig` *(Continued) (Continued)*

Parameter Name	Number	Description
`dft_gw_mac`	13	A string value that contains the MAC address of the default gateway as six hexadecimal byte values delimited by colons (`:`).
`backup_gw_ip`	14	A string value that contains the IP address of the backup gateway in dotted decimal notation.
`backup_gw_mac`	15	A string value that contains the MAC address of the backup gateway as six hexadecimal byte values delimited by colons (`:`).
`community`	16	A string value (up to 18 symbols) that is put into the `Community String` field in PET traps.
`destination_count`	17	The maximum number of LAN alert destinations supported on the LAN channel.
`destination_type`	18	The destination type identified by the specified set selector. If no set selector is given, all destination types are shown. Each destination type entry contains the following fields: • Destination type (0–7) • Alert acknowledge flag • Alert acknowledge timeout / retry interval in seconds (1–256) • Number of retries (0–7)
`destination_address`	19	The destination addresses associated with the specified set selector. If no set selector is given, all destination addresses are shown. Each destination address entry contains the following fields: • Gateway selector: `0` – use default, `1` – use backup • IP address (string in dotted decimal format) • MAC address (string of six hexadecimal byte values delimited by colons (`:`))

Example

Get and show the LAN parameter table for channel 1.

```
# clia getlanconfig 1
Pigeon Point Shelf Manager Command Line Interpreter
Authentication Type Support: 0x15 ( None MD5 Straight Password/Key )
Authentication Type Enables:
    Callback level: 0x00
    User level: 0x15 ( "None" "MD5" "Straight Password/Key" )
    Operator level: 0x15 ( "None" "MD5" "Straight Password/Key" )
    Administrator level: 0x15 ( "None" "MD5" "Straight Password/Key" )
```

```
    OEM level: 0x00
IP Address: 129.159.145.166
IP Address Source: Static Address (Manually Configured) (0x01)
MAC Address: 00:18:49:00:2d:4a
Subnet Mask: 255.255.255.0
IPv4 Header Parameters: 0x40:0x40:0x10
Primary RMCP Port Number: 0x026f
Secondary RMCP Port Number: 0x0298
BMC-generated ARP Control: 0x02
    Enable BMC-generated ARP Response
Gratuitous ARP Interval: 2.0 seconds
Default Gateway Address: 129.159.145.5
Default Gateway MAC Address: 00:30:85:11:34:c1
Backup Gateway Address: 0.0.0.0
Backup Gateway MAC Address: N/A
Community String: "public"
Number of Destinations: 16
Destination Type:
    N/A
Destination Address:
    N/A
#
```

setlanconfig

Syntax

`setlanconfig` *channel parameter-name additional-parameters*

`setlanconfig` *channel parameter-number additional-parameters*

Purpose

This command sets the value of the specified LAN configuration parameter on the specified channel.

TABLE 4-4 lists names and numbers of LAN configuration parameters supported by the `setlanconfig` command.

TABLE 4-4 LAN Configuration Parameters for `setlanconfig`

Parameter Name	Number	Description
`auth_enables`	2	Five 8-bit values that contain authentication type enable flags for Callback, User, Operator, Administrator, and OEM privilege levels for the LAN channel.
`ip`	3	A string value that contains the IP address assigned to the LAN channel in dotted decimal notation.
`subnet_mask`	6	A string value that contains the subnet mask assigned to the LAN channel in dotted decimal notation.
`ipv4_hdr_param`	7	Three 8-bit values that contain various IPv4 header parameters for sending RMCP packets: • Time-to-live • IP header flags (bits [7:5]) • Precedence (bits [7:5]) and type of service (bits [4:1])
`arp_control`	10	Two flags that control ARP behavior on the LAN channel: • Enable responding to ARP requests • Enable sending Gratuitous ARPs
`arp_interval`	11	The gratuitous ARP interval in a fixed-point format (where the integral part represents seconds and the fractional part represents milliseconds).
`dft_gw_ip`	12	A string value that contains the IP address of the default gateway in dotted decimal notation.
`backup_gw_ip`	14	A string value that contains the IP address of the backup gateway in dotted decimal notation.

TABLE 4-4 LAN Configuration Parameters for `setlanconfig` *(Continued) (Continued)*

Parameter Name	Number	Description
`community`	16	A string value (up to 18 symbols) that is put into the `Community String` field in PET traps.
`destination_type`	18	The destination type identified by the specified set selector. Set selector must be specified for this parameter. Each destination type entry contains the following fields: • Destination type (0-7) • Alert acknowledge flag • Alert acknowledge timeout / retry interval in seconds (1-256) • Number of retries (0-7)
`destination_address`	19	The destination addresses associated with the specified set selector. Set selector must be specified for this parameter. Each destination address entry contains the following fields: • Gateway selector: 0 – use default, 1 – use backup • IP address (string in dotted decimal format) • MAC address (string of six hexadecimal-byte values delimited by colons [`:`] symbols)

Example

```
# clia setlanconfig 1 destination_address 2 0 172.16.2.100 90:93:93:93:93:93
Pigeon Point Shelf Manager Command Line Interpreter
Destination Addresses set successfully
#
```

Other Commands

This section describes other clia sub-commands. They are:

alarm

Syntax

```
alarm [clear|minor|major|critical]
```

Purpose

This command provides access to the Telco alarm outputs. Parameters `minor`, `major`, and `critical` enable you to set the corresponding alarm output. Command invocation without parameters returns the status of Telco alarm outputs.

Example

```
# clia alarm
Pigeon Point Shelf Manager Command Line Interpreter
    alarm mask: 0x10
#
```

exit | quit

Syntax

```
exit
```

```
quit
```

Purpose

The command `exit` or `quit` exits the CLI interactive mode (which is entered by issuing `clia` without parameters).

Example

```
# exit
#
```

help

Syntax

`help [`*command* `[`*subcommand*`]]`

Purpose

This command shows help information for supported commands and their syntax.

Example

Get help for a specific command.

```
# clia help help
Pigeon Point Shelf Manager Command Line Interpreter
  Provides basic help information
  help [<command>]
#
```

sensordata

Syntax

`sensordata [`*IPMB-address* `[`*sensor-name* | `[`*lun*`:]`*sensor-number*`]]`

Purpose

This command shows the actual value of the specified sensor (for a threshold-based sensor) or the currently asserted states (for a discrete sensor).

The following information is shown for each sensor:

- IPMB address of the owning IPM controller
- Sensor number, sensor name (device ID string from the SDR), and the LUN by which the sensor can be accessed
- The sensor type and event-reading type code
- The sensor value (for threshold-based sensors) or the mask of currently asserted states (for discrete sensors) in raw form
- The threshold crossing status, in hexadecimal format and with decoding

Example

Get sensor data values for a temperature sensor `LM75_0` on line card 20 at IPMB address AA.

```
# clia sensordata aa 0:2
Pigeon Point Shelf Manager Command Line Interpreter
aa: LUN: 0, Sensor # 2 ("LM75_0")
    Type: Threshold (0x01), "Temperature" (0x01)
    Belongs to entity (0x03, 0x60): FRU # 0
```

```
    Status: 0xc0
        All event messages enabled from this sensor
        Sensor scanning enabled
        Initial update completed
    Raw data: 25 (0x19)
    Processed data: 25.000000 degrees C
    Status: 0x00
#
```

version

Syntax

```
version
```

Purpose

This command shows the version information for the shelf manager software.

Example

```
# clia version
Pigeon Point Shelf Manager Command Line Interpreter
Pigeon Point Shelf Manager ver. 2.5.1
Pigeon Point is a trademark of Pigeon Point Systems.
Copyright (c) 2002-2007 Pigeon Point Systems
All rights reserved
Build date/time: Feb 11 2008 14:34:10
Carrier: SUN_M1
Carrier subtype: 7; subversion: 5
#
```

CHAPTER 5

Switch Mapping and Routing Tables

This chapter provides tables that list the mapping and routing of ports within fabric cards and line cards. There is also a table that lists IPMB addresses used by the Sun Datacenter Switch 3456.

Topics in this chapter include:

IPMB Address Mappings

When you use the CLIA commands for managing the switch, you might need to provide the IPMB address of the respective CMC, fabric card, or line card for which you want to administer. TABLE 5-1 provides the IPMB addresses in hexadecimal format.

TABLE 5-1 IPMB Addresses for Switch Components

Item	IPMB Address	Item	IPMB Address	Item	IPMB Address	Item	IPMB Address
CMC0	10	LC9	94	LC20	aa	FC7	c0
CMC1	12	LC10	96	LC21	ac	FC8	c2
LC0	82	LC11	98	LC22	ae	FC9	c4
LC1	84	LC12	9a	LC23	b0	FC10	c6
LC2	86	LC13	9c	FC0	b2	FC11	c8

TABLE 5-1 IPMB Addresses for Switch Components *(Continued)*

Item	IPMB Address	Item	IPMB Address	Item	IPMB Address	Item	IPMB Address
LC3	88	LC14	9e	FC1	b4	FC12	ca
LC4	8a	LC15	a0	FC2	b6	FC13	cc
LC5	8c	LC16	a2	FC3	b8	FC14	ce
LC6	8e	LC17	a4	FC4	ba	FC15	d0
LC7	90	LC18	a6	FC5	bc	FC16	d2
LC8	92	LC19	a8	FC6	be	FC17	d4

Note – When you type the addresses into the CLIA commands, you might have to use upper case. For example, B8 instead of b8.

Fabric Card Mappings

Commands like `enableswitchport` and `checklinks` are more useful if you understand the relationship of the I3 switch chips and their ports to the iTRAC connectors and their ports.

Note – The mappings provided in the tables are the same for all fabric cards.

I3 Switch Chip to iTRAC Connector Map

TABLE 5-2 provides a mapping of the I3 switch chips and their ports to the iTRAC connector and its ports, for the fabric cards.

TABLE 5-2 Fabric Card I3 Switch Chip and Port to iTRAC Connector and Port Routing

I3	Port	→	iTRAC	Port	I3	Port	→	iTRAC	Port	I3	Port	→	iTRAC	Port	I3	Port	→	iTRAC	Port
0	1	→	2	7	2	1	→	2	3	4	1	→	2	2	6	1	→	2	6
0	2	→	8	7	2	2	→	8	3	4	2	→	8	2	6	2	→	8	6
0	3	→	1	7	2	3	→	1	3	4	3	→	1	2	6	3	→	1	6
0	4	→	14	7	2	4	→	14	3	4	4	→	14	2	6	4	→	14	6

TABLE 5-2 Fabric Card I3 Switch Chip and Port to iTRAC Connector and Port Routing *(Continued)*

I3	Port	→	iTRAC	Port	I3	Port	→	iTRAC	Port	I3	Port	→	iTRAC	Port	I3	Port	→	iTRAC	Port
0	5	→	20	7	2	5	→	20	3	4	5	→	20	2	6	5	→	20	6
0	6	→	13	7	2	6	→	13	3	4	6	→	13	2	6	6	→	13	6
0	7	→	6	7	2	7	→	6	3	4	7	→	6	2	6	7	→	6	6
0	8	→	0	7	2	8	→	0	3	4	8	→	0	2	6	8	→	0	6
0	9	→	7	7	2	9	→	7	3	4	9	→	7	2	6	9	→	7	6
0	10	→	18	7	2	10	→	18	3	4	10	→	18	2	6	10	→	18	6
0	11	→	12	7	2	11	→	12	3	4	11	→	12	2	6	11	→	12	6
0	12	→	19	7	2	12	→	19	3	4	12	→	19	2	6	12	→	19	6
0	13	→	23	7	2	13	→	23	3	4	13	→	23	2	6	13	→	23	6
0	14	→	17	7	2	14	→	17	3	4	14	→	17	2	6	14	→	17	6
0	15	→	22	7	2	15	→	22	3	4	15	→	22	2	6	15	→	22	6
0	16	→	11	7	2	16	→	11	3	4	16	→	11	2	6	16	→	11	6
0	17	→	5	7	2	17	→	5	3	4	17	→	5	2	6	17	→	5	6
0	18	→	10	7	2	18	→	10	3	4	18	→	10	2	6	18	→	10	6
0	19	→	15	7	2	19	→	15	3	4	19	→	15	2	6	19	→	15	6
0	20	→	21	7	2	20	→	21	3	4	20	→	21	2	6	20	→	21	6
0	21	→	16	7	2	21	→	16	3	4	21	→	16	2	6	21	→	16	6
0	22	→	3	7	2	22	→	3	3	4	22	→	3	2	6	22	→	3	6
0	23	→	9	7	2	23	→	9	3	4	23	→	9	2	6	23	→	9	6
0	24	→	4	7	2	24	→	4	3	4	24	→	4	2	6	24	→	4	6
1	1	→	2	5	3	1	→	2	1	5	1	→	2	4	7	1	→	2	8
1	2	→	8	5	3	2	→	8	1	5	2	→	8	4	7	2	→	8	8
1	3	→	1	5	3	3	→	1	1	5	3	→	1	4	7	3	→	1	8
1	4	→	14	5	3	4	→	14	1	5	4	→	14	4	7	4	→	14	8
1	5	→	20	5	3	5	→	20	1	5	5	→	20	4	7	5	→	20	8
1	6	→	13	5	3	6	→	13	1	5	6	→	13	4	7	6	→	13	8
1	7	→	6	5	3	7	→	6	1	5	7	→	6	4	7	7	→	6	8
1	8	→	0	5	3	8	→	0	1	5	8	→	0	4	7	8	→	0	8
1	9	→	7	5	3	9	→	7	1	5	9	→	7	4	7	9	→	7	8
1	10	→	18	5	3	10	→	18	1	5	10	→	18	4	7	10	→	18	8

TABLE 5-2 Fabric Card I3 Switch Chip and Port to iTRAC Connector and Port Routing *(Continued)*

I3	Port	→	iTRAC	Port	I3	Port	→	iTRAC	Port	I3	Port	→	iTRAC	Port	I3	Port	→	iTRAC	Port
1	11	→	12	5	3	11	→	12	1	5	11	→	12	4	7	11	→	12	8
1	12	→	19	5	3	12	→	19	1	5	12	→	19	4	7	12	→	19	8
1	13	→	23	5	3	13	→	23	1	5	13	→	23	4	7	13	→	23	8
1	14	→	17	5	3	14	→	17	1	5	14	→	17	4	7	14	→	17	8
1	15	→	22	5	3	15	→	22	1	5	15	→	22	4	7	15	→	22	8
1	16	→	11	5	3	16	→	11	1	5	16	→	11	4	7	16	→	11	8
1	17	→	5	5	3	17	→	5	1	5	17	→	5	4	7	17	→	5	8
1	18	→	10	5	3	18	→	10	1	5	18	→	10	4	7	18	→	10	8
1	19	→	15	5	3	19	→	15	1	5	19	→	15	4	7	19	→	15	8
1	20	→	21	5	3	20	→	21	1	5	20	→	21	4	7	20	→	21	8
1	21	→	16	5	3	21	→	16	1	5	21	→	16	4	7	21	→	16	8
1	22	→	3	5	3	22	→	3	1	5	22	→	3	4	7	22	→	3	8
1	23	→	9	5	3	23	→	9	1	5	23	→	9	4	7	23	→	9	8
1	24	→	4	5	3	24	→	4	1	5	24	→	4	4	7	24	→	4	8

iTRAC Connector to I3 Switch Chip Map

TABLE 5-3 provides a reverse-lookup mapping of the iTRAC connector and its ports, to the I3 switch chip and its port, for the fabric cards.

TABLE 5-3 Fabric Card iTRAC Connector and Port to I3 Switch Chip and Port Map

iTRAC	Port	→	I3	Port	iTRAC	Port	→	I3	Port	iTRAC	Port	→	I3	Port	iTRAC	Port	→	I3	Port
0	1	→	3	8	6	1	→	3	7	12	1	→	3	11	18	1	→	3	10
0	2	→	4	8	6	2	→	4	7	12	2	→	4	11	18	2	→	4	10
0	3	→	2	8	6	3	→	2	7	12	3	→	2	11	18	3	→	2	10
0	4	→	5	8	6	4	→	5	7	12	4	→	5	11	18	4	→	5	10
0	5	→	1	8	6	5	→	1	7	12	5	→	1	11	18	5	→	1	10
0	6	→	6	8	6	6	→	6	7	12	6	→	6	11	18	6	→	6	10
0	7	→	0	8	6	7	→	0	7	12	7	→	0	11	18	7	→	0	10
0	8	→	7	8	6	8	→	7	7	12	8	→	7	11	18	8	→	7	10
1	1	→	3	3	7	1	→	3	9	13	1	→	3	6	19	1	→	3	12

TABLE 5-3 Fabric Card iTRAC Connector and Port to I3 Switch Chip and Port Map *(Continued) (Continued)*

iTRAC	Port	→	I3	Port	iTRAC	Port	→	I3	Port	iTRAC	Port	→	I3	Port	iTRAC	Port	→	I3	Port
1	2	→	4	3	7	2	→	4	9	13	2	→	4	6	19	2	→	4	12
1	3	→	2	3	7	3	→	2	9	13	3	→	2	6	19	3	→	2	12
1	4	→	5	3	7	4	→	5	9	13	4	→	5	6	19	4	→	5	12
1	5	→	1	3	7	5	→	1	9	13	5	→	1	6	19	5	→	1	12
1	6	→	6	3	7	6	→	6	9	13	6	→	6	6	19	6	→	6	12
1	7	→	0	3	7	7	→	0	9	13	7	→	0	6	19	7	→	0	12
1	8	→	7	3	7	8	→	7	9	13	8	→	7	6	19	8	→	7	12
2	1	→	3	1	8	1	→	3	2	14	1	→	3	4	20	1	→	3	5
2	2	→	4	1	8	2	→	4	2	14	2	→	4	4	20	2	→	4	5
2	3	→	2	1	8	3	→	2	2	14	3	→	2	4	20	3	→	2	5
2	4	→	5	1	8	4	→	5	2	14	4	→	5	4	20	4	→	5	5
2	5	→	1	1	8	5	→	1	2	14	5	→	1	4	20	5	→	1	5
2	6	→	6	1	8	6	→	6	2	14	6	→	6	4	20	6	→	6	5
2	7	→	0	1	8	7	→	0	2	14	7	→	0	4	20	7	→	0	5
2	8	→	7	1	8	8	→	7	2	14	8	→	7	4	20	8	→	7	5
3	1	→	3	22	9	1	→	3	23	15	1	→	3	19	21	1	→	3	20
3	2	→	4	22	9	2	→	4	23	15	2	→	4	19	21	2	→	4	20
3	3	→	2	22	9	3	→	2	23	15	3	→	2	19	21	3	→	2	20
3	4	→	5	22	9	4	→	5	23	15	4	→	5	19	21	4	→	5	20
3	5	→	1	22	9	5	→	1	23	15	5	→	1	19	21	5	→	1	20
3	6	→	6	22	9	6	→	6	23	15	6	→	6	19	21	6	→	6	20
3	7	→	0	22	9	7	→	0	23	15	7	→	0	19	21	7	→	0	20
3	8	→	7	22	9	8	→	7	23	15	8	→	7	19	21	8	→	7	20
4	1	→	3	24	10	1	→	3	18	16	1	→	3	21	22	1	→	3	15
4	2	→	4	24	10	2	→	4	18	16	2	→	4	21	22	2	→	4	15
4	3	→	2	24	10	3	→	2	18	16	3	→	2	21	22	3	→	2	15
4	4	→	5	24	10	4	→	5	18	16	4	→	5	21	22	4	→	5	15
4	5	→	1	24	10	5	→	1	18	16	5	→	1	21	22	5	→	1	15
4	6	→	6	24	10	6	→	6	18	16	6	→	6	21	22	6	→	6	15
4	7	→	0	24	10	7	→	0	18	16	7	→	0	21	22	7	→	0	15

TABLE 5-3 Fabric Card iTRAC Connector and Port to I3 Switch Chip and Port Map *(Continued) (Continued)*

iTRAC	Port	→	I3	Port	iTRAC	Port	→	I3	Port	iTRAC	Port	→	I3	Port	iTRAC	Port	→	I3	Port
4	8	→	7	24	10	8	→	7	18	16	8	→	7	21	22	8	→	7	15
5	1	→	3	17	11	1	→	3	16	17	1	→	3	14	23	1	→	3	13
5	2	→	4	17	11	2	→	4	16	17	2	→	4	14	23	2	→	4	13
5	3	→	2	17	11	3	→	2	16	17	3	→	2	14	23	3	→	2	13
5	4	→	5	17	11	4	→	5	16	17	4	→	5	14	23	4	→	5	13
5	5	→	1	17	11	5	→	1	16	17	5	→	1	14	23	5	→	1	13
5	6	→	6	17	11	6	→	6	16	17	6	→	6	14	23	6	→	6	13
5	7	→	0	17	11	7	→	0	16	17	7	→	0	14	23	7	→	0	13
5	8	→	7	17	11	8	→	7	16	17	8	→	7	14	23	8	→	7	13

Line Card Mappings

Similar to the fabric card mappings, commands like `enableswitchport` and `checklinks` are more useful if you understand the relationship of the I3 switch chips and their ports to the iTRAC connectors and their ports and to the iPASS connectors and their LEDs.

Note – The mappings provided in the tables are the same for all line cards.

I3 Switch Chip to iTRAC Connector Map

TABLE 5-4 provides a mapping of the I3 switch chips and their ports to the iTRAC connector and its ports, for the line cards.

TABLE 5-4 Line Card I3 Switch Chip and Port to iTRAC Connector and Port Map

I3	Port	→	iTRAC	Port	I3	Port	→	iTRAC	Port	I3	Port	→	iTRAC	Port	I3	Port	→	iTRAC	Port
12	1	→	17	8	15	1	→	13	4	18	1	→	8	8	21	1	→	4	4
12	2	→	17	6	15	2	→	13	2	18	2	→	8	6	21	2	→	4	2
12	3	→	17	4	15	3	→	12	8	18	3	→	8	4	21	3	→	3	8
12	4	→	17	7	15	4	→	13	3	18	4	→	8	7	21	4	→	4	3

TABLE 5-4 Line Card I3 Switch Chip and Port to iTRAC Connector and Port Map *(Continued) (Continued)*

I3	Port	→	iTRAC	Port	I3	Port	→	iTRAC	Port	I3	Port	→	iTRAC	Port	I3	Port	→	iTRAC	Port
12	5	→	17	5	15	5	→	13	1	18	5	→	8	5	21	5	→	4	1
12	6	→	17	3	15	6	→	12	7	18	6	→	8	3	21	6	→	3	7
12	7	→	16	6	15	7	→	12	2	18	7	→	7	6	21	7	→	3	2
12	8	→	16	8	15	8	→	12	4	18	8	→	7	8	21	8	→	3	4
12	9	→	17	2	15	9	→	12	6	18	9	→	8	2	21	9	→	3	6
12	10	→	16	5	15	10	→	12	1	18	10	→	7	5	21	10	→	3	1
12	11	→	16	7	15	11	→	12	3	18	11	→	7	7	21	11	→	3	3
12	12	→	17	1	15	12	→	12	5	18	12	→	8	1	21	12	→	3	5
13	1	→	16	4	16	1	→	11	8	19	1	→	7	4	22	1	→	2	8
13	2	→	16	2	16	2	→	11	6	19	2	→	7	2	22	2	→	2	6
13	3	→	15	8	16	3	→	11	4	19	3	→	6	8	22	3	→	2	4
13	4	→	16	3	16	4	→	11	7	19	4	→	7	3	22	4	→	2	7
13	5	→	16	1	16	5	→	11	5	19	5	→	7	1	22	5	→	2	5
13	6	→	15	7	16	6	→	11	3	19	6	→	6	7	22	6	→	2	3
13	7	→	15	2	16	7	→	10	6	19	7	→	6	2	22	7	→	1	6
13	8	→	15	4	16	8	→	10	8	19	8	→	6	4	22	8	→	1	8
13	9	→	15	6	16	9	→	11	2	19	9	→	6	6	22	9	→	2	2
13	10	→	15	1	16	10	→	10	5	19	10	→	6	1	22	10	→	1	5
13	11	→	15	3	16	11	→	10	7	19	11	→	6	3	22	11	→	1	7
13	12	→	15	5	16	12	→	11	1	19	12	→	6	5	22	12	→	2	1
14	1	→	14	8	17	1	→	10	4	20	1	→	5	8	23	1	→	1	4
14	2	→	14	6	17	2	→	10	2	20	2	→	5	6	23	2	→	1	2
14	3	→	14	4	17	3	→	9	8	20	3	→	5	4	23	3	→	0	8
14	4	→	14	7	17	4	→	10	3	20	4	→	5	7	23	4	→	1	3
14	5	→	14	5	17	5	→	10	1	20	5	→	5	5	23	5	→	1	1
14	6	→	14	3	17	6	→	9	7	20	6	→	5	3	23	6	→	0	7
14	7	→	13	6	17	7	→	9	2	20	7	→	4	6	23	7	→	0	2
14	8	→	13	8	17	8	→	9	4	20	8	→	4	8	23	8	→	0	4
14	9	→	14	2	17	9	→	9	6	20	9	→	5	2	23	9	→	0	6

TABLE 5-4 Line Card I3 Switch Chip and Port to iTRAC Connector and Port Map *(Continued) (Continued)*

I3	Port	→	iTRAC	Port	I3	Port	→	iTRAC	Port	I3	Port	→	iTRAC	Port	I3	Port	→	iTRAC	Port
14	10	→	13	5	17	10	→	9	1	20	10	→	4	5	23	10	→	0	1
14	11	→	13	7	17	11	→	9	3	20	11	→	4	7	23	11	→	0	3
14	12	→	14	1	17	12	→	9	5	20	12	→	5	1	23	12	→	0	5

iTRAC Connector to I3 Switch Chip Map

TABLE 5-5 provides a reverse-lookup mapping of the iTRAC connector and its ports, to the I3 switch chip and its port, for the line cards.

TABLE 5-5 Line Card iTRAC Connector and Port to I3 Switch Chip and Port Map

iTRAC	Port	→	I3	Port	iTRAC	Port	→	I3	Port	iTRAC	Port	→	I3	Port	iTRAC	Port	→	I3	Port
0	1	→	23	10	4	5	→	20	10	9	1	→	17	10	13	5	→	14	10
0	2	→	23	7	4	6	→	20	7	9	2	→	17	7	13	6	→	14	7
0	3	→	23	11	4	7	→	20	11	9	3	→	17	11	13	7	→	14	11
0	4	→	23	8	4	8	→	20	8	9	4	→	17	8	13	8	→	14	8
0	5	→	23	12	5	1	→	20	12	9	5	→	17	12	14	1	→	14	12
0	6	→	23	9	5	2	→	20	9	9	6	→	17	9	14	2	→	14	9
0	7	→	23	6	5	3	→	20	6	9	7	→	17	6	14	3	→	14	6
0	8	→	23	3	5	4	→	20	3	9	8	→	17	3	14	4	→	14	3
1	1	→	23	5	5	5	→	20	5	10	1	→	17	5	14	5	→	14	5
1	2	→	23	2	5	6	→	20	2	10	2	→	17	2	14	6	→	14	2
1	3	→	23	4	5	7	→	20	4	10	3	→	17	4	14	7	→	14	4
1	4	→	23	1	5	8	→	20	1	10	4	→	17	1	14	8	→	14	1
1	5	→	22	10	6	1	→	19	10	10	5	→	16	10	15	1	→	13	10
1	6	→	22	7	6	2	→	19	7	10	6	→	16	7	15	2	→	13	7
1	7	→	22	11	6	3	→	19	11	10	7	→	16	11	15	3	→	13	11
1	8	→	22	8	6	4	→	19	8	10	8	→	16	8	15	4	→	13	8
2	1	→	22	12	6	5	→	19	12	11	1	→	16	12	15	5	→	13	12
2	2	→	22	9	6	6	→	19	9	11	2	→	16	9	15	6	→	13	9
2	3	→	22	6	6	7	→	19	6	11	3	→	16	6	15	7	→	13	6
2	4	→	22	3	6	8	→	19	3	11	4	→	16	3	15	8	→	13	3

TABLE 5-5 Line Card iTRAC Connector and Port to I3 Switch Chip and Port Map *(Continued) (Continued)*

iTRAC	Port	→	I3	Port	iTRAC	Port	→	I3	Port	iTRAC	Port	→	I3	Port	iTRAC	Port	→	I3	Port
2	5	→	22	5	7	1	→	19	5	11	5	→	16	5	16	1	→	13	5
2	6	→	22	2	7	2	→	19	2	11	6	→	16	2	16	2	→	13	2
2	7	→	22	4	7	3	→	19	4	11	7	→	16	4	16	3	→	13	4
2	8	→	22	1	7	4	→	19	1	11	8	→	16	1	16	4	→	13	1
3	1	→	21	10	7	5	→	18	10	12	1	→	15	10	16	5	→	12	10
3	2	→	21	7	7	6	→	18	7	12	2	→	15	7	16	6	→	12	7
3	3	→	21	11	7	7	→	18	11	12	3	→	15	11	16	7	→	12	11
3	4	→	21	8	7	8	→	18	8	12	4	→	15	8	16	8	→	12	8
3	5	→	21	12	8	1	→	18	12	12	5	→	15	12	17	1	→	12	12
3	6	→	21	9	8	2	→	18	9	12	6	→	15	9	17	2	→	12	9
3	7	→	21	6	8	3	→	18	6	12	7	→	15	6	17	3	→	12	6
3	8	→	21	3	8	4	→	18	3	12	8	→	15	3	17	4	→	12	3
4	1	→	21	5	8	5	→	18	5	13	1	→	15	5	17	5	→	12	5
4	2	→	21	2	8	6	→	18	2	13	2	→	15	2	17	6	→	12	2
4	3	→	21	4	8	7	→	18	4	13	3	→	15	4	17	7	→	12	4
4	4	→	21	1	8	8	→	18	1	13	4	→	15	1	17	8	→	12	1

I3 Switch Chip to I3 Switch Chip Map

TABLE 5-6 provides the routing of ports within the line card.

TABLE 5-6 Line Card I3 Switch Chip and Port to I3 Switch Chip and Port Map

I3	Port	→	I3	Port	I3	Port	→	I3	Port	I3	Port	→	I3	Port	I3	Port	→	I3	Port
0	1	→	19	23	6	1	→	16	21	12	13	→	2	4	18	14	→	10	12
0	2	→	16	23	6	2	→	12	23	12	14	→	5	11	18	15	→	3	11
0	3	→	22	23	6	3	→	17	19	12	15	→	0	9	18	16	→	11	11
0	4	→	20	23	6	4	→	18	21	12	16	→	4	1	18	17	→	0	8
0	5	→	14	15	6	5	→	23	20	12	17	→	11	1	18	18	→	2	12
0	6	→	21	22	6	6	→	19	19	12	18	→	3	8	18	19	→	5	5
0	7	→	13	20	6	7	→	21	23	12	19	→	7	2	18	20	→	9	10
0	8	→	18	17	6	8	→	13	18	12	20	→	1	4	18	21	→	6	4

TABLE 5-6 Line Card I3 Switch Chip and Port to I3 Switch Chip and Port Map *(Continued) (Continued)*

I3	Port	→	I3	Port	I3	Port	→	I3	Port	I3	Port	→	I3	Port	I3	Port	→	I3	Port
0	9	→	12	15	6	9	→	20	14	12	21	→	8	1	18	22	→	7	3
0	10	→	23	19	6	10	→	15	14	12	22	→	9	2	18	23	→	1	5
0	11	→	17	14	6	11	→	22	13	12	23	→	6	2	18	23	→	4	2
0	12	→	15	15	6	12	→	14	13	12	24	→	10	1	18	24	→	8	4
1	1	→	21	24	7	1	→	17	21	13	13	→	5	12	19	13	→	11	12
1	2	→	23	23	7	2	→	12	19	13	14	→	1	11	19	14	→	4	10
1	3	→	22	22	7	3	→	18	22	13	15	→	4	3	19	15	→	10	10
1	4	→	12	20	7	4	→	19	21	13	16	→	7	9	19	16	→	3	12
1	5	→	18	23	7	5	→	15	20	13	17	→	3	1	19	17	→	1	9
1	6	→	14	20	7	6	→	20	19	13	18	→	6	8	19	18	→	2	10
1	7	→	20	18	7	7	→	14	18	13	19	→	8	3	19	19	→	6	6
1	8	→	17	18	7	8	→	23	17	13	20	→	0	7	19	20	→	5	1
1	9	→	19	17	7	9	→	13	16	13	21	→	9	1	19	21	→	7	4
1	10	→	16	22	7	10	→	22	17	13	22	→	10	3	19	22	→	8	6
1	11	→	13	14	7	11	→	16	15	13	23	→	2	6	19	23	→	0	1
1	12	→	15	13	7	12	→	21	17	13	24	→	11	2	19	24	→	9	4
2	1	→	22	24	8	1	→	12	21	14	13	→	6	12	20	13	→	4	9
2	2	→	16	24	8	2	→	23	21	14	14	→	2	11	20	14	→	6	9
2	3	→	23	22	8	3	→	13	19	14	15	→	0	5	20	15	→	3	10
2	4	→	12	13	8	4	→	18	24	14	16	→	8	9	20	16	→	2	9
2	5	→	15	19	8	5	→	20	21	14	17	→	11	9	20	17	→	11	10
2	6	→	13	23	8	6	→	19	22	14	18	→	7	7	20	18	→	1	7
2	7	→	21	18	8	7	→	15	18	14	19	→	3	3	20	19	→	7	6
2	8	→	17	22	8	8	→	22	20	14	20	→	1	6	20	20	→	5	3
2	9	→	20	16	8	9	→	14	16	14	21	→	4	4	20	21	→	8	5
2	10	→	19	18	8	10	→	17	23	14	22	→	9	3	20	22	→	9	6
2	11	→	14	14	8	11	→	21	14	14	23	→	5	2	20	23	→	0	4
2	12	→	18	18	8	12	→	16	13	14	24	→	10	2	20	24	→	10	4
3	1	→	13	17	9	1	→	13	21	15	13	→	1	12	21	13	→	5	9
3	2	→	23	24	9	2	→	12	22	15	14	→	6	10	21	14	→	8	11

TABLE 5-6 Line Card I3 Switch Chip and Port to I3 Switch Chip and Port Map *(Continued) (Continued)*

I3	Port	→	I3	Port	I3	Port	→	I3	Port	I3	Port	→	I3	Port	I3	Port	→	I3	Port
3	3	→	14	19	9	3	→	14	22	15	15	→	0	12	21	15	→	4	7
3	4	→	15	21	9	4	→	19	24	15	16	→	9	9	21	16	→	3	9
3	5	→	17	24	9	5	→	21	20	15	17	→	10	9	21	17	→	7	12
3	6	→	16	20	9	6	→	20	22	15	18	→	8	7	21	18	→	2	7
3	7	→	22	18	9	7	→	16	16	15	19	→	2	5	21	19	→	10	6
3	8	→	12	18	9	8	→	23	14	15	20	→	7	5	21	20	→	9	5
3	9	→	21	16	9	9	→	15	16	15	21	→	3	4	21	21	→	11	4
3	10	→	20	15	9	10	→	18	20	15	22	→	4	6	21	22	→	0	6
3	11	→	18	15	9	11	→	22	14	15	23	→	11	3	21	23	→	6	7
3	12	→	19	16	9	12	→	17	13	15	24	→	5	4	21	24	→	1	1
4	1	→	12	16	10	1	→	12	24	16	13	→	8	12	22	13	→	6	11
4	2	→	18	23	10	2	→	14	24	16	14	→	10	7	22	14	→	9	11
4	3	→	13	15	10	3	→	13	22	16	15	→	7	11	22	15	→	5	7
4	4	→	14	21	10	4	→	20	24	16	16	→	9	7	22	16	→	4	11
4	5	→	17	20	10	5	→	22	19	16	17	→	11	8	22	17	→	7	10
4	6	→	15	22	10	6	→	21	19	16	18	→	4	12	22	18	→	3	7
4	7	→	21	15	10	7	→	16	14	16	19	→	5	6	22	19	→	10	5
4	8	→	23	18	10	8	→	17	16	16	20	→	3	6	22	20	→	8	8
4	9	→	20	13	10	9	→	15	17	16	21	→	6	1	22	21	→	11	6
4	10	→	19	14	10	10	→	19	15	16	22	→	1	10	22	22	→	1	3
4	11	→	22	16	10	11	→	23	15	16	23	→	0	2	22	23	→	0	3
4	12	→	16	18	10	12	→	18	14	16	24	→	2	2	22	24	→	2	1
5	1	→	19	20	11	1	→	12	17	17	13	→	9	12	23	13	→	11	5
5	2	→	14	23	11	2	→	13	24	17	14	→	0	11	23	14	→	9	8
5	3	→	20	20	11	3	→	15	23	17	15	→	5	10	23	15	→	10	11
5	4	→	15	24	11	4	→	21	21	17	16	→	10	8	23	16	→	5	8
5	5	→	18	19	11	5	→	23	13	17	17	→	11	7	23	17	→	7	8
5	6	→	16	19	11	6	→	22	21	17	18	→	1	8	23	18	→	4	8
5	7	→	22	15	11	7	→	17	17	17	19	→	6	3	23	19	→	0	10
5	8	→	23	16	11	8	→	16	17	17	20	→	4	5	23	20	→	6	5

TABLE 5-6 Line Card I3 Switch Chip and Port to I3 Switch Chip and Port Map *(Continued) (Continued)*

I3	Port	→	I3	Port	I3	Port	→	I3	Port	I3	Port	→	I3	Port	I3	Port	→	I3	Port
5	9	→	21	13	11	9	→	14	17	17	21	→	7	1	23	21	→	8	2
5	10	→	17	15	11	10	→	20	17	17	22	→	2	8	23	22	→	2	3
5	11	→	12	14	11	11	→	18	16	17	23	→	8	10	23	23	→	1	2
5	12	→	13	13	11	12	→	19	13	17	24	→	3	5	23	24	→	3	2

iPASS Connectors and LEDs to I3 Switch Chip and Port Map

TABLE 5-7 provides a mapping of the iPASS connector and its link LEDs to the respective I3 switch chip and port. The I3 switch chip and port are given as *chip-port*, where:

- *chip* – The identifying number of the I3 switch chip (00 - 11).
- *port* – The identifying number of the port (13 - 24).

TABLE 5-7 Connector and LED to I3 Switch Chip and Port Map

	iPASS Connector B			iPASS Connector A		
Connector Group	Left LED	Center LED	Right LED	Left LED	Center LED	Right LED
0	00-19	00-20	00-21	00-22	00-23	00-24
1	00-15	00-14	00-13	00-18	00-17	00-16
2	01-19	01-20	01-21	01-22	01-23	01-24
3	01-15	01-14	01-13	01-18	01-17	01-16
4	02-19	02-20	02-21	02-22	02-23	02-24
5	02-15	02-14	02-13	02-18	02-17	02-16
6	03-19	03-20	03-21	03-22	03-23	03-24
7	03-15	03-14	03-13	03-18	03-17	03-16
8	04-19	04-20	04-21	04-22	04-23	04-24
9	04-15	04-14	04-13	04-18	04-17	04-16
10	05-19	05-20	05-21	05-22	05-23	05-24
11	05-15	05-14	05-13	05-18	05-17	05-16
12	06-19	06-20	06-21	06-22	06-23	06-24
13	06-15	06-14	06-13	06-18	06-17	06-16

TABLE 5-7 Connector and LED to I3 Switch Chip and Port Map *(Continued) (Continued)*

	iPASS Connector B			iPASS Connector A		
Connector Group	**Left LED**	**Center LED**	**Right LED**	**Left LED**	**Center LED**	**Right LED**
14	07-19	07-20	07-21	07-22	07-23	07-24
15	07-15	07-14	07-13	07-18	07-17	07-16
16	08-19	08-20	08-21	08-22	08-23	08-24
17	08-15	08-14	08-13	08-18	08-17	08-16
18	09-19	09-20	09-21	09-22	09-23	09-24
19	09-15	09-14	09-13	09-18	09-17	09-16
20	10-19	10-20	10-21	10-22	10-23	10-24
21	10-15	10-14	10-13	10-18	10-17	10-16
22	11-19	11-20	11-21	11-22	11-23	11-24
23	11-15	11-14	11-13	11-18	11-17	11-16

For example, the center LED on iPASS connector 5A is I3 switch chip 02, port 17.

I3 Switch Chip and Port to iPASS Connector and LED Map

A reverse-lookup mapping of the I3 switch chip and port to iPASS connector and its link LEDs is provided in TABLE 5-8 and TABLE 5-9. The connector and LED are given as *connectorletter-location*, where:

- *connector* – The identifying number of the connector group (0 - 23).
- *letter* – The letter identifying the upper (B) or lower (A) connector.
- *location* – The location of the LED: left, center, or right.

TABLE 5-8 I3 Switch Chip (00 - 05) and Port to iPASS Connector and LED Map

Port	I3 Switch 00	I3 Switch 01	I3 Switch 02	I3 Switch 03	I3 Switch 04	I3 Switch 05
13	1B-right	3B-right	5B-right	7B-right	9B-right	11B-right
14	1B-center	3B-center	5B-center	7B-center	9B-center	11B-center
15	1B-left	3B-left	5B-left	7B-left	9B-left	11B-left
16	1A-right	3A-right	5A-right	7A-right	9A-right	11A-right
17	1A-center	3A-center	5A-center	7A-center	9A-center	11A-center
18	1A-left	3A-left	5A-left	7A-left	9A-left	11A-left

TABLE 5-8 I3 Switch Chip (00 - 05) and Port to iPASS Connector and LED Map *(Continued) (Continued)*

Port	I3 Switch 00	I3 Switch 01	I3 Switch 02	I3 Switch 03	I3 Switch 04	I3 Switch 05
19	0B-left	2B-left	4B-left	6B-left	8B-left	10B-left
20	0B-center	2B-center	4B-center	6B-center	8B-center	10B-center
21	0B-right	2B-right	4B-right	6B-right	8B-right	10B-right
22	0A-left	2A-left	4A-left	6A-left	8A-left	10A-left
23	0A-center	2A-center	4A-center	6A-center	8A-center	10A-center
24	0A-right	2A-right	4A-right	6A-right	8A-right	10A-right

TABLE 5-9 I3 Switch Chip (06 - 11) and Port to iPASS Connector and LED Map

Port	I3 Switch 06	I3 Switch 07	I3 Switch 08	I3 Switch 09	I3 Switch 10	I3 Switch 11
13	13B-right	15B-right	17B-right	19B-right	21B-right	23B-right
14	13B-center	15B-center	17B-center	19B-center	21B-center	23B-center
15	13B-left	15B-left	17B-left	19B-left	21B-left	23B-left
16	13A-right	15A-right	17A-right	19A-right	21A-right	23A-right
17	13A-center	15A-center	17A-center	19A-center	21A-center	23A-center
18	13A-left	15A-left	17A-left	19A-left	21A-left	23A-left
19	12B-left	14B-left	16B-left	18B-left	20B-left	22B-left
20	12B-center	14B-center	16B-center	18B-center	20B-center	22B-center
21	12B-right	14B-right	16B-right	18B-right	20B-right	22B-right
22	12A-left	14A-left	16A-left	18A-left	20A-left	22A-left
23	12A-center	14A-center	16A-center	18A-center	20A-center	22A-center
24	12A-right	14A-right	16A-right	18A-right	20A-right	22A-right

For example, I3 switch chip 06, port 18 is the left LED on connector 13A.

Route Through the Switch

By combining the information from the tables of this chapter, it is possible to determine a route through the Sun Datacenter Switch 3456. This section will describe a sample situation that might occur.

1. A route is initiated at line card LC6, connector 12A. The left LED blinks.

2. Using TABLE 5-7, it is determined that the link will route to I3 switch chip 06, through port 22.
3. The subnet manager instructs I3 switch chip 06 to use port 4 to forward the link.
4. Using TABLE 5-6, it is determined that the link will route across the line card to I3 switch chip 18 port 21.
5. The subnet manager instructs I3 switch chip 18 to use port 12 to forward the link.
6. Using TABLE 5-4, it is determined that the link will route through iTRAC connector 8 port 1.
7. The link leaves line card LC6 and enters fabric card FC8 at iTRAC connector 6 port 1.
8. Using TABLE 5-3, it is determined that the link will route to I3 switch chip 03 port 7.
9. The subnet manager instructs I3 switch chip 03 to use port 22 to forward the link.
10. Using TABLE 5-2, it is determined that the link will route to iTRAC connector 3 port 1.
11. The link leaves fabric card FC8 and enters line card LC3 at iTRAC connector 8 port 1.
12. Using TABLE 5-5, it is determined that the link will route to I3 switch chip 18 port 12.
13. The subnet manager instructs I3 switch chip 18 to use port 14 to forward the link.
14. Using TABLE 5-6 again, it is determined that the link will route across the line card to I3 switch chip 10 port 12.
15. The subnet manager instructs the I3 switch chip 10 to use port 24 to forward the link.
16. Using TABLE 5-9, it is determined that the link will exit line card LC3 at connector 20A. The right LED blinks.

Index

L

M

N

O

P

Q

R

S

T

U

V