Sun Netra CT900 Server

Administration and Reference Manual

Part No.: 819-1177-18
February 2011

Contents

READ AND DELETE

1. To display the "Beta Draft" footer , show the BetaDraft conditional setting.

2. After Beta, hide the BetaDraft conditional setting.

Figures

READ AND DELETE

1. To display the "Beta Draft" footer , show the BetaDraft conditional setting.

2. After Beta, hide the BetaDraft conditional setting.

Tables

Preface

This manual contains configuration and administration information for system administrators of Oracle's Sun Netra CT900 server. Shelf Manager and IPMI command reference information is also provided.

This manual assumes you are familiar with UNIX commands and networks, the PICMG 3.x AdvancedTCA Base Specification, and the Intelligent Platform Management Interface (IPMI).

Using UNIX Commands

This document might not contain information on 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
- Oracle Solaris Operating System (Oracle Solaris OS) documentation, which is at:

  `http://www.oracle.com/technetwork/indexes/documentation/index.html`

Shell Prompts

Shell	Prompt
C shell	*machine-name*%
C shell superuser	*machine-name*#
Bourne shell and Korn shell	$
Bourne shell and Korn shell superuser	#
U-Boot shell	shmm500
Linux shell	#
Shelf Manager command-line interface (CLI)	CLI>

Related Documentation

The Sun Netra CT900 server documentation is listed in the following table. Except for the *Important Safety Information for Sun Hardware Systems*, all the documents listed are available online at:

http://www.oracle.com/pls/topic/lookup?ctx=ct900&id=homepage

Application	Title	Format	Location
Start Here	*Sun Netra CT900 Server Start Here*	Printed PDF	Shipping kit Online
Latest information	*Sun Netra CT900 Server Product Notes*	PDF HTML	Online
Overview	*Sun Netra CT900 Server Overview*	PDF HTML	Online
Installation	*Sun Netra CT900 Server Installation Guide*	PDF HTML	Online
Service	*Sun Netra CT900 Server Service Manua*	PDF HTML	Online

Application	Title	Format	Location
Reference	*Sun Netra CP3140 Switch Software Reference Manual*	PDF HTML	Online
Safety	*Sun Netra CT900 Server Safety and Compliance Manual*	PDF HTML	Online
Safety	*Important Safety Information for Sun Hardware Systems (printed version only)*	Printed	Shipping kit

You might want to refer to documentation on the following products for additional information: the Oracle Solaris OS, OpenBoot PROM firmware, the Sun Netra CP3010 board, Sun Netra CP3020 board, Sun Netra ATCA CP3220 blade server, Sun Netra ATCA CP3060 blade server, and Sun Netra ATCA CP3260 blade server. These documents are available online.

Documentation, Support, and Training

These web sites provide additional resources:

- Documentation `http://www.oracle.com/technetwork/indexes/documentation/index.html`
- Support `https://support.oracle.com`
- Training `https://education.oracle.com`

CHAPTER 1

Introduction

This chapter includes the following sections:

Sun Netra CT900 Server Software

The Sun Netra CT900 server shelf software includes:

- Shelf Manager
- Operating systems
- Firmware (U-Boot, IPMC)

Note – The Advanced Telecom Computing Architecture® (ATCA) has adopted the term *shelf* for alignment with typical practice in telecommunications Traditionally, the term *chassis* has been used with essentially the same meaning.

The software is described in TABLE 1-1 and represented logically, with the hardware, in FIGURE 1-1.

TABLE 1-1 Sun Netra CT Server Software for System Administrators

Category	Name	Description
Shelf Management	Pigeon Point Shelf Manager	The Shelf Manager software runs on the shelf management card (ShMM) and is factory installed. It provides Remote Management Control Protocol (RMCP) and CLI access to IPMI for managing the server.
	Command-line interface (CLI)	The CLI is a on-board user interface to the Shelf Manager.
Operating System (on shelf management card)	Monterey Linux	Monterey Linux is installed on the shelf management card (ShMM) and runs the Shelf Manager software application.
Firmware	U-Boot	Firmware on the shelf management cards that performs power-on self-test (POST) and controls booting of the shelf management card software.
	Intelligent Platform Management Controller (IPMC)	System management controller firmware that enables communication over the IPMI controller on a Sun-supported node board, such as the Sun Netra CP3x60 boards.
Operating Systems (on Sun Netra ATCA node boards)	Oracle Solaris Operating System (Oracle Solaris OS)	Oracle Solaris OS runs on Sun Netra ATCA node boards, like the Sun Netra CP3010, Sun Netra CP3x20, and CP3x60 node boards. Oracle Solaris 10 is optionally preinstalled on the Sun Netra node boards. Oracle Solaris 10 and other versions of the Oracle Solaris OS can be downloaded and installed by the user.
	Linux OS	Some versions of the Linux OS run on the Sun Netra CP3x20 boards. Refer to the board's documentation for Linux OS support.

The Sun Netra CT900 server has two shelf management cards (ShMMs) and provides shelf management card failover from the active shelf management card to the backup shelf management card for certain hardware and software events. The *active* shelf management card is used for system-level configuration, administration, and management of most of the components connected to the midplane. The *backup* shelf management card provides redundancy and failover capability for the active shelf management card.

The switching fabric boards connect the shelf management card and the node boards internally, and have Ethernet ports on the rear for external connectivity.

Some Sun Netra ATCA node boards and rear transition modules (RTMs) accept peripherals, such as disks. The Sun Netra ATCA node boards also run user applications. In a Sun Netra CT 900 server, each node board runs its own copy of an operating system, and each is therefore considered a server. The shelf management cards, node boards, switching fabric boards, and the other system field-replaceable units (FRUs) make up a system.

Note – In this manual, the use of the term *node board* refers to a Sun Netra ATCA node board (or blade server), such as the Sun Netra CP3010 board, unless otherwise specified.

Third-party ATCA node boards that are PICMG® 3.x compliant can be used in the Sun Netra CT900 server. These boards do not necessarily run the Oracle Solaris OS, and they do not run the Sun Netra CT900 server system management software. Because of this, they cannot be managed to the same extent as the Sun Netra node boards.

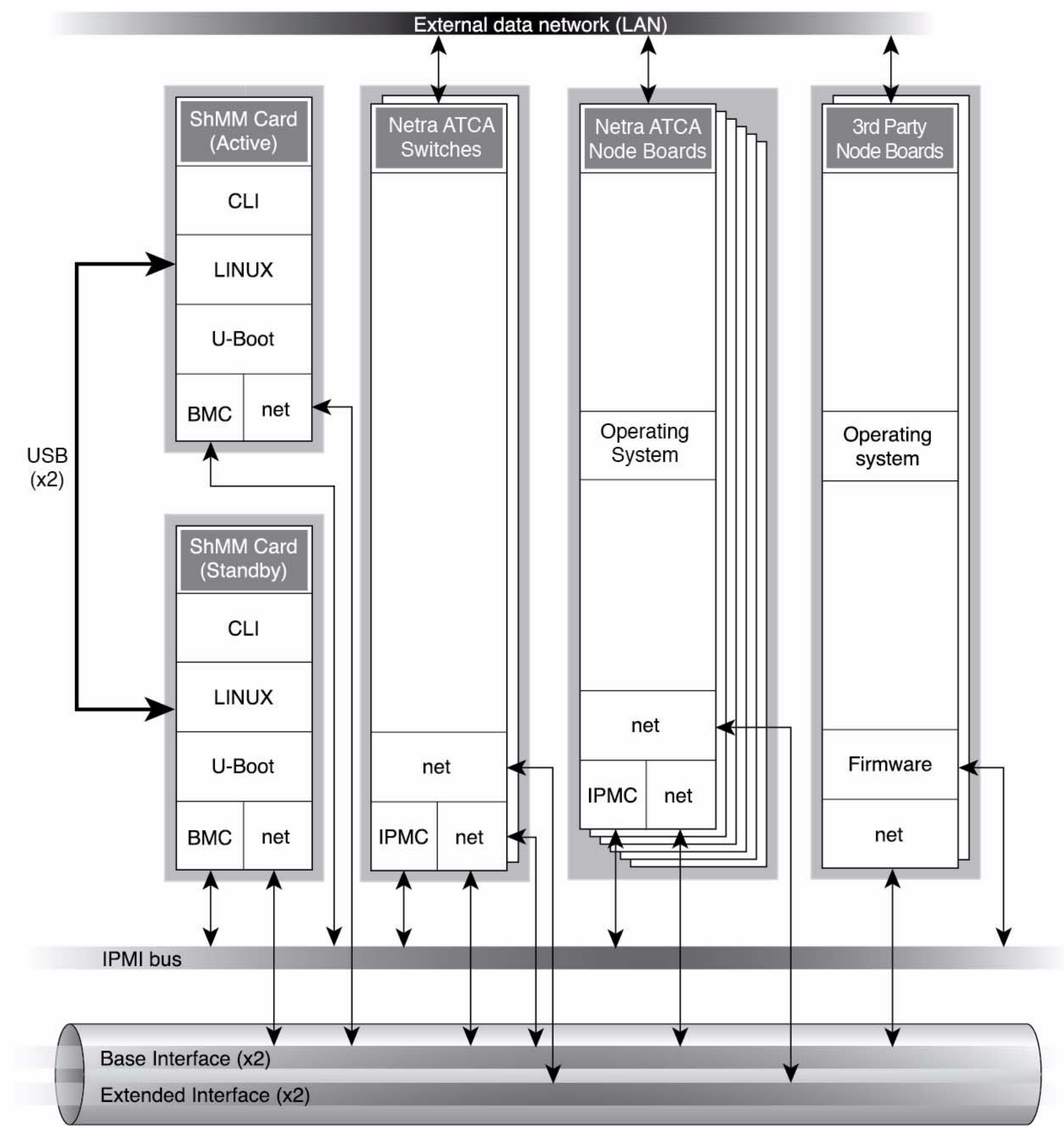
TABLE 1-2 summarizes how you can access the various boards. The shelf management card supports 22 sessions (1 Tip and 21 Telnet connections) at one time

TABLE 1-2 Sun Netra CT900 Server Board Access Methods

Board	Access Methods
Shelf Alarm Panel (SAP)	The front panel has the following ports: • Two serial (RS-232) ports with RJ-45 DTE connectors. The Serial 1 port is the console connection for the upper shelf management card (ShMM1), the default *active* card. The Serial 2 port is the console connection for the lower shelf management card (ShMM2), the default *backup* card. • Telco Alarm connector (DB-15)
Switch boards (slots 7 and 8)	Multiple Ethernet ports for Telnet connection on front panel. **Note -** A rear transition module (RTM) is required for rear access to these ports. Either the rear or front port connectors can be used, but not both at the same time. If you connect a cable to both ports, only the front port is active.
Node board (Sun Netra ATCA node boards) (slots 1 through 6 and 9 through 14)	For the Sun Netra CP3010 board, the front panel has the following ports: • Two serial ports (console) for Tip or ASCII terminal connection • Two Ethernet ports for Telnet connection • One 4X Serial Attached SCSI (SAS) port connector **Note -** An RTM is required for rear access to these ports. Either the rear or front port connectors can be used, but not both at the same time. If you connect a cable to both ports, only the front port is active. For the Sun Netra CP3020 and CP3060 boards, the front panel has the following ports: • One serial port (console) for Tip or ASCII terminal connection • Two Ethernet ports for Telnet connections For more information, refer to the Sun Netra ATCA CPU board documentation for the specific node board. All ATCA node boards can be accessed from the ShMM using the `console` CLI command.
Third-party node boards (slots 1 through 6 and 9 through 14)	Third-party board dependent.

The hardware interfaces include the Intelligent Platform Management Interface (IPMI), the base interface and extended interface, and the network interface on the shelf management cards, the node boards, and the switching fabric boards.

FIGURE 1-1 Logical Representation of Software and Hardware Interfaces in Sun Netra CT900 Server



Introduction to Shelf Manager

The Shelf Manager is a shelf-level management solution for ATCA products. The shelf management card provides the necessary hardware to run the Shelf Manager within an ATCA shelf. This overview focuses on aspects of the Shelf Manager and shelf management card that are common to any shelf management carrier used in an ATCA context.

Overview of Intelligent Platform Management in ATCA

The Shelf Manager and shelf management card are Intelligent Platform Management (IPM) building blocks designed for modular platforms like ATCA, in which there is a strong focus on a dynamic population of FRUs and maximum service availability. The IPMI specification provides a solid foundation for the management of such platforms, but requires significant extension to support them well. PICMG 3.0, the ATCA specification, defines the necessary extensions to IPMI.

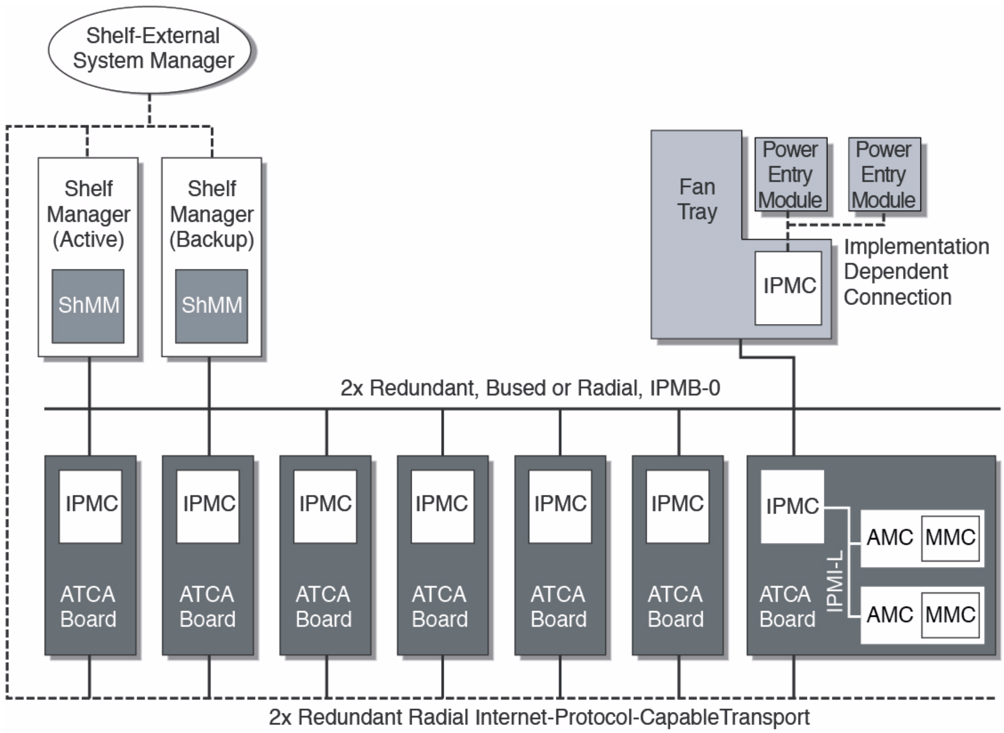
FIGURE 1-2 shows the logical elements of an example ATCA shelf, identified in terms of the ATCA specification.

An AdvancedTCA Shelf Manager communicates inside the shelf with IPM Controllers, each of which is responsible for local management of one or more field replaceable units (FRUs), such as boards, fan trays or power entry modules. Management communication within a shelf occurs primarily over the Intelligent Platform Management Bus (IPMB), which is implemented on a dual-redundant basis as IPMB-0 in AdvancedTCA.

The PICMG Advanced Mezzanine Card (AdvancedMC or AMC) specification, AMC.0, defines a hot-swappable mezzanine form factor designed to fit smoothly into the physical and management architecture of AdvancedTCA.

FIGURE 1-2 includes an AMC carrier with an IPMC and two installed AMC modules, each with a Module Management Controller (MMC). On-carrier management communication occurs over IPMB-L ("L" for Local).

FIGURE 1-2 Example of ATCA Shelf



An overall System Manager (typically external to the shelf) can coordinate the activities of multiple shelves. A System Manager typically communicates with each Shelf Manager over an Ethernet or serial interface.

FIGURE 1-2 shows three levels of management: board, shelf, and system. The next section addresses the Shelf Manager software and card which implement an ATCA-compliant shelf manager (ShMM).

Shelf Manager and Shelf Management Card

The Shelf Manager (consistent with ATCA Shelf Manager requirements) has two main responsibilities:

- Manage and track the FRU population and common infrastructure of a shelf, especially the power, cooling and interconnect resources and their usage. Within the shelf, this management and tracking primarily occurs through interactions between the Shelf Manager and the IPM controllers over Intelligent Platform Management bus 0 (IPMB-0).
- Enable the overall System Manager to join in that management and tracking through the System Manager interface, which is typically implemented over Ethernet.

Much of the Shelf Manager software is devoted to routine missions such as powering a shelf up or down and handling the arrival or departure of FRUs, including negotiating assignments of power and interconnect resources. In addition, the Shelf Manager can take direct action when exceptions are raised in the shelf. For instance, in response to temperature exceptions the Shelf Manager can raise the fan levels or, if that step is not sufficient, even start powering down FRUs to reduce the heat load in the shelf.

Shelf Manager Features

The Shelf Manager software features include:

- Executes on the shelf management cards installed in the shelf.
- Conforms to the ATCA specification.
- Monitors activities within the shelf via the ATCA-specified dual-redundant Intelligent Platform Management bus (IPMB).
- Accepts and logs events posted by any intelligent FRU in the shelf (reflecting exceptions in temperatures, voltages, etc.); posts alerts outside the shelf based on configurable IPMI Platform Event Filters.
- Supports hot swapping of Field Replaceable Units (FRUs), while maintaining full management visibility.
- Interfaces to standard Telco Alarm infrastructures, via shelf management implemented dry contact relays.
- Supports redundant Shelf Manager instances for high availability.
- Integrates a watchdog timer, which resets the shelf management card if not periodically strobed; such resets automatically trigger a switchover to the backup shelf management card, if configured.

- Includes a battery-backed real-time clock for time stamping events.
- Implements a rich set of shelf-external interfaces accessible over Ethernet, including RMCP, required by ATCA, and CLI.

Shelf Manager Switchover

The Shelf Manager can be configured with active/backup instances to maximize availability. FIGURE 1-3 shows how both instances are accessible to the System Manager, with only the active instance interacting at any given time. Similarly, only the active instance communicates over IPMB-0 with the IPM controller population in the shelf. The two instances communicate with each other over TCP/IP, with the active instance posting incremental state updates to the backup. As a result, the backup can quickly step into the active role if necessary.

FIGURE 1-3 Shelf Manager Switchover Signal

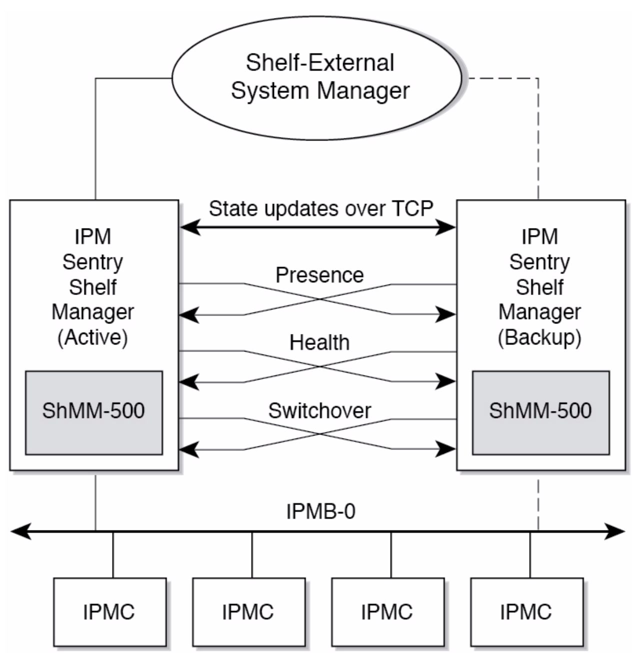


TABLE 1-3 list the signals and descriptions.

TABLE 1-3 Hardware Signals and Interfaces Supporting Switchover

Hardware	Description
USB interface	The primary interface between the shelf management cards; it is used to send heartbeats and state synchronization information. Both shelf management cards must view the same FRU, such as a particular fan tray or a node board in a certain slot, in the same state, for example, powered on
`#SWITCHOVER`	The backup instance can force a switchover if necessary.
`#PRSNT`	This signal indicates the presence of a shelf management card
`#HEALTHY`	This signal indicates the overall health of the shelf management card, including both the hardware and software

Switchover Details

The active Shelf Manager exposes the ShMM device (address 20h) on IPMB, manages IPMB and the IPM controllers, and interacts with the System Manager over RMCP and other shelf-external interfaces. It maintains an open TCP connection with the backup Shelf Manager. It communicates all changes in the state of the managed objects to the backup Shelf Manager.

The backup Shelf Manager does not expose the ShMM on IPMB, does not actively manage IPMB and IPM controllers, and does not interact with the System Manager via the shelf-external interfaces (with one exception noted below). Instead, it maintains the state of the managed objects in its own memory (volatile and non-volatile) and updates the state as directed by the active Shelf Manager.

The backup Shelf Manager can become active as the result of a switchover. Two types of switchover are defined:

- cooperative switchover – the active and backup Shelf Managers negotiate the transfer of responsibilities from the active to the backup Shelf Manager; this mode is supported via the CLI `switchover` command issued on the active or backup Shelf Manager.
- forced switchover – the backup Shelf Manager determines that the active Shelf Manager is no longer alive or healthy, and forcefully takes on the responsibilities of the active Shelf Manager.

The backup Shelf Manager recognizes the departure of the active Shelf Manager when the Remote Healthy or Remote Presence low-level signal becomes inactive. The Remote Presence signal monitors the presence of the peer Shelf Manager; this signal going inactive means that the board hosting the peer Shelf Manager has been removed from the shelf. The Remote Healthy signal is set by the peer Shelf Manager during initialization; this signal going inactive means that the remote Shelf Manager has become unhealthy (typically, has been powered off or reset).

Another situation that needs some action from the backup Shelf Manager is when the TCP connection between the Shelf Managers gets closed. This happens when the communication link between the two Shelf Managers is broken, when the shelf management process on the active Shelf Manager terminates (either voluntarily or involuntarily), or when a software exception occurs. Since the TCP `keepalive` option is enabled on the connection, it closes shortly after the active shelf management card is switched off or reset.

In the case of Shelf Manager termination, it is possible that the TCP connection is closed *before* the Remote Healthy signal becomes inactive. To determine why the TCP connection closed, the backup Shelf Manager samples the state of the Remote Healthy signal immediately and, if it is still active, again after some delay. When the Remote Healthy signal finally goes inactive, the backup Shelf Manager concludes that the active Shelf Manager is dead, and initiates a switchover.

If the Remote Healthy signal stays active, the backup Shelf Manager concludes that the communication link between the Shelf Managers is broken. In that case, no switchover is initiated; instead, the backup Shelf Manager repeatedly reinitializes itself and tries to establish a connection with the active Shelf Manager, until the communication link is restored. Reinitialization is achieved by rebooting the shelf management card and automatically restarting the Shelf Manager after the reboot. Special logic in the Shelf Manager guarantees that it does not try to become active at startup if the peer Shelf Manager is already active.

The Shelf Manager uses a watchdog timer to protect against becoming unresponsive due to infinite loops or other software bugs. If the watchdog timer on the active Shelf Manager triggers, that shelf management card is reset, causing the Remote Healthy signal on the backup shelf management card to become inactive, thus triggering a switchover.

After a switchover, the now-active Shelf Manager reinitalizes, activates the cached state information, and collects the necessary information from the IPM controllers on IPMB. This active Shelf Manager then exposes the ShMC device (address 20h) on IPMB, and assumes the IP address that was used for RMCP and other shelf-external interactions between the formerly active Shelf Manager and the System Manager. Since the RMCP session information is propagated from the active Shelf Manager to the backup Shelf Manager, RMCP sessions survive the switchover. For the System Manager using RMCP, the switchover is transparent.

After the switchover the formerly active Shelf Manager can cease to exist or reinitialize itself as the backup Shelf Manager. Reinitializing as the backup Shelf Manager requires rebooting the operating system on the formerly active shelf management card.

System Administrator Interface Options

Another major subsystem of the Shelf Manager implements the System Administrator Interface. The System Administrator is a logical concept that can include software as well as human operators in an operations center. The Shelf Manager provides two System Administrator interface options that provide different mechanisms of access to similar kinds of information and control regarding a shelf:

- IPMI Local Area Network (LAN) interface
- Command-line interface (CLI)

The IPMI LAN interface is used to maximize interoperability among independently implemented shelf products. This interface is required by the ATCA specification and supports IPMI messaging with the Shelf Manager through RMCP. A System Administrator that uses RMCP to communicate with shelves should be able to interact with any ATCA-compliant Shelf Manager. This low-level interface provides access to the IPMI aspects of a shelf, including the ability for the System Administrator to issue IPMI commands to IPM controllers in the shelf, using the Shelf Manager as a proxy.

RMCP is a standard network interface to an IPMI controller via LAN and is defined by the IPMI 1.5 specification.

The CLI provides a comprehensive set of textual commands that can be issued to the Shelf Manager through either a physical serial connection or a Telnet connection.

System Administration Tasks

Sun Netra CT900 server system administration typically includes installation, configuration, and administration tasks.

Oracle Solaris OS administration on the Sun Netra CT900 server, including adding Oracle Solaris user accounts, is performed by logging into the node board. Server administration is performed by logging into the shelf management card and using the shelf management card CLI. The shelf management card can be used as the single point of entry in the Sun Netra CT900 server for configuration and administration purposes.

System administration tasks are described in the following chapters.

Physical Address to Logical Slot Mapping

When viewing the Sun Netra CT900 server from the front, the physical slots are sequentially numbered from left to right. TABLE 1-4 shows the physical-to-logical-slot mapping and addresses. TABLE 1-5 lists the shelf's physical address and associated FRUs.

TABLE 1-4 Physical Address to Logical Slot Mapping

Physical Slot	**1**	**2**	**3**	**4**	**5**	**6**	**Switch 7**	**Switch 8**	**9**	**10**	**11**	**12**	**13**	**14**	**ShMM #1**	**ShMM #2**
Logical Slot	13	11	9	7	5	3	1	2	4	6	8	10	12	14	N/A	N/A
HW Address (hex)	4D	4B	49	47	45	43	41	42	44	46	48	4A	4C	4E	8	9
IPMB Address (hex)	9A	96	92	8E	8A	86	82	84	88	8C	90	94	98	9C	10	12

TABLE 1-5 Shelf's Physical Address and FRUs

	Address (hex)	**FRU #**
Shelf/Chassis	20	N/A
PPS BMC	20	0
Shelf EEPROM 1	20	1
Shelf EEPROM 2	20	2
Fan Tray 1	20	3
Fan Tray 2	20	4
Fan Tray 3	20	5
PEM 1	20	6
PEM 2	20	7
SAP	20	8

CHAPTER 2

Configuring Your System

The Sun Netra CT900 server is configured primarily through the active shelf management card command-line interface (CLI). The active shelf management card CLI enables system-level configuration, administration, and management that includes the node boards, the switch boards, the shelf management cards (ShMM), power entry modules (PEM), and fan trays. The shelf management card CLI interface can be used both locally and remotely.

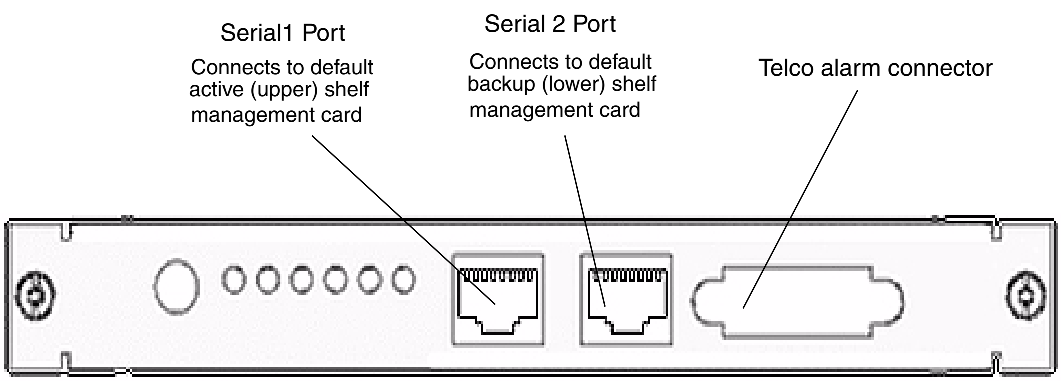
This chapter includes the following sections:

Accessing the Shelf Management Cards

The Shelf Manager runs on top of a specialized implementation of Linux. The lowest layer of Linux is the U-Boot firmware monitor. When you initially access either shelf management card (ShMM), you must do so over the serial port (console), using an ASCII terminal or the Tip program. Each shelf management card supports multiple sessions (Tip and Telnet connections) at once. The default TCP/IP address for the active shelf management card is `192.168.0.2`.

When connecting to a shelf management card (ShMM) through a serial port, connect a serial terminal or emulator to the one of the two serial ports on the front of the shelf alarm panel (SAP). Use the Serial 1 port to connect to the upper shelf management card (ShMM1), which is the default *active* shelf management card. The Serial 2 port connects to the default *backup* card (ShMM2). FIGURE 2-1 shows the location of the serial ports for the upper and lower shelf management cards. The terminal or modem setting should be set to `115200`, `N`, `8`, `1`.

FIGURE 2-1 Shelf Alarm Panel Connector



When you first access the shelf management card, log in as `root` and use the default password `sunct900`. This account is set to full authorization (permissions). This account can not be deleted. However, you should change the password on this account for security purposes, before your server is operational.

Use the Linux `passwd` command to change the root password as follows:

```
# passwd

Changing password for root
Enter the new password (minimum of 5, maximum of 8 characters)
Please use a combination of upper and lower case letters and
numbers.
Enter new password: xxxxxxx
Re-enter new password: xxxxxxx
Password changed.
#
```

Once you are logged on, use the `clia shmstatus` command to verify that you are logged onto the *active* shelf management card before continuing. If you are logged onto the standby shelf management card, you can use the `clia switchover` command to change the shelf management card to `Active`, or exit and log onto the active shelf management card. (See "`shmstatus`" on page 335 and "`switchover`" on page 338 for more information.)

The following sections provide information on configuring shelf management card Ethernet ports and setting up user accounts and passwords using the shelf management card CLI. For more information on using the shelf management card CLI, see Chapter 3.

Note – The term *shelf management card* as used in this manual refers to either the active or standby shelf management card, unless otherwise specified. In this manual, the prompt for both is shortened to `ShMM #`.

Setting Up U-Boot

On a power-on and reboot of the shelf management card (ShMM), the hardware starts executing the U-Boot firmware in Flash memory. The firmware performs basic initialization of the ShMM, and unless the user explicitly disables the Autoboot feature (thus forcing the firmware to switch to the maintenance user command interface), commences booting the Linux kernel. Linux is booted from the kernel and `root` file system images residing in Flash. U-Boot relocates the kernel image to RAM, sets up kernel parameters, and passes control to the kernel entry point.

U-Boot Interface

U-Boot is accessible via the serial port of the ShMM and requires configuration specific to the intended operational environment. When the ShMM is powered up, the following information is displayed on the console:

```
U-Boot 1.1.2 (Nov 11 2005 - 11:32:08)

CPU: Au1550 324 MHz, id: 0x02, rev: 0x00
Board: ShMM-500
S/N: 08004610
DRAM:  128 MB
Flash:  64 MB
In:    serial
Out:   serial
Err:   serial
Net:   Au1X00 ETHERNET
Hit any key to stop autoboot:  0
#
```

is the prompt allowing for user commands to be entered.

U-Boot Environment Variables

U-Boot includes a set of environment variables that should be configured prior to use. TABLE 2-1 describes the default set of environment variables.

TABLE 2-1 Default U-Boot Environment Variables

Environment Variable	Description
addmisc	Appends quiet, reliable_upgrade, and console settings to bootargs. This variable is normally not modified.
baudrate	Serial port baud rate, default=115200.
bootargs	Command line to be passed to the Linux kernel. Can contain references to other U-Boot environment variables, which are resolved at runtime. The default value is: `root=/dev/ram rw console=ttyS0,115200 reliable_upgrade=y`
bootcmd	U-Boot command executed to accomplish the auto-booting. Default is: `run ramargs addmisc; bootm $(kernel_start) $(rfs_start)`
bootdelay	Autoboot delay value, in seconds.
bootfile	Parameter that specifies what kernel image should be used by the net and nfs boot options.
console	Setting for the kernel and init script console port and baud rate. Default is console=ttyS0,115200.
ethaddr	MAC address of the primary on-chip Ethernet controller. The value of this variable is set automatically by U-Boot. This address is passed to the kernel Ethernet driver.
eth1addr	MAC address of the secondary Ethernet controller. The value of this variable is set automatically by U-Boot. This address is passed to the kernel Ethernet driver.
flash_reset	Instructs Linux to erase the Flash file systems (/etc and /var), restoring to factory default (y/n). The system startup script sets this variable back to n after the Flash erase. Default is n.
gatewayip	Default gateway IP address. This variable can be passed as a part of the kernel command line to automatically configure routing for the network interfaces.
hostname	Network host name. Default is shmm500.
io_config	Determines if the PSC controllers are configured for the dual-slave address configuration (y/n). Default is: y.

TABLE 2-1 Default U-Boot Environment Variables *(Continued)*

Environment Variable	Description
ipaddr	IP address used by the primary on-chip Ethernet interface. This variable is used to configure the network interface specified by ipdevice automatically if the rc_ifconfig variable is set to y. Note that the system startup script sets the least significant bit of this variable to the least significant bit of the Hardware Address for the ShMM carrier; that is, if the Hardware Address is an even value, the last bit in the IP address is set to 0, otherwise it is set to 1. This is done in the startup script /etc/netconfig to support coordinated IP address configurations on redundant ShMMs. To disable this functionality, simply remove the /etc/readhwaddr file.Default is 192.168.0.22
ip1addr	IP address used by the USB interface. This variable can be passed as a part of the kernel command line to automatically configure the corresponding kernel network interface. Default is 192.168.1.2
ip6F or 2A (Sun legacy)ddr	IP address used by the secondary Ethernet interface. This variable can be passed as a part of the kernel command line to automatically configure the corresponding kernel network interface. Default is 192.168.2.1
ipdevice	Device corresponding to ipaddr. Default is eth0.
ip1device	Device corresponding to ip1addr. Default is usb0.
ip2device	Device corresponding to ip6F or 2A (Sun legacy)ddr. Default is eth1.
kernel_start	The absolute starting address of the kernel image in Flash. This variable is set automatically by U-Boot during bootstrap.
logging	Specifies if messages log file should be maintained in ram or flash. Default is ram, which is the recommended option.
monitor_daemons	This variable enables the OpenHPI monitor when set to y. Default is n.
net	This variable can be used as a replacement for bootcmd as a means of booting a kernel and .rfs image from a TFTP server.
netmask	Network netmask, default=255.255.255.0
password_reset	Instructs Linux to restore factory default password (which is the sunct900 password for user root). Default is n.
post_normal	Determines the list of POST tests that are executed on each boot. If not set, compile-time default settings are used. The test names listed in a value of this variable are separated by space characters.
post_poweron	Determines the list of POST tests that are executed after power-on reset only (instead of on each boot). If not set, compile-time default settings are used. The test names listed in a value of this variable are separated by space characters.

TABLE 2-1 Default U-Boot Environment Variables *(Continued)*

Environment Variable	Description
quiet	Instructs the kernel upon boot not to print progress messages to the serial console. Default is quiet=quiet.
ramargs	Sets the kernel command line in the bootargs variable as appropriate for the root file system to be mounted from a ramdisk.
ramdisk	Specifies what .rfs image should be used by the net and nfs boot options. Default is sentry.rfs.
ramsize	Size of the system memory, in bytes. Default setting: calculated from the SDRAM configuration encoding in the build-time configuration block
rc_ifconfig	Allows the /etc/rc script to set up the IP address instead of getting the address from the shelfman file. Default is n (allow shelfman to set up IP addresses).
rc2	Specifies secondary RC script that is to be invoked. This is the carrier-specific startup script. Default is /etc/rc.acb3 or other appropriate script for given target platform.
reliable_upgrade	Determines if the reliable software upgrade procedure is enabled on the ShMM (y/n). Default setting: y. Setting this variable to n is not currently supported. If the variable is set to n, on the ShMM's next boot, it issues an error message and hangs.
rfs_start	The absolute starting address of the root file system image in Flash. This variable is set automatically by U-Boot during bootstrap.
rmcpaddr	Default IP address for the RMCP service. Default is 192.168.0.2.
serverip	IP address of the TFTP server. Default is 192.168.0.7.
start_rc2_daemons	Instructs the secondary startup script to start or not start the snmpd/boa and shelfman daemons after boot. Default is y.
time_server	Time server for synchronization at runtime. If this variable is not specified, time is extracted from the hardware clock at system startup. **Note -** When this variable is specified, the ip1device variable must be set to usb0 for proper synchronization.
time_proto	Protocol used to retrieve time from a network time server; possible values are ntp and rdate.
timezone	Local time zone in CCC*n* format where *n* is the offset from Greenwich Mean Time (GMT) and optionally negative, while CCC identifies the time zone. The default is UTC. For DST time zones, see "Setting the Time Zone for Daylight Saving" on page 66.

Displaying Environment Variable Values

Displaying the environmental variables can be done in either the U-Boot shell or the Linux shell.

Displaying Environment Variables in U-Boot Shell

To display the current environment variable values from within the U-Boot shell, use the `printenv` command. If no arguments is given, the `printenv` outputs all the environment variable values:

`printenv [`*variable_name*`]`

For example:

```
shmm500 printenv bootdelay
bootdelay=3
```

Displaying Environment Variables in Linux Shell

To display the current environment variable values from the Linux shell, use the `getenv` command. If no arguments is given, the `getenv` outputs all the environment variable values:

`getenv [`*variable_name*`]`

For example:

```
# getenv bootdelay
bootdelay=3
```

Assigning Values to Environment Variables

Assigning values to the environmental variables can be done in either the U-Boot shell or the Linux shell using the `setenv` command. When variables are set in the U-Boot shell, the `saveenv` command is needed to save the changes to the Flash.

To assign a value to an environment variable, use the format:

`setenv` *variable_name new_value*

For example:

```
setenv bootdelay 1
```

If you are setting variables in the U-Boot shell, save the changes by entering:

```
saveenv
```

Default Environment Variables for the Shelf Manager

When U-Boot is started for the first time, the following default environment variables are defined:

```
baudrate=115200
ipaddr=192.168.0.22
serverip=192.168.0.7
netmask=255.255.255.0
bootfile=sentry.kernel
ramdisk=sentry.rfs
rootpath=/rootfs
ramargs=setenv bootargs root=/dev/ram rw
net=tftpboot 80400000 $(bootfile); tftpboot 81200000 $(ramdisk); run ramargs
addmisc; bootm 80400000 81200000
nfsargs=setenv bootargs root=/dev/nfs rw nfsroot=$(serverip):$(rootpath)
addip=setenv bootargs $(bootargs)
ip=$(ipaddr):$(serverip):$(gatewayip):$(netmask):$(hostname):$(ipdevice)
addmisc=setenv bootargs $(bootargs) $(quiet) console=$(console),$(baudrate)
reliable_upgrade=$(reliable_upgrade)
nfs=tftpboot 80800000 $(bootfile); run nfsargs addip addmisc; bootm
bootcmd=run ramargs addmisc; bootm $(kernel_start) $(rfs_start)
console=ttyS0
quiet=quiet
ipdevice=eth0
ip1device=eth1
ip1addr=192.168.1.2
gatewayip=192.168.0.1
rmcpaddr=192.168.0.2
hostname=shmm500
flash_reset=n
password_reset=n
logging=ram
timezone=UTC
rc_ifconfig=n
start_rc2_daemons=y
reliable_upgrade=y
ethact=Au1X00 ETHERNET
stdin=serial
stdout=serial
stderr=serial
```

```
ethaddr=00:18:49:00:0c:42
eth1addr=00:18:49:00:0c:43
serial#=08009847
kernel_start=0xbe080000
rfs_start=0xbe440000
rc2=/etc/rc.acb3
bootdelay_old=3
post_normal=
bootdelay=3
```

Several of these environment variables must reconfigured with values that are appropriate to the network context in which the ShMMs are used.

Recommendations for Changing U-Boot Variables

Because changing certain U-Boot variables can cause system problems, administrators should be aware of the following recommendations before making any changes.

The following U-Boot variables can be changed:

```
baudrate
console
gatewayip
hostname
ipaddr
ip6F or 2A (Sun legacy)ddr
password_reset
netmask
rmcpaddr
serverip
time_proto
time_server
timezone
```

The following variables should only be changed for debugging by advanced users:

```
bootfile
boot_delay
flash_reset
logging
net
nfs
post_normal
post_poweron
quiet
ramdisk
start_rc2_daemons
rc_ifconfig
```

The following variables should not be changed:

```
addmisc
bootargs
bootcmd
ethaddr
eth1addr
io_config
ipdevice
ip1device
ip1addr
ip2device
kernel_start
ramargs
ramsize
rc2
reliable_upgrade
rfs_start
```

The following Ethernet configuration variables should not be changed:

```
rc_ifconfig = y
ipdevice = eth0
ip1device = usb0
ip1addr = 192.168.1.2
ip2device = eth1
```

Configuring Shelf Management Card Ethernet Ports

Each shelf management card uses two Ethernet ports that are connected to the redundant switch cards. Since RMCP is the only shelf external interface that is required by ATCA, the shelf's external Ethernet ports are referenced as the RMCP ports, though the other shelf's external interfaces (Telnet) are accessible via this port as well.

Once connected, you must be logged in to the shelf management card with a user account that has full permissions. You configure the ports with CLI commands, and then reboot the shelf management card for the changes to take effect.

Using the First Ethernet Interface

Since the RMCP Ethernet port is directly connected to the site network, the IP address should be set up appropriately for that network. For example, if the site uses the IP address range `192.168.0.x`, the RMCP Ethernet port should be set to a unique IP address within that range, such as `192.168.0.2`. In a redundant ShMM setup, only one ShMM (the active ShMM) has the RMCP IP address enabled on the RMCP Ethernet port. The backup ShMM assigns the same IP address to the RMCP Ethernet port, but only enables it when that ShMM assumes the active role. This way, the RMCP IP address maintains availability in a failover situation.

Assigning an Additional IP Address to the First Network Interface

In the default configuration, no IP address is assigned to the first network interface (and the ShMM is not accessible over the network) until the Shelf Manager starts and the RMCP IP address is assigned. However, it might be useful in some cases to assign an IP address to the RMCP network interface and have the ShMM accessible over the network as soon as the operating system is booted. In that case, it is also desirable for the RMCP IP address to coexist with the originally assigned IP address, rather than replacing it when the Shelf Manager is started.

To achieve this configuration, it is necessary to instruct the Shelf Manager to assign the RMCP IP address not to the first network adapter itself (`eth0`) but to its first alias (`eth0:1`). In that case, the initial IP address is assigned to the network adapter itself (`eth0`) during the start of the operating system. This initial assignment happens in the initialization script `/etc/rc`; it is accomplished by:

1. Enabling the U-Boot variable `rc_ifconfig` as follows:

```
setenv rc_ifconfig y
```

2. Assigning the original IP address to the U-Boot variable `ipaddr`. For example:

```
setenv ipaddr 192.168.1.240
```

3. Saving the changes before rebooting the ShMM.

```
U-Boot 1.1.4 (Feb 26 2009 - 06:28:24)
CPU: Au1550 324 MHz, id: 0x02, rev: 0x00
Board: ShMM-500
S/N: 08002600
DRAM:  (Samsung K4S511633F) 128 MB
Flash: 64 MB
In:    serial
Out:   serial
Err:   serial
ADM1060: configuration version 80
Net:   Au1X00 ETHERNET
Hit any key to stop autoboot:  3

shmm500 setenv rc_ifconig y
rc_ifconfig=y
shmm500
shmm500 setenv ipaddr 192.168.1.240
shmm500 printenv ipaddr
ipaddr=192.168.1.240
shmm500
shmm500 saveenv
Saving Environment to EEPROM...
shmm500
shmm500 boot
```

The IP address, 192.168.1.240, will show up as the `eth0` interface of the ShMM.

4. Changing the value of the `RMCP_NET_ADAPTER` in Shelf Manager configuration file `/etc/shelfman.conf` to `eth0:1`. For example, changing RMCP_NET_ADAPTER = $IPDEVICE to:

```
RMCP_NET_ADAPTER = eth0:1
```

Note – The RMCP address (the output of `[clia] getlanconfig 1 ip`) is now showing up in interface `eth0:1` instead of `eth0` due to this change.

In a redundant configuration, the U-Boot variable `ipaddr` is allowed to have the same value on both ShMMs. The actual initial IP address assigned to each of the two redundant ShMMs is based on the value of `ipaddr` but modified depending on the hardware address of the ShMM. The least significant bit of the IP address is set to the least significant bit of the hardware address. In the example above, the IP address would be `192.168.1.240` for the ShMM with an even hardware address, and `192.168.1.241` for the ShMM with an odd hardware address. This modification of the IP address can be turned off by removing the file `/etc/readhwaddr`.

RMCP Address Propagation

An optional feature of the Shelf Manager allows the backup ShMM to also be exposed on the external network with an IP address that is different from the RMCP IP address only in the least significant bit. The netmask and default gateway on the backup ShMM would be the same as on the active ShMM. For example, if the RMCP IP address is `192.168.0.2`, the backup ShMM would have the corresponding IP address `192.168.0.3`, with the same netmask and default gateway. To enable this feature, it is necessary to define the Shelf Manager configuration parameter `PROPAGATE_RMCP_ADDRESS` as `TRUE` in the Shelf Manager configuration file (`/etc/shelfman.conf`).

Using the Second Ethernet Interface

The second network interface connects the Shelf Manager with one of the ATCA network hub boards. Dual USB-based network interfaces are used for communication between the redundant Shelf Managers.

Using Dual USB Network Interfaces for Redundant Communication

On the ShMM, two additional network interfaces are implemented over the two USB connections. In this configuration, they always connect the two redundant Shelf Managers. These interfaces are named `usb0` and `usb1`. The interface `usb0` always exists, while the interface `usb1` exists only if the interface `usb0` is active on the peer

Shelf Manager (which means that the peer Shelf Manager is physically installed and running). Also, the interfaces are cross-connected: `usb0` on the first Shelf Manager is connected to `usb1` on the second Shelf Manager, and vice versa.

The Shelf Manager supports usage of the USB network interfaces for communication between the redundant Shelf Managers. To use this feature, it is necessary to define two redundancy network adapters in the Shelf Manager configuration file `/etc/shelfman.conf`, as follows:

```
REDUNDANCY_NET_ADAPTER = "usb0"
REDUNDANCY_NET_ADAPTER2 = "usb1"
```

One additional consideration relates to the definition of the subnet mask for the redundancy network interfaces. In the legacy case, when only one redundant network adapter is used, two different IP addresses are derived from the redundancy IP address specified in `/etc/shelfman.conf`. They are assigned to the two endpoints of the redundancy connection and differ only in the least significant bit.

However, when two redundancy network adapters are used, four different IP addresses are used, one for each of the endpoints (two endpoints on each of the two redundant Shelf Managers). To ensure proper operation, the two endpoints on the same Shelf Manager (`usb0` and `usb1`) must belong to different logical networks, while `usb0` on one Shelf Manager and `usb1` on the other Shelf Manager must belong to the same logical network. Based on these considerations, the two additional IP addresses are derived by toggling the least significant bit of the subnet mask in the redundancy IP address specified in `/etc/shelfman.conf`. The subnet mask must therefore be stricter than the default for the specified redundancy IP address class. If the subnet mask is not specified, it is set by default to `255.255.255.128`; this is also a recommended value for this parameter in `/etc/shelfman.conf` if USB network interfaces are used for redundancy.

Here is an example of deriving IP addresses for the USB network interfaces.

Suppose that the following definitions are in `/etc/shelfman.conf`:

```
REDUNDANCY_IP_ADDRESS = 192.168.1.2
REDUNDANCY_NETMASK = 255.255.255.128
```

On the ShMM with the *even* hardware address, the assignment of IP addresses would look like the following:

`usb0: 192.168.1.2` (no changes)
`usb1: 192.168.1.130` (toggling the least significant bit of the netmask)

On the ShMM with the *odd* hardware address, the assignment of IP addresses would looks like the following:

`usb0: 192.168.1.131` (toggling the least significant bit of the IP address and the least significant bit of the netmask)

`usb1: 192.168.1.3` (toggling the least significant bit of the IP address)

Changing the Default ShMM Network Parameters

Configuring a ShMM to work in a specific network environment requires changing the following network parameters:

- RMCP IP address
- RMCP GATEWAY address
- RMCP Netmask

Changing the RMCP network parameters is a multistep process. The U-Boot network environment variables must be updated, then the booted ACTIVE ShMM module network settings must be updated using the Shelf Manager CLI.

▼ To Change Default ShMM Network Parameters

1. **Attach a serial port console connection to the ShMM module.**

   This typically is 115200 baud, N/8/1. Reboot the ShMM carrier and press the space bar to interrupt the automatic boot process. You should see the following:

```
U-Boot 1.1.2 (Nov 11 2005 - 11:32:08)

CPU: Au1550 324 MHz, id: 0x02, rev: 0x00
Board: ShMM-500
S/N: 00 00 00 00 00 00 00 00 00 03 03 03
DRAM:  128 MB
Flash: 64 MB
In:    serial
Out:   serial
Err:   serial
Net:   Au1X00 ETHERNET
Hit any key to stop autoboot: <any_key> #Hit any key now
shmm500
```

2. **Echo the current network settings.**

```
shmm500 getenv rmcpaddr netmask gatewayip
rmcpaddr=192.168.0.44
netmask=255.255.255.0
gatewayip=192.168.0.1
shmm500
```

3. **Change the settings and commit to non-volatile storage.**

```
shmm500 setenv rmcpaddr 10.1.1.10
shmm500 setenv netmask 255.255.0.0
shmm500 setenv gatewayip 10.1.1.1
shmm500 saveenv
Un-Protected 1 sectors
Erasing sector  0 ... Erasing sector at 0x  800000
ok.
Saving Environment to EEPROM...done.
shmm500
```

4. **Boot the ShMM to full operational state and log in as user** `root`**.**

```
shmm500 reset

U-Boot 1.1.2 (Nov 11 2005 - 11:32:08)

CPU: Au1550 324 MHz, id: 0x02, rev: 0x00
Board: ShMM-500
S/N: 00 00 00 00 00 00 00 00 00 03 03 03
DRAM:  128 MB
Flash: 64 MB
In:    serial
Out:   serial
Err:   serial
Net:   Au1X00 ETHERNET
Hit any key to stop autoboot:  0
## Booting image at bfb00000 ...
   Image Name:   MIPS Linux-2.4.26
   Created:      2005-05-07  17:35:21 UTC
   Image Type:   MIPS Linux Kernel Image (gzip compressed)
   Data Size:    843144 Bytes = 823.4 kB
   Load Address: 80100000
   Entry Point:  802bc040
   Verifying Checksum ... OK
   Uncompressing Kernel Image ... OK
## Loading Ramdisk Image at bfc40000 ...
   Image Name:   sentry RFS Ramdisk Image
…
…
sentry login: root

BusyBox v0.60.5 (2005.05.07-17:27+0000) Built-in shell (msh)
#
```

5. **Allow the ShMM to start.**

Note – The settings that were changed in the U-Boot firmware are not necessarily propagated to the Linux environment. The reason is that the Shelf Manager must maintain its own copy of the network configuration data in order to manage failover situations.

If this is the first time the Shelf Manager has been booted, or if the flash devices have been reset to factory default prior to boot, then the Shelf Manager uses the network settings provided by U-Boot to set up this networking context (and thus the changes you made in U-Boot are propagated forward).

Otherwise, then the following steps are required to configure the network settings in the Shelf Manager context.

6. **Check to see if you are the active Shelf Manager.**

   You only need to make changes on the active shelf management card using the `cpld` command which also updates the backup with the network configuration changes through the redundancy interface. If you are not the active ShMM, then connect to the other ShMM device and continue to Step 7.

```
# cpld
CPLD word: E806
        0002h - Local Healthy
        0004h - Switchover Request Local
        0800h - Hot Swap Latch Open
        2000h - Active
        4000h - Interrupt Status
        8000h - Reboot Was Caused By Watchdog
#
```

7. **Get the current IP settings**

```
# 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: 206.25.139.28
IP Address Source: Static Address (Manually Configured) (0x01)
MAC Address: 00:50:c2:22:50:30
Subnet Mask: 0.0.0.0
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: 206.25.139.3
Default Gateway MAC Address: 00:00:00:00:00:00
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
#
```

8. **Change the IP settings as shown.**

```
# clia setlanconfig 1 ip 10.1.1.10

Pigeon Point Shelf Manager Command Line Interpreter
IP set successfully

# clia setlanconfig 1 subnet_mask 255.255.0.0

Pigeon Point Shelf Manager Command Line Interpreter
Subnet Mask set successfully

# clia setlanconfig 1 dft_gw_ip 10.1.1.1

Pigeon Point Shelf Manager Command Line Interpreter
Default Gateway Address set successfully
#
```

Assigning Different IP Addresses to Upper and Lower ShMMs

In the default configuration, the active ShMM can be accessed using the RMCP IP address and backup ShMM accessed using the RMCP IP1 address (which is assigned automatically). In some networks is might be desirable to identify the ShMMs using IP addresses that are different from the RMCP address. The advantage being that the active ShMM can be accessed using either its IP address or RMCP address, and backup ShMM can be accessed using its IP address.

To configure ShMMs with one IP address each and a single RMCP address that is shared (that is; only the active ShMM will have the RMCP address), do the following:

1. **Configure the upper ShMM.**

   a. **Change the U-Boot variables on the upper ShMM as shown. See** "Assigning Values to Environment Variables" on page 22 **for more information.**

```
setenv ipaddr IP-addr-for-upper-ShMM
setenv ipdevice eth0                    # Default DO NOT CHANGE
setenv ip1addr 192.168.1.2              # Default DO NOT CHANGE
setenv ip1device usb0                   # Default DO NOT CHANGE
setenv gatewayip gateway-ip-address
setenv netmask netmask                  # Usually 255.255.255.0
setenv rc_ifconfig y
setenv hostname hostname-of-lower-ShMM  # Optional
saveenv                                 # Optional - use in U-Boot
                                        # shell to save variables
```

   b. **Edit the** `/etc/shelfman.conf` **file on the upper ShMM:**

      i. **Change the value of the** `PROPAGATE_RMCP_ADDRESS` **variable to** `FALSE`**.**

```
PROPAGATE_RMCP_ADDRESS = FALSE
```

      ii. **Change the** `RMCP_NET_ADAPTER` **to** `eth0:1`**.**

   c. **Delete** `/etc/readhwaddr` **from active ShMM.**

2. **Configure the lower ShMM.**

   a. **Change the U-Boot variables on the lower ShMM as shown. See** "Assigning Values to Environment Variables" on page 22 **for more information.**

```
setenv ipaddr IP-addr-for-lower-ShMM   # Must be different from upper
                                       # ShMM
setenv ipdevice eth0                   # Default DO NOT CHANGE
setenv ip1addr 192.168.1.2             # Default DO NOT CHANGE
setenv ip1device usb0                  # Default DO NOT CHANGE
setenv gatewayip gateway-ip-address
setenv netmask netmask                 # Usually 255.255.255.0
setenv rc_ifconfig y
setenv hostname hostname-lower-ShMM    # Optional-different from
                                       # upper ShMM
saveenv                                # Optional - use in U-Boot
                                       # shell to save variables
```

b. **Edit the `/etc/shelfman.conf` file on the lower ShMM:**

   i. **Change the value of the `PROPAGATE_RMCP_ADDRESS` variable to `FALSE`.**

```
PROPAGATE_RMCP_ADDRESS = FALSE
```

   ii. **Change the `RMCP_NET_ADAPTER` to `eth0:1`.**

c. **Delete `/etc/readhwaddr` from active ShMM.**

3. **After rebooting both ShMMs, can verify the changes by using the commands `ifconfig eth0` and `ifconfig eth0:1` on both ShMMs.**

   On the backup ShMM, device eth0:1 will be undefined.

Setting Up the Shelf Manager Configuration File

The Shelf Manager configuration file (`shelfman.conf`) is located in the `/etc` directory. Each line in the file is either a comment line (starting with `#`) or a *name* = *value* pair, representing the assignment for the configuration parameter. The *name* and the *value* are separated with the equal sign (=).

The configuration parameter name is case insensitive. Each configuration parameter is one of the following types: Boolean, number, string, or IP address.

Format of the value conforms to the type of the configuration parameter as follows:

Boolean	A Boolean can be represented by either the strings `FALSE` or `TRUE`, or by their numerical representations of `0` or `1` respectively.
Number	A whole (possibly signed) numeric value; hexadecimal notation "`0x`..." is also supported.
String	A string, quoted or unquoted (double quotes " " are used). Quoted strings can contain blanks; unquoted strings are terminated by the first blank. The maximum string size is specified separately for each string-oriented configuration parameter.
IP-address	The IP address in decimal-dot (*xxx.xxx.xxx.xxx*) notation.

It is possible to specify a value of an environment variable as a configuration parameter value, using the notation `$envvar`; in that case, the value of the variable `envvar` is substituted when the configuration file is read. For example:

```
DEFAULT_RMCP_IP_ADDRESS = $IPADDR
```

After the Shelf Manager has been brought up for the first time, the IP addresses are stored with the IPMI LAN configuration parameters. The LAN configuration parameters can be accessed or modified via any of the RMCP or CLI shelf-external interfaces and take precedence over the `shelfman` configuration file when the Shelf Manager is restarted. This is to ensure the persistency of any modifications that are made to the LAN IP addresses and gateway via those interfaces. If however, the Shelf Manager IP connection record in the Shelf FRU Information contains an IP address, it take precedence over all other settings of the shelf-external or RMCP IP address. It is recommended that the Shelf FRU Information either not specify this address or set it to `0.0.0.0` to ensure that addresses can be controlled through the Shelf Manager configuration file and the IPMI LAN configuration parameters.

The following configuration parameters are currently supported:

TABLE 2-2 Shelf Manager Configuration Parameters

Name	Type	Default	Description
`2_X_SYSTEM`	Boolean	None	If specified, this parameter explicitly specifies the current system as AdvancedTCA (if `FALSE`). If not specified (`TRUE`), the choice of the system type is made automatically. It is not recommended to specify this parameter, unless it is necessary to override a wrong hardware detection algorithm for the system type.
`ACTIVATE_LOCAL_WITHOUT_SHELF_FRU`	Boolean	FALSE	If set to TRUE, both IPM controllers exposed by the active Shelf Manager (representing the physical and the logical Shelf Managers) are activated even if the Shelf FRU Information cannot be found. This option should be used with caution, because the power consumption of the payload of the physical Shelf Manager IPM controllers may potentially exceed the power capability of the corresponding slot in the shelf.
`ALARM_CUTOFF_TIMEOUT`	Number	`600` seconds (10 minutes)	The alarm cutoff timeout (time after which the alarm cutoff is deactivated), in seconds.
`ALLOW_ALL_COMMANDS_FROM_IPMB`	Boolean	`FALSE`	If set to `TRUE`, most of the commands allowed from the RMCP interface are allowed from IPMB-0 as well (except for session-related commands). For example, Cold Reset and user management commands are allowed from IPMB-0.

TABLE 2-2 Shelf Manager Configuration Parameters *(Continued)*

Name	Type	Default	Description
ALLOW_CHANGE_EVENT_RECEIVER	Boolean	TRUE	If set to TRUE, the Event receiver address for the Shelf Manager can be set to an address other than 20h, LUN 0. If set to FALSE, any attempt to change event receiver address for the Shelf Manager is rejected.
ALLOW_CLEARING_CRITICAL_ALARM	Boolean	FALSE	If set to TRUE, the critical alarm condition can be cleared by the CLI command `clia alarm clear`.
ALLOW_RESET_STANDALONE	Boolean	FALSE	If set to TRUE, the command `Cold Reset` is accepted even if the Shelf Manager does not have an available backup, and reboots the Shelf Manager. By default, the command `Cold Reset` is accepted only in a dual redundant configuration and causes a switchover.
ALTERNATE_CONTROLLER	Boolean	TRUE	Use alternate controller on the Shelf Manager with the `address =` *ShMM hardware address*.
ATCA_TESTER_COMPATIBILITY	Boolean	FALSE	This variable, if set, turns off event handling optimizations in the Shelf Manager, so that the Shelf Manager behavior is compatible with the Polaris ATCA Tester.
AUTO_SEND_MESSAGE	Boolean	TRUE	Automatically convert an RMCP request sent to a non-Shelf Manager IPMB address into a `Send Message` request directed to that address.
CARRIER	String(16)	PPS	The name of the specific carrier board on which the ShMM is installed.
CARRIER_OPTIONS	String (256)	SUNCT900	The carrier-specific options; defined separately for each supported carrier.
CONSOLE_LOGGING_ENABLED	Boolean	FALSE	Output log messages to the console on which the Shelf Manager was started.

TABLE 2-2 Shelf Manager Configuration Parameters *(Continued)*

Name	Type	Default	Description
COOLING_FAN_DECREASE_TIMEOUT	Number	0	The minimum timeout between successive decrements of the fan speed during operation of the cooling algorithm in Normal state. Should be a multiple of COOLING_POLL_TIMEOUT; if not, it is rounded up to the next multiple. If the parameter is omitted or set to 0, this timeout is equal to COOLING_POLL_TIMEOUT.
COOLING_FAN_INCREASE_TIMEOUT	Number	0	The minimum timeout between successive increments of the fan speed during operation of the cooling algorithm in Minor Alert state. Should be a multiple of COOLING_POLL_TIMEOUT; if not, it is rounded up to the next multiple. If the parameter is omitted or set to 0, this timeout is equal to COOLING_POLL_TIMEOUT.
COOLING_IGNORE_LOCAL_CONTROL	Boolean	FALSE	Do not use local control capabilities on fan devices; Shelf Manager explicitly manages the fan level.
COOLING_MANAGEMENT	String (64)	“” (undefined)	If specified, the name of the shared library that implements cooling management. The actual name of the library is libcooling_<*xxx*>.so, where <*xxx*> is the value of this configuration parameter. This library is dynamically loaded by the Shelf Manager and must be located in /var/bin or /lib.
COOLING_POLL_TIMEOUT	Number	30 seconds	The maximum time (in seconds) between successive invocations of the cooling monitoring and management thread.
CPLD_ACTIVE_WORKAROUND	Boolean	N/A	N/A
CTCA_FRU_RESET_TIMEOUT	Number	N/A	N/A
CTCA_HEALTHY_TIMEOUT	Number	N/A	N/A
CTCA_INITIAL_FAN_LEVEL	Number	N/A	N/A

TABLE 2-2 Shelf Manager Configuration Parameters *(Continued)*

Name	Type	Default	Description
DEFAULT_GATEWAY_IP_ADDRESS	IP-address	None	The default IP address used for the gateway for shelf-external (RMCP-based) communication, if the corresponding parameter is set to 0.0.0.0 in the IPMI LAN Configuration Parameters for channel 1. If a non-zero gateway IP address is provided in the LAN Configuration Parameters, the value provided in the Shelf Manager configuration file is ignored.
DEFAULT_GATEWAY_IP_ADDRESS2	IP-address	None	The default IP address used for the gateway for shelf-external (RMCP-based) communication on the second network interface, if the corresponding parameter is set to 0.0.0.0 in the IPMI LAN Configuration Parameters for channel 2. If a non-zero gateway IP address is provided in the LAN Configuration Parameters, the value provided in the Shelf Manager configuration file is ignored. Not defined by default. This parameter is used only if the value of USE_SECOND_CHANNEL is TRUE.
DEFAULT_RMCP_IP_ADDRESS	IP-address	None	The default IP address used for shelf-external (RMCP-based) communication; it is switched over between the redundant instances of the Shelf Manager. This IP address is used only if the corresponding parameter is set to 0.0.0.0 in the IPMI LAN Configuration Parameters for channel 1 and in the Shelf Manager IP Connection record in Shelf FRU Information. If a non-zero IP address is provided in the LAN Configuration Parameters and/or Shelf FRU Information, the value provided in the Shelf Manager configuration file is ignored.

TABLE 2-2 Shelf Manager Configuration Parameters *(Continued)*

Name	Type	Default	Description
DEFAULT_RMCP_IP_ADDRESS2	IP-address	None	The default IP address used for shelf-external (RMCP-based) communication on the second network interface; it is switched over between the redundant instances of the Shelf Manager. This IP address is used only if the corresponding parameter is set to 0.0.0.0 in the IPMI LAN Configuration Parameters for channel 2. If a non-zero IP address is provided in the LAN Configuration Parameters, the value provided in the Shelf Manager configuration file is ignored.
DEFAULT_RMCP_NETMASK	IP-address	Variable	The network mask for the network adapter used for RMCP communication. This mask is used only if the corresponding parameter is set to 0.0.0.0 in the IPMI LAN Configuration Parameters for channel 1 and in the Shelf Manager IP Connection record in Shelf FRU Information. The default value depends on the class of the default IP address used for the gateway for shelf-external (RMCP-based) communication. (see parameter `DEFAULT_RMCP_IP_ADDRESS`). For example, for an IP address of class C, this parameter is set to `255.255.255.0`.
DEFAULT_RMCP_NETMASK2	IP-address	Variable	The network mask for the second network adapter used for RMCP communication. This mask is used only if the corresponding parameter is set to 0.0.0.0 in the IPMI LAN Configuration Parameters for channel 2. The default value depends on the class of the default IP address used for the gateway for shelf-external (RMCP-based) communication. (see parameter `DEFAULT_RMCP_IP_ADDRESS2`). For example, for an IP address of class C, this parameter is set to `255.255.255.0`.

TABLE 2-2 Shelf Manager Configuration Parameters *(Continued)*

Name	Type	Default	Description
`DEFAULT_VLAN_ID`	Number	0	The default Virtual LAN ID used for the first LAN channel. This value is used only as the default value for the corresponding LAN configuration parameter; once LAN configuration parameters are defined and stored on the ShMM, the value provided in the Shelf Manager configuration file is ignored.
`DEFAULT_VLAN_ID2`	Number	0	The default Virtual LAN ID used for the second LAN channel. This value is used only as the default value for the corresponding LAN configuration parameter; once LAN configuration parameters are defined and stored on the ShMM, the value provided in the Shelf Manager configuration file is ignored.
`DETECT_DEADLOCKS`	Boolean	TRUE	This variable turns on the deadlock detection in CLI and RMCP server facilities in the Shelf Manager. The detection is based on the internal watchdog that must be periodically strobed by the threads serving CLI and RMCP requests. If one of the threads fails to strobe the internal watchdog, the actual watchdog does not get strobed, and ultimately the ShMM resets, initiating a failover to the backup Shelf Manager. In addition, lock data structures are periodically checked directly for the presence of a deadlock.
`DEVICE_POLL_TIMEOUT`	Number	`10` seconds	The time (in seconds) between successive polls of the IPMB devices by the Shelf Manager by the `Get Device ID` command.
`DHCP_FOR_RMCP_ONLY`	Boolean	FALSE	If this variable is set, only the RMCP IP addresses are assigned via the DHCP mechanism; private IP addresses of the ShMMs are not touched by DHCP.

TABLE 2-2 Shelf Manager Configuration Parameters *(Continued)*

Name	Type	Default	Description
`DHCP_SERVER_ADDRESS`	IP-address	None	This parameter is the IP address of the DHCP server; it applies only if the variable USE_DHCP is TRUE. If this parameter is omitted or set to 0.0.0.0, and the USE_DHCP variable is TRUE, the Shelf Manager accepts address information from any DHCP server that responds to its broadcast discovery request.
`ENABLE_DIRECT_SHELF_FRU_WRITE`	Boolean	FALSE	This variable controls whether the Shelf Manager allows direct writes to the Shelf FRU Info (FRU Device ID #254 on the IPM controller at 20h on IPMB-0) via the IPMI command Write FRU Data. By default, direct writes are prohibited (as mandated by PICMG 3.0 R2.0 ECN-002, which requires that Shelf FRU Info writes use a special locking protocol so that only one writer is active at once). Setting this variable to TRUE enables direct writes for compatibility with System Manager applications that rely on the pre-ECN-002 behavior.
`EXIT_IF_HEALTHY_LOST_IN_STANDALONE_MODE`	Boolean	FALSE	This variable defines what to do if the Shelf Manager runs without backup and detects a loss of the local Healthy bit. If this variable is TRUE, the Shelf Manager exits, the ShMM reboots and the Shelf Manager is restarted. If this variable is FALSE, the Shelf Manager sets the Healthy bit and continues operation,
`EXIT_IF_NO_SHELF_FRU`	Boolean	`FALSE`	If `TRUE`, the Shelf Manager exits (probably resetting the ShMM) if no Shelf FRU can be found.
`EXTERNAL_EVENT_HANDLER`	String (255)	“”	This is the path to an executable file (or a script file) on the Shelf Manager that performs local handling of events via PEF.

TABLE 2-2 Shelf Manager Configuration Parameters *(Continued)*

Name	Type	Default	Description
FAN_FULL_SPEED_DELAY	Number	0	The delay in seconds after Shelf Manager startup or after a switchover, during which the cooling algorithm does not check the number of present fan trays, giving the existing fan trays enough time to activate. This applies to the carriers where cooling management raises the fan speed to maximum if the actual number of fan trays in the shelf is fewer than what is specified in the Shelf Address Table.
FAN_LEVEL_STEP_DOWN	Number	1	The number of fan steps by which the fan speed is decreased during operation of the cooling algorithm in the Normal state. This parameter may be overridden by a ShMM carrier-specific cooling algorithm.
FAN_LEVEL_STEP_UP	Number	1	The number of fan steps by which the fan speed is increased during operation of the cooling algorithm in the Minor Alert state. This parameter may be overridden by a carrier-specific cooling algorithm.
HPDL	Boolean	FALSE	Turns on Hardware Platform Description Language (HPDL) support in the Shelf Manager. The carrier and chassis HPDL data and SDRs are taken from the FRU Information or from the files and are used to define the behavior of the platform, plus the number and types of managed FRUs and sensors.
HPDL_ON_SUBSIDIARY_FRUS	Boolean	FALSE	Turns on support of HPDL information stored on subsidiary FRUs. If TRUE, the Shelf Manager looks for HPDL data and SDRs in the FRU Information of its subsidiary FRUs. If these data are found for a specific FRU, they are used to substitute definitions for that FRU from the carrier or chassis HPDL data and/or SDRs.

TABLE 2-2 Shelf Manager Configuration Parameters *(Continued)*

Name	Type	Default	Description
IGNORE_FAILED_DIRECTED_ POWER_DOWN	Boolean	TRUE	This parameter tells the Shelf Manager to ignore board power down failure in case of Critical Alert and not to power down all boards in chassis in this case. This parameter also instructs the Shelf Manager to ignore board power level decrease failure in case of Major Alert and not to decrease power level for all boards in chassis in this case. If this parameter is FALSE, all boards in chassis are powered down if board(s) that caused Critical Alert cooling state fail to power off; also power level for all boards in chassis is decreased if decreasing power level fails for board(s) that caused the Major Alert cooling state.
INITIAL_FAN_LEVEL	Number	5	The initial fan level that the Shelf Manager applies to fan trays. Usually fan level values are in 0..15 range, where 0 is the slowest, and 15 is the fastest possible fan speed.
INITIAL_SLOW_LINK_DELAY	Number	100	The initial delay, in seconds, before the Shelf Manager starts testing the integrity of the physical network link between the Shelf Manager and the System Manager (the RMCP link; see the description of the configuration parameter SWITCHOVER_ON_BROKEN_LINK). A non-zero delay can be used to accommodate slow network links that need significant time to initialize after shelf power up.
INNER_SEQUENCE_NUMBER IN_SEND_MSG_RESPONSE	Boolean	TRUE	This variable controls which sequence number is used in the response to a Send Message command bridged from LAN to IPMB. If TRUE, the sequence number of the command encapsulated in the Send Message request is used. If FALSE, the sequence number of the Send Message request itself is used. According to a clarification being proposed for the latest version of the PICMG 3.0 specification, the first variant is correct, while the Shelf Manager historically used the second variant. The old behavior can be restored by setting to the value to FALSE.

TABLE 2-2 Shelf Manager Configuration Parameters *(Continued)*

Name	Type	Default	Description
`IPMB_ADDRESS`	Number	`0`	The IPMB address of the Shelf Manager, overriding the hardware address. A value of 0 causes the Shelf Manager to read the hardware address from hardware and set IPMB address to hardware address * 2.
`IPMB_LINK_ISOLATION_TIMEOUT`	Number	`-1`	In radial shelves, if an IPMB link is disabled due to the isolation algorithm, the link is automatically enabled after this time interval (in seconds). -1 (the default) indicates "forever".
`IPMB_RETRIES`	Number	`3`	The number of attempts to resend an IPMB request before finally giving up, if no response is received to the request.
`IPMB_RETRY_TIMEOUT`	Number	4 seconds	The amount of time the Shelf Manager waits for a response after sending an IPMB request, before retrying the request.
`IPMB_RETRY_TIMEOUT_MSEC`	Number	0	The millisecond part of the retry timeout value. If the retry timeout is less than a second, this configuration variable contains the actual timeout, while the value of the configuration variable `IPMB_RETRY_TIMEOUT` is 0.
`IPMC_PRESERVE ON_REVISION_CHANGE`	Boolean	TRUE	Setting this variable to TRUE preserves the Shelf Manager's identification of an IPM controller after a firmware upgrade if only the Firmware Revision and/or Auxiliary Firmware Revision information is changed. If the variable is set FALSE, any change in Get Device ID response data during the Shelf Manager's regular polls is considered to signal the presence of a different IPM controller.
`ISOLATE_MUX_ADDRESS`	Number	0x70	The 7-bit I^2C multiplexer address (on platforms where `SHMM_GPIO8` is used to control access from the Shelf Manager to the multiplexer on the master-only I^2C bus).

TABLE 2-2 Shelf Manager Configuration Parameters *(Continued)*

Name	Type	Default	Description
ISOLATE_MUX_IGNORE_COUNT	Number	10	This parameter instructs the isolation algorithm to skip this number of accesses to a faulty bus before trying to enable it (on the platforms where SHMM_GPIO8 is used to control access from the Shelf Manager to the multiplexer on the masteronly I^2C bus).
ISOLATE_MUX_ON_GPIO8	Boolean	FALSE	Must be set to TRUE for the platforms where SHMM_GPIO8 is used to control access from the Shelf Manager to the multiplexer on the master-only I^2C bus.
LOCAL_SHELF_FRU	Boolean	TRUE	Create a local FRU 1 on the Shelf Manager that exposes the Shelf FRU Information (obtained from the file /var/nvdata/shelf_fru_info).
M7_TIMEOUT	Number	-1 (second) or 600 in ShMM version 2.4.9-R3U2 and above.	The maximum time (in seconds) for a FRU to stay in M7 state; after the expiration of this time, the FRU automatically transitions to M0. -1 (the default) stands for *forever*. Setting this parameter to 0 completely prevents FRUs from going into state M7.
MAX_ALERT_POLICIES	Number	64	The maximum number of PEF Alert Policies available.
MAX_ALERT_STRINGS	Number	64	The maximum number of PEF Alert Strings available.
MAX_ATCA_FANLEVEL	Number	8	Determines maximum fan level to use when the NEBS Strategy is in effect.
MAX_ATCA_TEMP	Number	25	Determines the upper limit of intake air temperature for the NEBS strategy to be effective.
MAX_CMD_SUBSCRIBER_IDLE_TIME	Number	60	This parameter defines the maximum timeout (in seconds) for command subscriber to read notifications from socket. If this timeout is exceeded the subscriber is considered dead and is automatically unregistered.
MAX_DEFERRED_ALERTS	Number	32	The maximum number of outstanding PEF alerts.
MAX_EVENT_FILTERS	Number	64	The maximum number of PEF event filters available.

TABLE 2-2 Shelf Manager Configuration Parameters *(Continued)*

Name	Type	Default	Description
MAX_EVENT_SUBSCRIBERS	Number	64	The maximum number of entities that can simultaneously subscribe to receive event notifications from the Shelf Manager.
MAX_EVENT_SUBSCRIBER_IDLE_TIME	Number	60 seconds	The maximum timeout for an event subscriber, in seconds, between the moment when an event arrives and the moment when the subscriber retrieves this event from the Shelf Manager. If this timeout is exceeded, the subscriber is considered dead and is automatically unregistered.
MAX_INCOMING_IPMB_REQUESTS	Number	128	The size of the internal Shelf Manager queue for incoming IPMB requests. Incoming IPMB requests are stored in this queue before processing.
MAX_NODE_BUSY_RETRANSMISSIONS	Number	255	The maximum number of retransmissions of an IPMB command if the receiver always returns the completion code Node Busy in response.
MAX_OEM_FILTERS	Number	16	The maximum number of special PEF event filters for handling OEM-type System Event Log (SEL) entries.
MAX_PENDING_CMD_NOTIFICATIONS	Number	32	This parameter specifies the maximum number of command pending notifications.
MAX_PENDING_EVENT_NOTIFICATIONS	Number	1024	The maximum number of outstanding event notifications for each active subscriber.
MAX_PENDING_IPMB_REQUESTS	Number	192	The maximum number of pending IPMB requests awaiting response.
MAX_SEL_ENTRIES	Number	1024	The maximum number of entries in the system event log (SEL).
MAX_SESSIONS	Number	32	The maximum number of simultaneous IPMI sessions.
MAX_USERS	Number	32	The maximum number of IPMI users.
MICRO_TCA	Boolean	FALSE	If TRUE, the Shelf Manager operates as a MicroTCA Shelf Manager (the second RMCP channel is used for interaction with Carrier Managers).

TABLE 2-2 Shelf Manager Configuration Parameters *(Continued)*

Name	Type	Default	Description
MIN_FAN_LEVEL	Number	1	The minimum fan level; the cooling management code can not reduce the fan level of any fan below this value when controlling the fan level automatically.
MIN_SHELF_FRUS	Number	2	The minimum number of Shelf FRUs in the shelf that the Shelf Manager must detect to start up successfully.
NORMAL_STABLE_TIME	Number	3600	The time in seconds for which the Shelf Manager preserves the minimum fan level dynamically found in Normal mode (that is, the minimum fan level that does not cause thermal alerts). After this time expires, the cooling algorithm decreases the minimum fan level, if possible, to allow the shelf to decrease the fan level if the thermal load in it has also decreased.
NOTIFY_POLL_PERIOD_NSEC	Number	1000000	This parameter sets the time (in nanoseconds) for subsequent checks in the notification thread for new notifications in the internal queue. The default value is 1000000 (1 millisecond).
PET_FORMAT	Number	0	Specifies the format of the Platform Event Traps that are sent by the Shelf Manager as the Alert action initiated by event processing in the Platform Event Filtering facility. The values are defined as follows: • 0 the default IPMI format defined by IPMI Platform Event Trap Format v1.0 specification. • 1 plain text format; all the event details are sent as plain ASCII text in a single variable. • 2 multi-variable format; each event field is encoded as a separate variable.
PHYSICAL_SENSORS	Boolean	TRUE	Create IPMI sensors based on physical sensors hosted by ADM1026 and LM75 chips.
POWER_UNLISTED_FRUS	Boolean	TRUE	Allow the FRUs not listed in the power management table in the Shelf FRU Information to be activated and powered up.

TABLE 2-2 Shelf Manager Configuration Parameters *(Continued)*

Name	Type	Default	Description
PROPAGATE_RMCP_ADDRESS	Boolean	FALSE	If TRUE, the active Shelf Manager propagates the RMCP IP address to the backup Shelf Manager, which configures the network interface specified by the RMCP_NET_ADAPTER variable using that IP address, but with the least significant bit inverted.
REDUNDANCY_ENABLED	Boolean	TRUE	Run the Shelf Manager in redundant mode. Must be set to TRUE on systems with two shelf management cards.
REDUNDANCY_NET_ADAPTER	String(16)	usb0	The name of the network adapter used for communication between redundant instances of the Shelf Manager.
REDUNDANCY_NET_ADAPTER2	String(16)	usb1	The name of the second network adapter used for communication between redundant instances of the Shelf Manager (if the dual-USB network interface is used for this purpose).
REDUNDANCY_NETMASK	Number	0	The netmask to assign to the redundancy IP address; by default (if 0), the netmask is determined automatically from the class of the IP address.
REDUNDANCY_PORT	Number	1040	The TCP port used for interactions between redundant instances of the Shelf Manager.
REDUNDANT_IP_ADDRESS	IP-address	None	The IP address used for redundant communications. This address actually specifies a pair of IP addresses that differ only in the least significant bit. They are assigned to redundant Shelf Managers according to their hardware addresses.
RESERVATION_RETRIES	Number	10	The maximum number of times the Shelf Manager retries the Reserve Device SDR command.
RMCP_NET_ADAPTER	String(16)	eth0	The name of the network adapter used for RMCP-based communication.
RMCP_NET_ADAPTER2	String(16)	None	The name of the alternate network adapter used for RMCP-based communications, if cross-connect links are supported by the hardware.

TABLE 2-2 Shelf Manager Configuration Parameters *(Continued)*

Name	Type	Default	Description
RMCP_WITHOUT_SHELF_FRU	Boolean	FALSE	RMCP is available in the absence of the Shelf FRU only if this configuration parameter is defined and set to TRUE.
SDR_READ_RETRIES	Number	3	The maximum number of times the Shelf Manager retries the `Read Device SDR` command.
SEL_HIGH_WATERMARK	Number	0	The *high watermark* for the algorithm that controls automatic purging of the SEL; if the actual percentage of free entries in the SEL falls below this value, or the SEL overflows, the Shelf Manager starts a thread that purges old records from the SEL in order of decreasing age.
SEL_LOW_WATERMARK	Number	0	The *low watermark* for the algorithm that controls automatic purging of the SEL; if the thread that purges old records from the SEL starts, it will purge records until the percentage of occupied entries in the SEL falls below this value.
SENSOR_POLL_INTERVAL	Number	1	The time (in seconds) between successive polls of local Shelf Manager sensors by the Shelf Manager.
SESSION_SEQUENCE_WINDOW	Number	128	This is the window of acceptable RMCP sequence numbers; the wider this window, the more tolerant is the Shelf Manager to RMCP packets being dropped during transfer. If the difference in the sequence numbers of a received packet and the previous packet exceeds the window size, the Shelf Manager closes the RMCP session as a corrupted one.
SHELF_FRU_IN_EEPROM	Boolean	TRUE	If `TRUE`, the Shelf FRU information is retrieved from EEPROMs on the backplane in a carrier-specific way; if `FALSE`, the Shelf FRU information is obtained from a file on the Flash file system.
SHELF_FRU_IPMB_SOURCE1	Number	0	If defined (non-zero), specifies the IPMB address of the first designated source of Shelf FRU Information in the shelf. (Shelf FRU is located at FRU 1.) If this value is defined, the search for the Shelf FRU on the IPMB is limited to the designated sources only.

TABLE 2-2 Shelf Manager Configuration Parameters *(Continued)*

Name	Type	Default	Description
SHELF_FRU_IPMB_SOURCE2	Number	0	If defined (non-zero), specifies the IPMB address of the second designated source of Shelf FRU Information in the shelf. (Shelf FRU is located at FRU 1.) If this value is defined, the search for the Shelf FRU on the IPMB is limited to the designated sources only.
SHELF_FRU_TIMEOUT	Number	5 seconds	The time interval during initialization that the Shelf Manager waits for Shelf FRU Information devices to be detected.
SHORT_SEND_MSG_RESPONSE	Boolean	TRUE	Determines the type of the Send Message response provided by the Shelf Manager: required by the PICMG 3.0 ECR (if TRUE) or compatible with the previous versions of the Shelf Manager (if FALSE).
SWAPPED_CROSS_CONNECTS	Boolean	FALSE	Swaps the names of network adapters used for cross-connects on the ShMM with the odd hardware address.
SWITCHOVER_ON_HANDLE_OPEN	Boolean	FALSE	If TRUE, switchover-related behavior of the Shelf Manager is affected by the state of its hot-swap handle, as follows: • If the active Shelf Manager goes to the state M1 due to its hot swap-handle being open, a switchover to the backup Shelf Manager is initiated; • If the active Shelf Manager goes to the state M5 and there is no available backup Shelf Manager, the active Shelf Manager is not deactivated and stays in M5 indefinitely.
SWITCHOVER_TIMEOUT_ON_BROKEN_LINK	Number	10 seconds	This parameter affects when or whether the Shelf Manager initiates a switchover when the physical network link between the Shelf Manager and the System Manager (the RMCP link) is broken. If the link remains broken for at least the number of seconds given in this parameter, a switchover takes place; if the link is restored during this time-out period, no switchover takes place. If the value of this parameter is -1, no automatic switchovers take place on broken RMCP links.
SYSLOG_LOGGING_ENABLED	Boolean	TRUE	Output log messages to the system log.

TABLE 2-2 Shelf Manager Configuration Parameters *(Continued)*

Name	Type	Default	Description
SYSTEM_MANAGER_TRUNCATES_SEL	Boolean	FALSE	If TRUE, the Shelf Manager algorithm for truncating the SEL automatically is disabled; the System Manager is responsible for truncating the SEL by monitoring the value of the sensor SEL State of the type Event Logging Disabled on the Shelf Manager, and removing events from the SEL by sending the `Delete SEL Entry` command to the Shelf Manager.
TACHOMETER_THRESHOLD_UPDATE_DELAY	Number	15	Controls the delay between setting the fan level and updating the fan tachometer thresholds in shelves where dynamic fan tachometer thresholds are supported.
TASKLET_RETRIES	Number	3	The number of times each Shelf Manager tasklet (activation, deactivation, getting information) retries before finally giving up.
TURBO_MODE_MIN_FAN_FAILURES	Number	1	This parameter specifies the number of tachometer underspeed faults (crossing Major or Critical thresholds) cause the remaining fans to go to the TURBO mode (that is, run at full speed) in the default HPDL cooling management algorithm. If the shelf implements zoned cooling, the specified number of tachometer faults is counted relative to each zone and setting the TURBO mode affects only the fan trays that participate in cooling of the corresponding zone.
UNCONDITIONAL_SDR_REREAD_ON_VERSION_CHANGE	Boolean	FALSE	If its value is TRUE, the Shelf Manager unconditionally re-reads SDRs from an IPM controller when it receives a Version Change event from that controller, even if the Sensor Population Change Indicator in the "Get Device SDR Info" response does not change. This is done for the benefit of ATCA boards that are not fully IPMI-compliant in this respect. The default value of this variable is FALSE.

TABLE 2-2 Shelf Manager Configuration Parameters *(Continued)*

Name	Type	Default	Description
USE_DHCP	Boolean	FALSE	Requests assignment of RMCP-accessible and private IP addresses for the Shelf Manager from a DHCP server; the configuration parameter PREFERRED_DHCP_SERVER can be used to specify the IP address of the preferred DHCP server.
USE_SECOND_CHANNEL	Boolean	FALSE	This parameter applies only if two network interfaces on the ShMM are used for RMCP communication. If TRUE, the two network interfaces on the ShMM are used in parallel mode; if FALSE, they are used in redundant mode.
VERBOSITY	Number	7	The Shelf Manager verbosity level. The default verbosity level is 7, errors, warnings and informational messages. See "Verbosity Level Description" on page 63 for more information.
VERBOSITY_CONSOLE	Number	N/A	The Shelf Manager verbosity level for console output.
VERIFY_SHELF_FRU_CHECKSUM	Boolean	TRUE	Enable verification of checksums in Shelf FRU Information records; if set to FALSE, Shelf Manager ignores checksums.
WATCHDOG_ENABLED	Boolean	TRUE	Use the hardware watchdog timer supported by the CPLD.

By default, the Configuration file variables are used automatically when the ShMM is brought up for the first time. The default configuration file imports the following environment variables set by U-Boot:

$IPADDR	Default RMCP IP Address
$IPDEVICE	Default RMCP network adapter
$IP1ADDR	Default Redundant IP Address
$IP1DEVICE	Default Redundant network adapter
$GATEWAY	Default gateway used for RMCP communication

The environment variables $CARRIER and $CARRIER_OPTIONS are set by the secondary RC script. The name of this carrier-specific startup script is defined by the U-Boot environment variable rc2. The shelf manager can be reset to factory default parameter values if needed (see "Re-initializing the File System" on page 124).

A copy of default configuration file is shown in EXAMPLE 2-1:

EXAMPLE 2-1 Default `shelfman.conf` File

```
# /etc/shelfman.conf
#
# This is the PPS Shelf Manager configuration file.
# Copyright (c) 2005 Pigeon Point Systems.
# All rights reserved.
#

# CARRIER: This parameter is the name of the carrier-specific module to use.
#   Default is PPS.
CARRIER = $CARRIER

# CARRIER_OPTIONS: This parameter specifies the carrier-specific options.
#   Default is an empty string.
CARRIER_OPTIONS = $CARRIER_OPTIONS

# ALTERNATE_CONTROLLER: This parameter of boolean type specifies whether to
#   use the alternate controller on the Shelf Manager with the address
#   equal to the ShM hardware address. Default is TRUE.
#
ALTERNATE_CONTROLLER = TRUE

# ALLOW_CLEARING_CRITICAL_ALARM: This parameter of boolean type enables the
#   ability to clear the critical alarm condition without the alarm cutoff
#   button. Default is FALSE.
#
ALLOW_CLEARING_CRITICAL_ALARM = FALSE

# ALARM_CUTOFF_TIMEOUT: This parameter specifies the time interval in
# seconds for the Shelf Manager to hold the Alarm Cutoff state. Default
# interval is 600 seconds.
#
ALARM_CUTOFF_TIMEOUT = 600

# COOLING_IGNORE_LOCAL_CONTROL: This parameter of boolean type specifies
#   whether the Shelf Manager should use local control capabilities on fan
#   devices i.e. whether the Shelf Manager should explicitly manage fan
#   levels or not. Default is FALSE.
#
COOLING_IGNORE_LOCAL_CONTROL = FALSE

# COOLING_POLL_TIMEOUT: This parameter specifies the maximum time (in
# interval is between subsequent invocations of the cooling monitoring and
# management facility. Default is 30 seconds.
```

EXAMPLE 2-1 Default `shelfman.conf` File *(Continued)*

```
#
COOLING_POLL_TIMEOUT = 30

# DEVICE_POLL_TIMEOUT: This parameter specifies the time (in seconds)
# between subsequent polls of the IPMB-0 devices by the Shelf Manager via
# sending the "Get Device ID" command to them. Default is 10 seconds.
#
DEVICE_POLL_TIMEOUT = 10

# IPMB_ADDRESS: This parameter defines the IPMB address of the Shelf
# Manager's slot. This parameter overrides the hardware address. The default
# value of0 forces the Shelf Manager to use the hardware address and set its
# IPMB address to hardware address * 2.
#
# IPMB_ADDRESS = 0

# IPMB_RETRIES: This parameter is the number of attempts to re-send an IPMB
#    request before finally giving up, if no response is received to this
#    request. Default is 3.
#
IPMB_RETRIES = 3

# IPMB_RETRY_TIMEOUT: This parameter is the amount of time (in seconds) the
#   Shelf Manager waits for a response after sending an IPMB request, before
#    retrying it. Default is 4 seconds.
#
IPMB_RETRY_TIMEOUT = 4

# M7_TIMEOUT: This parameter specifies the maximum time interval (in
#   seconds for a FRU to stay in M7 state. After the expiration of this time
# the FRU automatically transitions into the M0 state. Default is -1 which
# means "forever". Setting this parameter to 0 completely prevents FRUs from
# going into the M7 state.
#
M7_TIMEOUT = -1

# MAX_ALERT_POLICIES: This parameter specifies the number of available
entries
#    in the PEF Alert Policy table. Default is 64.
#
MAX_ALERT_POLICIES = 64

# MAX_ALERT_STRINGS: This parameter specifies the number of available
entries
#    in the PEF Alert String table. Default is 64.
```

EXAMPLE 2-1 Default `shelfman.conf` File *(Continued)*

```
#
MAX_ALERT_STRINGS = 64

# MAX_DEFERRED_ALERTS: This parameter sets the maximum number of outstanding
#    PEF alerts. Default is 32.
#
MAX_DEFERRED_ALERTS = 32

# MAX_EVENT_FILTERS: This parameter specifies the number of available
entries
#    in the PEF Event Filter table.
#
MAX_EVENT_FILTERS = 64

# MAX_OEM_FILTERS: This parameter specifies the number of available entries
#    in the PEF OEM Event Filter table. Default is 16.
#
MAX_OEM_FILTERS = 16

# MAX_PENDING_IPMB_REQUESTS: The parameter sets the maximum number of
#    pending IPMB requests awaiting response. Default is 192.
#
MAX_PENDING_IPMB_REQUESTS = 192

# MAX_SEL_ENTRIES: The parameter defines the SEL capacity in records.
# Default is 1024.
#
MAX_SEL_ENTRIES = 1024

# SEL_HIGH_WATERMARK: This parameter is the "high watermark" for the
algorithm
# algorithm that controls automatic SEL purging. The purging process will
# start when the actual percentage of free entries in SEL falls below this
# value or the SEL is full. During the purge the oldest SEL records are
# removed according their timestamp. Default is 10 percent i.e. start
# purging when SEL is full.
#
SEL_HIGH_WATERMARK = 10

# SEL_LOW_WATERMARK: This parameter is the "low watermark" for the algorithm
#    that controls automatic SEL purging. When the SEL purging thread starts
#    it removes records one by one until the percentage of remaining occupied
#    entries in the SEL falls below this value. Default is 50 percent.
#
SEL_LOW_WATERMARK = 50
```

EXAMPLE 2-1 Default `shelfman.conf` File *(Continued)*

```
# MAX_SESSIONS: This parameter specifies the maximum number of simultaneous
#    IPMI sessions. Default 32.
#
MAX_SESSIONS = 32

# MAX_USERS: This parameter specifies the maximum number of IPMI users.
#    Default is 32.
#
MAX_USERS = 32

# INITIAL_FAN_LEVEL: This parameter specifies the initial fan level that the
# Shelf Manager applies to fan trays. Usually fan levels values are in
# 0..15 range where 0 is the slowest, and 15 is the fastest possible fan
# speed. This parameter has an alias CTCA_INITIAL_FAN_LEVEL for CompactPCI
# systems. Default is 5.
#
INITIAL_FAN_LEVEL = 5

# MIN_FAN_LEVEL: This parameter specifies the minimal fan level that can be
# set by the Cooling Management. Default is 0.
#
MIN_FAN_LEVEL = 1

# PHYSICAL_SENSORS: This parameter of boolean type specifies whether the
# Shelf Manager should create IPMI sensors based on physical sensors hosted
# by ADM1026 and LM75. Default is TRUE.
#
PHYSICAL_SENSORS = TRUE

# POWER_UNLISTED_FRUS: This parameter of boolean type specifies whether the
# Shelf Manager should power up and activate FRU devices that are not listed
# in the Power Management table of the Shelf FRU Information. Default is
# TRUE.
#
POWER_UNLISTED_FRUS = TRUE

# AUTO_SEND_MESSAGE: This parameter of boolean type specifies whether to
# auto-convert RMCP requests targeting a non-ShM IPMB address into "Send
# Message" requests directed to that address. Default is TRUE.
#
AUTO_SEND_MESSAGE = TRUE

# SHORT_SEND_MSG_RESPONSE: This parameter of boolean type determines the
# type of response on the Send Message command provided by the Shelf
```

EXAMPLE 2-1 Default `shelfman.conf` File *(Continued)*

```
# Manager:required by the PICMG 3.0 R1.0 ECN-001 if TRUE or compatible with
# previous versions of the Shelf Manager if FALSE. Default is TRUE.
#
SHORT_SEND_MSG_RESPONSE = TRUE

# SDR_READ_RETRIES: This parameter sets the number of times the Shelf
# Manager retries the "Read Device SDR" command. Default is 3.
#
SDR_READ_RETRIES = 3

# RESERVATION_RETRIES: This parameter specifies the number of times the
# Shelf Manager retries the "Reserve Device SDR" command. Default is 10.
#
RESERVATION_RETRIES = 10

# TASKLET_RETRIES: This parameter specifies the number of times each Shelf
# Manager tasklet (activation, deactivation, getting information) is
# retried before finally giving up. The default is 3.
#
TASKLET_RETRIES = 3

# SHELF_FRU_IN_EEPROM: This parameter of boolean type tells the Shelf
# Manager if it should use SEEPROMs as the Shelf FRU Info storage. If set
# to FALSE the "/var/nvdata/shelf_fru_info" file contents are used. Default
# is TRUE.
#
SHELF_FRU_IN_EEPROM = TRUE

# LOCAL_SHELF_FRU: This parameter of boolean type specifies whether the
# Shelf Manager should create a local FRU#1 that will expose the Shelf FRU
# Info (obtained from the "/var/nvdata/shelf_fru_info" file). If the Shelf
# FRU Info is acquired from EEPROM as a result of the SHELF_FRU_IN_EEPROM
# set to TRUE then this parameter ignored. Default is TRUE.
#
LOCAL_SHELF_FRU = TRUE

# SHELF_FRU_TIMEOUT: This parameter specifies the time interval (in seconds)
# during which the Shelf Manager detects and reads the Shelf FRU Information
# source devices at initial startup. Default is 15 seconds.
#
SHELF_FRU_TIMEOUT = 15

# MIN_SHELF_FRUS: This parameter specifies the minimum number of valid and
# equal Shelf FRU Information instances that must be found to determine the
# true Shelf FRU Information. Default is 2.
```

EXAMPLE 2-1 Default `shelfman.conf` File *(Continued)*

```
#
MIN_SHELF_FRUS = 2

# EXIT_IF_NO_SHELF_FRU: This parameter of boolean type tells the Shelf
# Manage if it should exit if no valid Shelf FRU Information data is found.
# Default is FALSE.
#
EXIT_IF_NO_SHELF_FRU = FALSE

# VERIFY_SHELF_FRU_CHECKSUM: This parameter boolean type specifies whether
# the Shelf FRU Information record checksums should be validated. The
# default is TRUE.
#
VERIFY_SHELF_FRU_CHECKSUM = TRUE

# WATCHDOG_ENABLED: This parameter of boolean type tells the Shelf Manager
# whether it should use the hardware watchdog timer supported by the CPLD or
# not. The default is TRUE.
#
WATCHDOG_ENABLED = TRUE

# REDUNDANCY_ENABLED: This parameter of boolean type tells Shelf Manager if
#    it should run in redundant mode or not. Default is TRUE.
#
REDUNDANCY_ENABLED = TRUE

# REDUNDANCY_PORT: The parameter specifies the TCP port number used for
# inter-host communications by redundant instances of the Shelf Manager.
# Default is 1040.
#
REDUNDANCY_PORT = 1040

# REDUNDANCY_NET_ADAPTER: This parameter specifies the name of network
# adapter used for communication between redundant ShMMs. Default is eth0 if
# it does not conflict with RMCP_NET_ADAPTER.
#
REDUNDANCY_NET_ADAPTER = $IP1DEVICE

# REDUNDANCY_NET_ADAPTER2: This parameter specifies the name of the second
# network adapter used for communication between redundant ShMMs (if USB
# interface is used for redundancy). By default, this parameter is not
# defined.
#REDUNDANCY_NET_ADAPTER2 = “usb1”

# REDUNDANT_IP_ADDRESS: This parameter specifies the IP address for network
```

EXAMPLE 2-1 Default `shelfman.conf` File *(Continued)*

```
# adapter used for redundant communications. This address actually provides
# a pair of IP addresses that differ in the least significant bit. They are
# assigned to redundant ShMs according to their hardware addresses, so they
# are equal on both ShMs. This parameter has no default value and must
# always be set.
#
REDUNDANT_IP_ADDRESS = $IP1ADDR

# REDUNDANCY_NETMASK: This parameter sets the network mask for the network
# adapter used for redundancy communications. Default is 255.255.255.0
#
# REDUNDANCY_NETMASK = 255.255.255.0

# RMCP_NET_ADAPTER: This parameter specifies the name of network adapter
# used for RMCP-based communications. Default is eth0:1 if it does not
# conflict with REDUNDANCY_NET_ADAPTER.
#
RMCP_NET_ADAPTER = $IPDEVICE

# RMCP_NET_ADAPTER2: This parameter specifies the alternate name of network
# adapter used for RMCP-based communications, if cross-connect links are
# supported by hardware. Undefined by default.
#
#RMCP_NET_ADAPTER2 = "eth1"

# DEFAULT_RMCP_IP_ADDRESS: This parameter specifies the default IP address
# for network adapter used for RMCP communications. It is switched over
# between redundant instances of the Shelf Manager. This address is only
# used if no IP address is set in the LAN Configuration Parameters for
# channel # 1. Default is the REDUNDANT_IP_ADDRESS parameter value.
#
DEFAULT_RMCP_IP_ADDRESS = $RMCPADDR

# PROPAGATE_RMCP_ADDRESS: This parameter specifies whether the RMCP IP
# address should be propagated to the backup Shelf Manager. If set, the
# backup Shelf Manager configures its network interface specified by
# RMCP_NET_ADAPTER using given IP address with the least significant bit
# inverted. Default is FALSE.
#
PROPAGATE_RMCP_ADDRESS = FALSE

# DEFAULT_RMCP_NETMASK: This parameter specifies the network mask for
# network adapter used for RMCP communications. Default is 255.255.255.0
#
# DEFAULT_RMCP_NETMASK = 255.255.255.0
```

EXAMPLE 2-1 Default `shelfman.conf` File *(Continued)*

```
# DEFAULT_GATEWAY_IP_ADDRESS: This parameter specifies the default gateway
# IP address used for RMCP-based communications. It should be equal for the
# redundant instances of the Shelf Manager. This address is only used if no
# gateway address is set in the LAN Configuration Parameters for channel 1.
# Default is no gateway.
#
DEFAULT_GATEWAY_IP_ADDRESS = $GATEWAY

# SWITCHOVER_TIMEOUT_ON_BROKEN_LINK: This parameter sets the number of
# seconds to wait before switchover if the RMCP link is down, i.e. system
# manager is inaccessible from the shelf manager. A zero value of this
# parameter leads to an immediate switchover on RMCP link fault detection.
# With a -1 value, no automatic switchovers on RMCP link faults will occur.
# The default value is 10 second.
#
SWITCHOVER_TIMEOUT_ON_BROKEN_LINK = 10

# CONSOLE_LOGGING_ENABLED: This parameter of boolean type enables or
# disables log messages output to the console from which the Shelf Manager
# was started. Default is FALSE.
#
CONSOLE_LOGGING_ENABLED = FALSE

# SYSLOG_LOGGING_ENABLED: This parameter of boolean type enables or disables
# logging messages to the syslog facility. Default is TRUE.
#
SYSLOG_LOGGING_ENABLED = TRUE

# VERBOSITY: This parameter sets the Shelf Manager verbosity level. This
# value is actually a bitmask with each bit enabling a corresponding class
# of output messages. The current bit layout has 8 classes:
#       Errors:                 0x01
#       Warnings:               0x02
#       Information:            0x04
#       Verbose Info:           0x08
#       Debug Trace Messages: 0x10 (not recommended)
#       Verbose Debug Trace:  0x20 (not recommended)
#       Demo Messages:          0x40 (not recommended)
#       Locks Information:     0x80 (not recommended)
# The default verbosity level is 7 i.e. errors, warnings and information.
#
VERBOSITY = 7
```

EXAMPLE 2-1 Default `shelfman.conf` File *(Continued)*

```
### PICMG 2.x specific settings

# 2_X_SYSTEM: If configured, this parameter explicitly specifies the current
# system as CompactPCI (if TRUE) or AdvancedTCA (if FALSE). If not specified
# the choice of the system type is made automatically. It is not recommended
# to specify this parameter, unless it is necessary to override an incorrect
# hardware detection algorithm for the system type. Default is FALSE.
#
# 2_X_SYSTEM = FALSE

# CTCA_FRU_RESET_TIMEOUT: This parameter specifies the time interval in
# msecs which is used to holds the BD_SEL# line low in order to reset a
# CompactPCI board. Default is 500 milliseconds.
#
# CTCA_FRU_RESET_TIMEOUT = 500

# CTCA_HEALTHY_TIMEOUT: This parameter specifies the time interval in
# seconds during which the Shelf Manager waits for the HEALTHY# signal to
# appear after powering on a CompactPCI board. If the board HEALTHY# signal
# is not detected within the specified time, the Shelf Manager will
# deactivate this board. Default is 0 which means endless waiting.
#
# CTCA_HEALTHY_TIMEOUT = 0
#
### Notification settings

# MAX_EVENT_SUBSCRIBERS: The parameter defines the maximum number of
# entities
# that can simultaneously subscribe to receive event notifications
# from the Shelf Manager.
#
MAX_EVENT_SUBSCRIBERS = 64

# MAX_PENDING_EVENT_NOTIFICATIONS: The parameter defines the maximum number
# of outstanding event notifications for each active subscriber.
#
MAX_PENDING_EVENT_NOTIFICATIONS = 1024

# MAX_EVENT_SUBSCRIBER_IDLE_TIME: This parameter defines the maximum timeout
# for an event subscriber, in seconds, between the moment when an event
# arrives and the moment when the subscriber retrieves this event from the
# Shelf Manager. If this timeout is exceed, the subscriber is considered
# dead and is automatically unregistered.
#
MAX_EVENT_SUBSCRIBER_IDLE_TIME = 60
```

Carrier-Specific Configuration File

After reading the common configuration file `/etc/shelfman.conf`, the Shelf Manager reads the carrier-specific configuration file `/etc/shelfman.conf.`*carrier-name*, where *carrier-name* is the name of the ShMM carrier used in the relevant shelf, in lowercase characters. Settings in the carrier-specific configuration file override settings for the same variable in the common configuration file.

This mechanism allows redefinition of common settings on a carrier-specific basis. Typically, only a few critical configuration variables are defined in the carrier-specific file. For instance, the appropriate value for the `MIN_FAN_LEVEL` parameter may well be determined by the shelf architecture and the fan facilities that it implements. This mechanism allows such shelf-specific constraints to be enforced.

One result of this mechanism is that to change the effective value of a configuration parameter that is specified in the carrier-specific configuration file, the change must be made in that configuration file. A change for such a variable in the common configuration file will not have any effect.

Caution – Avoid changing the variables in the `shelfman.conf.acb` file. They contain carrier-specific settings and take precedence over the values in the `/etc/shelfman.conf` file. Changing these system-specific variables can result in an unstable system.

Two exceptions are the `SWITCHOVER_TIMEOUT_ON_BROKEN_LINK` and `PROPAGATE_RMCP_ADDRESS` variables. In releases prior to R3U1, these variables must be changed in `shelfman.conf.acb` file to take effect. In release R3U1 and above, these variables are removed from the `shelfman.conf.acb` file and should be changed in `/etc/shelfman.conf` file.

Verbosity Level Description

The verbosity level allows for additional output to be sent to either the console or to the Syslog depending on how the configuration parameters `CONSOLE_LOGGING_ENABLED` and `SYSLOG_LOGGING_ENABLED` are set. The `VERBOSITY` configuration parameter is a hexadecimal bit mask, each bit enabling output of a specific type of message:

0x01	Error messages
0x02	Warning messages
0x04	Informational messages

0x08	Verbose informational messages
0x10	Trace messages (not recommended)
0x20	Verbose trace messages
0x40	Messages displayed for important commands sent to the IPM controllers during their initialization (not recommended)
0x80	Verbose messages about acquiring and releasing internal locks (not recommended)

The default debug level is `7` which allows error, warning, and informational messages to be output.

Setting the Date and Time

When the system is brought up for the first time, the clock is not set and must be initialized. Initially the clock is set to January 1, 1970. The date can be accessed via the serial console.

```
# date
Thu Jan  1 03:16:30 UTC 1970
```

To change the date, type in the correct date using the `date` application. The format for the `date` command is *MMDDHHmmCCYY*`.`*ss*, where:

MM	Month
DD	Day
HH	Hour (use 24 hour notation)
mm	Minutes
CC	Century
YY	Year
.*ss*	Seconds

For example:

```
# date 092916282007.10
Sat Sep 29 16:28:00 UTC 2007
```

To make the date persistent, you must store it using the `hwclock` application.

```
# hwclock -systohc
```

In some cases, you might get the error message:

```
mktime: cannot convert RTC time to UNIX time
```

This error can be ignored. It is due to the original date being in an uninitialized state.

Obtaining Date and Time from a Time Server

If the shelf management card does not have an real-time clock (RTC) battery, it is possible to obtain the system date and time from a time server during system startup and synchronize it periodically thereafter. The selected time server must support RFC 868 over TCP as required by the `rdate` utility. To enable this feature, it is necessary to define the U-Boot variable `time_server` and optionally the additional variable `timezone`.

The `time_server` variable contains the IP address of the time server that the Shelf Manager is to query for the system time after the startup. This variable is propagated to the Linux level as the environment variable `TIMESERVER`. If this variable is set, the startup script `/etc/netconfig` starts the script `/etc/timesync` as a daemon, which runs in an endless loop and queries the time server with a default interval of 300 seconds. To change this interval, edit the script `/etc/timesync` and change the value of the variable `INTERVAL`.

Note – When `time_server` variable is specified, the `ip1device` variable must be set to `usb0` for proper synchronization.

The variable `timezone` contains the name of the current time zone followed by its offset from Greenwich Mean Time (GMT). The offset is positive for time zones to the west of Greenwich and negative for time zones to the east of Greenwich. This variable is propagated to the Linux level as the environment variable `TZ`. The default value of this variable is `UTC0`; that is, Universal Coordinated Time (UTC), which matches Greenwich time.

The time sent by time servers is GMT; if the time zone on the Shelf Manager is not set or not set correctly, the time obtained from the time server will be interpreted incorrectly. The three-letter name of the time zone is not used by the Shelf Manager, but is propagated to set the Linux time zone. (For instance, if the time zone name XXX0 is used, the `date` command produces output like `Thu Sep 9 21:24:24 XXX 2004`.

Here is an example of a `timezone` definition for US Eastern Time:

```
timezone = EST5
```

Here the digit 5 specifies that the time zone is five hours west of GMT. Any arbitrary three letters can replace EST; they are used to identify the time zone in (for example) Linux date command output.

Setting the Time Zone for Daylight Saving

The U-Boot timezone variable must be set accordingly to set the timezone for daylight saving. The timezone variable is propagated to the Linux level as the environment variable TZ and uses a format similar that of the TZ Linux variable.

▼ To Set the Time Zone for Daylight Saving

1. **Stop the autoboot process at the U-Boot prompt by hitting any key at the prompt:**

```
U-Boot 1.x.x (Nov 11 2008 - 11:32:08)
      :
      :
Hit any key to stop autoboot: 0
```

2. **Set the** timezone **variable (for example PST/PDT)**

```
shmm500 setenv timezone PST8PDT,M3.2.0/2,M11.1.0/2
```

3. **Save the new timezone value:**

```
shmm500 saveenv
```

4. **Reset the ShMM:**

```
shmm500 reset
```

Setting Up User Accounts on the Shelf Management Card

User accounts for RMCP access are set up using the Shelf Manager's CLI. User information is entered on the active shelf management card, and immediately *mirrored*, or shared, on the standby shelf management card. The shelf management card supports 32 accounts with passwords.

▼ To Add a User Account For RMCP Access

1. **Log in to the active shelf management card.**
2. **Add a user:**

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

where the variable parameters have the following meaning:

userid – a valid user ID

user-name – the user name (up to 16 characters)

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)

The following example shows how to add user 9 with the name `root`, administrator privilege level, and password `PICMG guru`.

```
# clia user add 9 "root" 0x40 4 "PICMG guru"
Pigeon Point Shelf Manager Command Line Interpreter
User 9 added successfully
#
# clia user
Pigeon Point Shelf Manager Command Line Interpreter
1: ""
        Channels 0-15 Privilege level: "Administrator"
```

```
              Flags: "IPMI Messaging"
9: "root"
        Channels 0-15 Privilege level: "Administrator"
              Flags: "IPMI Messaging"
#
```

See "`user`" on page 339 for more information about permissions and the `clia user` command.

Username Restrictions

The username field has a maximum length of 16 characters. It must contain at least one lowercase alphabetic character, and the first character must be alphabetic.

Valid characters for *username* include:

- Alphabetic characters
- Numeric characters
- Period (.)
- Underscore (_)
- Hyphen (-)

Passwords

Passwords can be up to 16 characters long, anything over 16 is truncated.

Disabling Remote `root` Login

When using `telnet`, `ssh`, and `ftp` to remotely logging into the ShMM as `root`, the `root` password can be visible over the network. Disabling these remote root logins provides the additional security needed to protect the root password.

Disabling remote root login involves removing the `root` userid from the `telnet`, `ssh`, and `ftp` configuration files on both the active and standby ShMMs, adding a non-root user, and giving the new user `su -` capability. To do this, use the following procedures:

1. **Log on the active ShMM as** `root`**.**

2. **Disable remote root login for SSH**

   Edit the `/etc/inetd.conf` file and add the `-g` option to `/bin/sshd -i` command at the end the line starting with `ssh`. For example:

```
ssh stream  tcp     nowait  root    /bin/tcpd /bin/sshd -i -g
```

3. **Disable root login for ftp**

   Edit the `/etc/ftpaccess` file and comment out the `allow-uid %0` parameter by inserting a # (pound sign) in the beginning of the line and add the `deny-uid %0` parameter as a new line in the file. For example:

```
#allow-uid %0
deny-uid %0
```

4. **Disable root login for telnet**

   Edit the `/etc/securetty` file and remove or comment out all the entries starting with `ttyp`. Be sure not to delete or comment out the `console` and `ttyS0` lines or else `root` access on serial console will also be denied.

5. **Add a new non-root user:**

```
#adduser username
password: password
#
```

6. **Give the new userid** `su -` **capability**

a. **Create an** `/etc/busybox.conf` **file and add the two lines shown in example.**

```
# cat /etc/busybox.conf
[SUID]
su = ssx root.0     # su can be performed from non-root
passwd = ssx root.0 # non-root user can change his password
#
```

b. **Change the file ownership and group to** `root`**. For example:**

```
# chown 0.0 /etc/busybox.conf
# chmod 600 /etc/busybox.conf
```

7. **Repeat the same steps on Standby ShMM and reboot**

Configuring OpenHPI on the Shelf Manager

The Shelf Manager includes support for OpenHPI which is an open source implementation of the SA Forum's Hardware Platform Interface (HPI). HPI provides an interface to managing computer hardware, typically for chassis and rack based servers. Access to HPI is done through the OpenHPI SNMP subagent using the SNMP MIB. Refer to the *Sun Netra CT900 Server Software Developer's Guide* for more information on OpenHPI and SNMP.

There are two configuration files that require the system administrator's attention:

- `/etc/openhpi.conf` – OpenHPI configuration file
- `/etc/snmpd.conf` – SNMP agent configuration file

The `/etc/openhpi.conf` File

If you are using an ATCA release earlier than R3U3, the OpenHPI configuration file, `/etc/openhpi.conf`, must be updated to provide the correct IP address for the ShMM. After the configuration file is updated, the ShMM must be reset to implement the changes.

Note – Do not use a variable for the IP address in the `/etc/openhpi.conf` file. Therefore `$IPADDR` and `localhost` can not be used.

With R3U3 and newer releases, it is not necessary to set the RMCP address in `/etc/openhpi.conf`. The ShMM software automatically reads the Shelfman IP connection record from the midplane FRU that has the RMCP address and starts `openhpid` with that RMCP address.

▼ To Modify the `/etc/openhpi.conf` File

1. **If using an ATCA release earlier than R3U3, edit the `/etc/openhpi.conf` file and change the value of the `libpigeonpoint addr` parameter to the ShMM's IP address.**

   A snip of the `/etc/openhpi.conf` file follows:

```
handler libpigeonpoint {
    entity_root = "{SYSTEM_CHASSIS,1}"
    name = "lan"            # RMCP
    addr = "10.12.235.18"   # Host name or IP address
    port = "623"            # RMCP port
    auth_type = "none"      # none, md5 or straight
    auth_level = "admin"    # operator or admin
    logflags = ""           # "" means logging off; also use "file stdout"
    logfile = "openhpi"     # log file name prefix; ${logfile}.log
    logfile_max = "50"      # maximum log file size in kilobytes
    UseCachedSdrs = "no"   # set it to "yes" to use Cached Device SDR repository
}
```

2. **Reboot the ShMM by issuing the `reboot` command at the prompt.**

   For example:

```
# reboot
```

The `/etc/snmpd.conf` File

The SNMP agent configuration file, `/etc/snmpd.conf`, defines how the SNMP agent operates and includes directives for access control and setting traps. Information on access control, SNMPv3 configuration, and setting traps is provided in the subsequent sections. Refer to the *Sun Netra CT900 Server Software Developer's Guide* for more information.

Access Control

The SNMP agent supports the View-Based Access Control Model (VACM) a defined in RFC 2575. To this end, it recognizes the following keywords in the configuration file:

- `com2sec`
- `group`
- `access`
- `view`

In addition, it recognizes some easier-to-use wrapper directives:

- `rocommunity`
- `rwcommunity`
- `rouser`
- `rwuser`

This section defines how to configure the `snmpd` program to accept various types and levels of access.

`rouser` *user* `[noauth|auth|priv] [`*OID*`]`
`rwuser` *user* `[noauth|auth|priv] [`*OID*`]`

Creates an SNMPv3 USM user in the VACM access configuration tables. It is more efficient (and powerful) to use the combined `group`, `access`, and `view` directives, but these wrapper directives are much simpler.

The minimum level of authentication and privacy the user must use is specified by the first token (which defaults to `auth`). The *OID* (object identifier) parameter restricts access for that user to everything below the given *OID*.

`rocommunity` *community* `[`*source*`] [`*OID*`]`
`rwcommunity` *community* `[`*source*`] [`*OID*`]`

Create read-only and read-write communities that can be used to access the agent. They are a quick wrapper around the more complex and powerful `com2sec`, `group`, `access`, and `view` directive lines. They are not as efficient as these, because groups are not created, so the tables are potentially larger. These directives are not recommended for complex environments. If your environment is relatively simple or you can sustain a small negative performance impact, use these directives.

The format of the *source* token is described in the `com2sec` directive section below. The *OID* token restricts access for that community to everything below that given *OID*.

`com2sec` *name source community*

Specifies the mapping from a *source/community* pair to a security *name*. `source` can be a hostname, a subnet, or the word `default`. A subnet can be specified as IP/mask or IP/bits. The first *source/community* combination that matches the incoming packet is selected.

`group` *name model security*

Defines the mapping from *securitymodel/securityname* to a `group`. *model* is one of `v1`, `v2c`, or `usm`.

`access` *name context model level prefx read write notify*

Maps from *group/security* and *model/security* level to a view. *model* is one of `any`, `v1`, `v2c`, or `usm`. *level* is one of `noauth`, `auth`, or `priv`. *prefx* specifies how *context* should be matched against the context of the incoming PDU, either `exact` or `prefix`. *read*, *write* and *notify* specify the view to be used for the corresponding access. For v1 or v2c access, *level* is `noauth`, and *context* is empty.

`view` *name type subtree* [*mask*]

Defines the named view. *type* is either `included` or `excluded`. *mask* is a list of hex octets, separated by a period (.) or a colon (:). The mask defaults to `ff` if not specified. Use of the *mask* allows you to control access to one row in a table in a relatively simple way. As an example, as an ISP you might consider giving each customer access to his or her own interface:

```
view cust1 included interfaces.ifTable.ifEntry.ifIndex.1 ff.a0
view cust2 included interfaces.ifTable.ifEntry.ifIndex.2 ff.a0

# interfaces.ifTable.ifEntry.ifIndex.1 == .1.3.6.1.2.1.2.2.1.1.1
# ff.a0 == 11111111.10100000
```

These entries cover up and include the row index, yet still allow the user to vary the field of the row.

The following are VACM examples:

```
# sec.name source community
com2sec local localhost private
com2sec mynet 10.10.10.0/24 public
com2sec public default public

# sec.model sec.name
group mygroup v1 mynet
group mygroup v2c mynet
group mygroup usm mynet
group local v1 local
group local v2c local
group local usm local
group public v1 public
group public v2c public
group public usm public

# incl/excl subtree mask
view all included .1 80
view system included system fe
view mib2 included .iso.org.dod.internet.mgmt.mib-2 fc

# context sec.model sec.level prefix read write notify
access mygroup "" any noauth exact mib2 none none
access public "" any noauth exact system none none
access local "" any noauth exact all all all
```

SNMPv3 Configuration

`engineID` *string*

The subagent needs to be configured with an `engineID` to be able to respond to SNMPv3 messages. With this configuration file line, the `engineID` is configured from *string*. The default value of the `engineID` is configured with the first IP address found for the hostname of the machine.

`createUser` *username* `(MD5|SHA)` *authpassphrase* `[DES]`[*privpassphrase*]

`MD5` and `SHA` are the authentication types to use, but you must have built the package with OpenSSL installed in order to use SHA. The only privacy protocol currently supported is `DES`. If the *privpassphrase* is not specified, it is assumed to be the same as *authpassphrase*.

Note – The users created are useless unless they are also added to the VACM access control tables described above.

Note – The minimum pass phrase length is 8 characters.

Setting Up Traps and Informing Destinations

`trapcommunity` *string*

Defines the default community *string* to be used when sending traps. Note that this command must be used prior to any of the three commands (immediately following) that are intended for use with this community string.

`trapsink` *host* [*community* [*port*]]
`trap2sink` *host* [*community* [*port*]]
`informsink` *host* [*community* [*port*]]

Define the hosts to receive traps (or inform notifications with `informsink`). The daemon sends a Cold Start trap when it starts up. If enabled, it also sends traps on authentication failures. You can specify multiple `trapsink`, `trap2sink` and `informsink` lines to specify multiple destinations. Use `trap2sink` to send SNMPv2 traps and `informsink` to send inform notifications. If *community* is not specified, the string from a preceding `trapcommunity` directive is used. If *port* is not specified, the well-known SNMP trap port (162) is used.

`trapsess` [*snmpcmdargs*] *host*

A more generic trap configuration token that allows any type of trap destination to be specified with any version of SNMP. This requires that you specify a version number of `v2c` or `v3` as well.

▼ To Update the `/etc/snmpd.conf` File

1. **Edit the `/etc/snmpd.conf` file to add, change, or delete directives as needed.**
2. **Reboot the ShMM by issuing the `reboot` command at the prompt.**

Enabling the OpenHPI Daemon Monitor

When you enable the daemon monitor, the OpenHPI process is monitored and, if the daemon for any reason dies or exits, the following actions are taken.

- When active and backup ShMMs are running on a system:
  - If daemon dies or exits on active ShMM, then switchover occurs
  - If daemon dies or exits on backup ShMM, then reboot occurs

- If only one ShMM is running on a system, the ShMM is rebooted

By default this feature is disabled. After installing the R3U1 updates on the ShMM, you can enable the monitor as follows.

1. **Set the U-Boot variable** `monitor_daemons` **to** `y`**.**

   See "Setting Up U-Boot" on page 18 for more information.

```
# setenv monitor_daemons y
# reboot
```

Note – If you are setting variables in the U-Boot shell, enter `saveenv` command before the `reboot` command to save the changes:

---

2. **Repeat the same commands for the backup ShMM.**

Configuring RADIUS on the Shelf Manager

RADIUS (Remote Authentication Dial-In User Service) is a client/server protocol and software that enables remote access servers to communicate with a central server to authenticate dial-in users and authorize their access to the requested system or service. The protocol is covered by RFC2865.

The Shelf Manager version 2.4.x (and above) provides RADIUS client support. The system administration only needs to edit the `/etc/raddb/server` file on the Shelf Manager and provide the server information. The following is a sample `/etc/raddb/server` file:

```
#<radius_server_ip_address> <radius_secret> <wait_period>
127.0.0.1         secret            1
other-server      other-secret      3
```

Where

- *radius_server_ip_address* is the IP address of a RADIUS server.
- *radius_secret* is a secret text string that the RADIUS server and RADIUS client share. Both server and client must be configured with the same secret text string in order for them to communicate.

- *wait_period* is the timeout in seconds and controls how many seconds the module waits before deciding that the server has failed to respond. The default timeout is `3` seconds.

If multiple RADIUS servers are provided, they are tried in order. The first server to return success or failure causes the module to return success or failure. Only if a server fails to response is it skipped and the next server in turn is used.

By default, the superuser (`root`) can log in to Shelf Manager using `console`, `telnet`, or `ssh` using either the factory default password or RADIUS password.

To enble the Shelf Manager to accept *only* RADIUS passwords or to accept *only* local `root` password, the standard PAM configuration files need to be modified accordingly. These files are `/etc/pam.d/login` and `/etc/pam.d/sshd` on the Shelf Manager. (Refer to standard PAM (Port to Application Mapping) rules for configuration).

CHAPTER 3

Administering Your System

You administer your system using the shelf management card command-line interface (CLI) or via Ethernet using the RMCP interface.

This chapter contains the following sections:

IPMI LAN Interface

The IPMI LAN interface is required by the ATCA specification and supports IPMI messaging with the Shelf Manager through the Remote Management Control Protocol (RMCP). A system administrator that uses RMCP to communicate with shelves is able to interact with any ATCA-compliant Shelf Manager. This low-level interface provides access to the IPMI aspects of a shelf, including the ability for the system administrator to issue IPMI commands to IPM controllers in the shelf, using the Shelf Manager as a proxy.

Supported IPMI Commands

The standard IPMI commands are documented in the PICMG 3.0, ATCA specification. TABLE 3-1 shows the IPMI commands implemented by the Shelf Manager. Due to security considerations, the treatment of a given command may be different, depending on whether it is received over the RMCP interface or on IPMB-0.

TABLE 3-1 IPMI Commands Implemented by the Shelf Manager

Command	NetFn	CMD	Arriving from RMCP Interface	Arriving from IPM Controller
Get Device ID	App	01h	Supported	Supported
Cold Reset	App	02h	Supported	Supported
Warm Reset	App	03h	Not Supported	Not supported
Get Self Test Results	App	04h	Supported	Supported
Manufacturing Test On	App	05h	Not Supported	Not supported
Set ACPI Power State	App	06h	Supported	Supported
Get ACPI Power State	App	07h	Supported	Supported
Get Device GUID	App	08h	Supported	Supported
Reset Watchdog Timer	App	22h	Supported	Supported
Set Watchdog Timer	App	24h	Supported	Supported
Get Watchdog Timer	App	25h	Supported	Supported
Set BMC Global Enables	App	2Eh	Supported	Supported
Get BMC Global Enables	App	2Fh	Supported	Supported
Clear Message Flags	App	30h	Supported	Supported

TABLE 3-1 IPMI Commands Implemented by the Shelf Manager *(Continued)*

Command	NetFn	CMD	Arriving from RMCP Interface	Arriving from IPM Controller
Get Message Flags	App	31h	Supported	Supported
Enable Message Channel Receive	App	32h	Not Supported	Not supported
Get Message	App	33h	Not Supported	Not supported
Send Message	App	34h	Supported	Supported
Read Event Message Buffer	App	35h	Not Supported	Not supported
Get BT Interface Capabilities	App	36h	Not Supported	Not supported
Get System GUID	App	37h	Supported	Supported
Get Channel Authentication Capabilities	App	38h	Supported	Supported(*)
Get Session Challenge	App	39h	Supported	Not supported
Activate Session	App	3Ah	Supported	Not supported
Set Session Privilege Level	App	3Bh	Supported	Not supported
Close Session	App	3Ch	Supported	Not supported
Get Session Info	App	3Dh	Supported	Supported(*)
Get AuthCode	App	3Fh	Supported	Supported(*)
Set Channel Access	App	40h	Supported	Supported(*)
Get Channel Access	App	41h	Supported	Supported(*)
Get Channel Info	App	42h	Supported	Supported(*)
Set User Access	App	43h	Supported	Supported(*)
Get User Access	App	44h	Supported	Supported(*)
Set User Name	App	45h	Supported	Supported(*)
Get User Name	App	46h	Supported	Supported(*)
Set User Password	App	47h	Supported	Supported(*)
Activate Payload	App	48h	Not Supported	Not supported
Deactivate Payload	App	49h	Not Supported	Not supported
Get Payload Activation Status	App	4Ah	Not Supported	Not supported
Get Payload Instance Info	App	4Bh	Not Supported	Not supported
Set User Payload Access	App	4Ch	Not Supported	Not supported
Get User Payload Access	App	4Dh	Not Supported	Not supported
Get Channel Payload Support	App	4Eh	Not Supported	Not supported
Get Channel Payload Version	App	4Fh	Not Supported	Not supported

TABLE 3-1 IPMI Commands Implemented by the Shelf Manager *(Continued)*

Command	NetFn	CMD	Arriving from RMCP Interface	Arriving from IPM Controller
Get Channel OEM Payload Info	App	50h	Not Supported	Not supported
Master Write-Read	App	52h	Not Supported	Not supported
Get Channel Cipher Suites	App	54h	Not Supported	Not supported
Suspend/Resume Payload Encryption	App	55h	Not Supported	Not supported
Set Channel Security Keys	App	56h	Not Supported	Not supported
Get System Interface Capabilities	App	57h	Not Supported	Not supported
Get Chassis Capabilities	Chassis	00h	Supported	Supported
Get Chassis Status	Chassis	01h	Supported	Supported
Chassis Control	Chassis	02h	Supported	Supported
Chassis Reset	Chassis	03h	Not Supported	Not supported
Chassis Identify	Chassis	04h	Not Supported	Not supported
Set Chassis Capabilities	Chassis	05h	Supported	Supported
Set Power Restore Policy	Chassis	06h	Not Supported	Not supported
Get System Restart Cause	Chassis	07h	Not Supported	Not supported
Set System Boot Options	Chassis	08h	Not Supported	Not supported
Get System Boot Options	Chassis	09h	Not Supported	Not supported
Set Front Panel Button Enables	Chassis	0Ah	Not Supported	Not supported
Set Power Cycle Interval	Chassis	0Bh	Not Supported	Not supported
Get POH Counter	Chassis	0Fh	Not Supported	Not supported
Set LAN Configuration Parameters	Transport	01h	Supported	Supported(*)
Get LAN Configuration Parameters	Transport	02h	Supported	Supported
Suspend BMC ARPs	Transport	03h	Supported	Supported(*)
Get IP/UDP/RMCP statistics	Transport	04h	Not Supported	Not supported
Set Serial/Modem Configuration	Transport	10h	Not Supported	Not supported
Get Serial/Modem Configuration	Transport	11h	Not Supported	Not supported
Set Serial/Modem Mux	Transport	12h	Not Supported	Not supported
Get TAP Response Codes	Transport	13h	Not Supported	Not supported
Set PPP UDP Proxy Transmit Data	Transport	14h	Not Supported	Not supported
Get PPP UDP Proxy Transmit Data	Transport	15h	Not Supported	Not supported
Send PPP UDP Proxy Packet	Transport	16h	Not Supported	Not supported

TABLE 3-1 IPMI Commands Implemented by the Shelf Manager *(Continued)*

Command	NetFn	CMD	Arriving from RMCP Interface	Arriving from IPM Controller
Get PPP UDP Proxy Receive Data	Transport	17h	Not Supported	Not supported
Serial/Modem Connection Active	Transport	18h	Not Supported	Not supported
Callback	Transport	19h	Not Supported	Not supported
Set User Callback Options	Transport	1Ah	Supported	Supported(*)
Get User Callback Options	Transport	1Bh	Supported	Supported(*)
SOL Activating	Transport	20h	Not Supported	Not supported
Set SOL Configuration Parameters	Transport	21h	Not Supported	Not supported
Get SOL Configuration Parameters	Transport	22h	Not Supported	Not supported
Get FRU Inventory Area Info	Storage	10h	Supported	Supported
Read FRU Data	Storage	11h	Supported	Supported
Write FRU Data	Storage	12h	Supported	Supported
Get SDR Repository Info	Storage	20h	Supported	Supported
Get SDR Repository Allocation Info	Storage	21h	Not Supported	Not supported
Reserve SDR Repository	Storage	22h	Supported	Supported
Get SDR	Storage	23h	Supported	Supported
Add SDR	Storage	24h	Supported	Supported
Partial Add SDR	Storage	25h	Supported	Supported
Delete SDR	Storage	26h	Supported	Supported
Clear SDR Repository	Storage	27h	Supported	Supported
Get SDR Repository Time	Storage	28h	Supported	Supported
Set SDR Repository Time	Storage	29h	Supported	Supported
Enter SDR Repository Update Mode	Storage	6F or 2A (Sun legacy)h	Not Supported	Not supported
Exit SDR Repository Update Mode	Storage	2Bh	Not Supported	Not supported
Run Initialization Agent	Storage	2Ch	Not Supported	Not supported
Get SEL Info	Storage	40h	Supported	Supported
Get SEL Allocation Info	Storage	41h	Supported	Supported
Reserve SEL	Storage	42h	Supported	Supported
Get SEL Entry	Storage	43h	Supported	Supported
Add SEL Entry	Storage	44h	Supported	Supported

TABLE 3-1 IPMI Commands Implemented by the Shelf Manager *(Continued)*

Command	NetFn	CMD	Arriving from RMCP Interface	Arriving from IPM Controller
Partial Add SEL Entry	Storage	45h	Supported	Supported
Delete SEL Entry	Storage	46h	Supported	Supported
Clear SEL	Storage	47hh	Supported	Supported
Get SEL Time	Storage	48h	Supported	Supported
Set SEL Time	Storage	49h	Supported	Supported
Get Auxiliary Log Status	Storage	5Ah	Not Supported	Not supported
Set Auxiliary Log Status	Storage	5Bh	Not Supported	Not supported
Set Event Receiver	S/E	00h	Supported	Supported
Get Event Receiver	S/E	01h	Supported	Supported
Event Message	S/E	02h	Supported	Supported
Get PEF Capabilities	S/E	10h	Supported	Supported
Arm PEF Postpone Timer	S/E	11h	Supported	Supported
Set PEF Configuration Parameters	S/E	12h	Supported	Supported
Get PEF Configuration Parameters	S/E	13h	Supported	Supported
Set Last Processed Event ID	S/E	14h	Supported	Supported
Get Last Processed Event ID	S/E	15h	Supported	Supported
Alert Immediate	S/E	16h	Supported	Supported
PET Acknowledge	S/E	17h	Supported	Supported
Get Device SDR Info	S/E	20h	Supported	Supported
Get Device SDR	S/E	21h	Supported	Supported
Reserve Device SDR Repository	S/E	22h	Supported	Supported
Get Sensor Reading Factors	S/E	23h	Supported	Supported
Set Sensor Hysteresis	S/E	24h	Supported	Supported
Get Sensor Hysteresis	S/E	25h	Supported	Supported
Set Sensor Threshold	S/E	26h	Supported	Supported
Get Sensor Threshold	S/E	27h	Supported	Supported
Set Sensor Event Enable	S/E	28h	Supported	Supported
Get Sensor Event Enable	S/E	29h	Supported	Supported
Re-arm Sensor Events	S/E	6F or 2A (Sun legacy)h	Supported	Supported

TABLE 3-1 IPMI Commands Implemented by the Shelf Manager *(Continued)*

Command	NetFn	CMD	Arriving from RMCP Interface	Arriving from IPM Controller
Get Sensor Event Status	S/E	2Bh	Supported	Supported
Get Sensor Reading	S/E	2Dh	Supported	Supported
Set Sensor Type	S/E	2Eh	Supported	Supported
Get Sensor Type	S/E	2Fh	Supported	Supported
Get PICMG Properties	PICMG	00h	Supported	Supported
Get Address Info	PICMG	01h	Supported	Supported
Get Shelf Address Info	PICMG	02h	Supported	Supported
Set Shelf Address Info	PICMG	03h	Supported	Supported
FRU Control	PICMG	04h	Supported	Supported
Get FRU LED Properties	PICMG	05h	Supported	Supported
Get LED Color Capabilities	PICMG	06h	Supported	Supported
Set FRU LED State	PICMG	07h	Supported	Supported
Get FRU LED State	PICMG	08h	Supported	Supported
Set IPMB State	PICMG	09h	Supported	Supported
Set FRU Activation Policy	PICMG	0Ah	Supported	Supported
Get FRU Activation Policy	PICMG	0Bh	Supported	Supported
Set FRU Activation	PICMG	0Ch	Supported	Supported
Get Device Locator Record ID	PICMG	0Dh	Supported	Supported
Set Port State	PICMG	0Eh	Supported	Supported
Get Port State	PICMG	0Fh	Supported	Supported
Compute Power Properties	PICMG	10h	Supported	Supported
Set Power Level	PICMG	11h	Supported	Supported
Get Power Level	PICMG	12h	Supported	Supported
Renegotiate Power	PICMG	13h	Not Supported	Supported
Get Fan Speed Properties	PICMG	14h	Supported	Supported
Set Fan Level	PICMG	15h	Supported	Supported
Get Fan Level	PICMG	16h	Supported	Supported
Bused Resource	PICMG	17h	Not Supported	Supported
Get IPMB Link Info	PICMG	18h	Supported	Supported
Get Shelf Power Allocation	PICMG	19h	Supported	Supported

TABLE 3-1 IPMI Commands Implemented by the Shelf Manager *(Continued)*

Command	NetFn	CMD	Arriving from RMCP Interface	Arriving from IPM Controller
Get Shelf Manager IPMB Address	PICMG	1Bh	Supported	Supported
Set Fan Policy	PICMG	1Ch	Not Supported	Not supported
Get Fan Policy	PICMG	1Dh	Not Supported	Not supported
FRU Control Capabilities	PICMG	1Eh	Supported	Supported
FRU Inventory Device Lock Control	PICMG	1Fh	Supported	Supported
FRU Inventory Device Write	PICMG	20h	Supported	Supported
Get Shelf Manager IP Addresses	PICMG	21h	Supported	Supported

Note - Commands marked by (*) are supported from the IPMB-0 side *only* if the configuration parameter `ALLOW_ALL_COMMANDS_FROM_IPMB` is set to `TRUE`.

`Get Self Test Results` IPMI Command

The `Get Self Test Results` command returns the results of the POST tests performed by the U-Boot utility when the ShMM is powered up. If all tests are passed, the status code 0x55 is returned. If any tests fail, the device-specific failure code 0x59 is returned and the third byte contains the following bit mask:

[7:5]	Reserved
[4]	1b = Ethernet test failed
[3]	1b = UART test failed
[2]	1b = U-Boot CRC test failed
[1]	1b = I^2C test failed
[0]	1b = Memory test failed

IPMI OEM Extension Commands

In addition to the specification-defined IPMI commands listed in TABLE 3-1, the Shelf Manager implements several OEM-defined IPMI commands for the convenience of the System Manager. The Shelf Manager IPMI extension commands are listed in TABLE 3-2 and described in Appendix B.

TABLE 3-2 Shelf Manager IPMI Extension Commands

Command	NetFn	Opcode
`Get Shelf FRU Record Data`	3Eh	1h
`Set Shelf FRU Record Data`	2Eh	05h
`Notify Shelf Manager About an Extracted FRU`	2Eh	2h
`Initiate Shelf Manager Switchover`	3Eh	3h
`Subscribe for Event Notifications`	2Eh	4h

Shelf Manager Command-Line Interface

The Shelf Manager command-line interface (CLI) can be used to communicate with the intelligent management controllers of the shelf, with boards, and with the Shelf Manager itself, through text commands. The CLI is an IPMI-based set of commands that can be accessed directly or through a higher-level management application or a script. Administrators can access the CLI through a Telnet connection or the shelf management card serial port. Using the CLI, operators can access information about the current state of the shelf including current FRU population, current sensor values, threshold settings, recent events, and overall shelf health.

Starting the Command-Line Interface

To use the CLI, first log on to the Linux operation system on the shelf management card. Once logged in, run the executable `clia` from the command line with specific parameters. The first parameter is the command verb. The `clia` executable is located on the virtual `root` file system maintained by Linux running on the shelf management card. The `clia` executable connects to the main Shelf Manager software process, passes the command information to it, and retrieves the results. The Shelf Manager must be running prior to starting the CLI.

For example:

```
# clia ipmc 20

Pigeon Point Shelf Manager Command Line Interpreter

20: Entity: (d0, 0) Maximum FRU device ID: 20
    PICMG Version 2.0
   Hot Swap State: M4, Previous: M3, Last State Change Cause: Normal State Change
(0)
#
```

If started without parameters, `clia` enters an interactive mode. In that mode, the program repeatedly issues a prompt to the terminal, accepts user input as the next command with parameters, executes that command, and shows the results on the terminal, until the user types the command `exit` or `quit`. For example:

```
# clia

Pigeon Point Shelf Manager Command Line Interpreter

CLI> ipmc 20

20: Entity: (d0, 0) Maximum FRU device ID: 20
    PICMG Version 2.0
   Hot Swap State: M4, Previous: M3, Last State Change Cause: Normal State Change
(0)

CLI> exit
#
```

Shelf Manager CLI Commands

The Shelf Manager CLI implements the following commands. They are described in detail in Appendix A, "Shelf Manager CLI Commands" on page 157 with a subsection for each command, in alphabetical order of the command names.

TABLE 3-3 Shelf Manager CLI Command Summary

Command	Parameters	Description
`activate`	IPMB address FRU device ID	Activates the specified FRU.
`airfilterreplaced`	dd.mm.yyyy (optional)	Sets a date when the air filter should be replaced.
`alarm`	alarm type clear information	Activates or clears Telco alarms. Also displays alarm information
`amcportstate`	IPMB address FRU device ID or AMC number (optional)	Shows AMC port state information for a specified AMC. If AMC number is not specified, the AMC port state information is reported for all active AMCs for the designated IPM controller.
`board`	slot number (optional)	Shows information about boards.
`boardreset`	slot number	Resets the specified ATCA board.
`busres`	subcommand, with its parameters	Performs the specified operation on the Bused E-Keying-managed resources.
`console`	slot number	Opens a console session on the node board in the specified slot.
`deactivate`	IPMB address FRU device ID	Deactivates the specified FRU.
`debuglevel`	new debug level (optional)	Gets current debug level for the Shelf Manager or sets a new debug level.
`exit|quit`		Exits from the interpreter in interactive mode.
`fans`	IPMB address (optional) FRU device ID (optional)	Shows information about fans.
`flashupdate`	Server IP address Pathname to firmware image	Downloads and updates system firmware on the Sun Netra CP3x60 node boards from the specified server and pathname.

TABLE 3-3 Shelf Manager CLI Command Summary *(Continued)*

Command	Parameters	Description
`fru`	IPMB address (optional) FRU device ID (optional) FRU type (optional)	Shows information about one or a group of FRUs in the shelf; FRUs are selected by type or by the parent IPM controller.
`frucontrol`	IPMB address FRU device ID Command Options	Sends FRU Control command to the specified FRU.
`frudata`	IPMB address (optional) FRU device ID (optional) block/byte offset (optional) data (optional)	Provides raw access to the FRU Information on the specified FRU.
`frudatar`	IPMB address FRU device ID File name	Reads the FRU data area of the specified FRU and stores the data in the specified file.
`frudataw`	IPMB address FRU device ID File name	Writes the FRU data in the specified file into the FRU data area of the specified FRU.
`fruinfo`	IPMB address FRU device ID	Provides user-friendly FRU Information output.
`getacousticlevel`	ETSI NEBS-A NEBS-U	Shows the system acoustic level and fan speed.
`getbootdev`	IPMB address FRU device ID or AMC address address	Shows system boot device parameter
`getfanlevel`	IPMB address (optional) FRU device ID (optional)	Shows the current level of the fan controlled by the specified FRU.
`getfanpolicy`	IPMB address (optional) FRU device ID (optional) <site-type> (optional) <site-number> (optional)	Retrieves information about Fan Tray(s) control mode and/or FRUs coverage by the specified Fan Tray(s). This command returns two different pieces of data: whether or not the site(s) are enabled/disabled for autonomous control by the Shelf Manager (based on `Set Fan Policy` commands), and whether or not the FRU site(s) are covered by the fans (according to the Fan Geography record).

TABLE 3-3 Shelf Manager CLI Command Summary *(Continued)*

Command	Parameters	Description
`getfruledstate`	IPMB address (optional) FRU device ID (optional) LED ID or `ALL` (optional)	Shows the FRU LED state.
`gethysteresis`	IPMB address (optional) sensor name (optional) sensor number (optional)	Shows both the positive and negative hystereses of the specified sensor.
`getipmbstate`	IPMB address IPMB link number (optional)	Shows the current state of IPMB-0 at the target address. If a link number is specified and the target IPMC is an IPMB switch, information about a specific link is shown.
`getlanconfig`	channel number parameter name or number (optional) set selector (optional)	Gets and shows a LAN configuration parameter for a specific channel.
`getmgmtportroute`	slot number	Displays the management port routing configuration.
`getmuxconfig`	slot number (optional)	Gets the multiplexer (MUX) configuration information from the shelf midplane records.
`getpefconfig`	parameter name or number (optional) set selector (optional)	Gets and shows a PEF configuration parameter.
`getsensoreventenable`	IPMB address (optional) sensor name (optional) sensor number (optional)	Shows the current sensor event mask values for the supported events of the specified sensors.
`getthreshold | threshold`	IPMB address (optional) sensor name (optional) sensor number (optional)	Shows threshold information about a specific sensor.
`help`		Shows the list of supported commands.
`ipmc`	IPMB address (optional)	Shows information about one or all IPM controllers in the shelf.
`localaddress`		Retrieves the IPMB address of the current Shelf Manager.
`minfanlevel`	fan level (optional)	Shows or sets the minimum fan level.
`mgmtportstate`	slot	Employs the IPMC OEM command `get ethernet port access` to query the IPMC for management port state or port routing configuration.

TABLE 3-3 Shelf Manager CLI Command Summary *(Continued)*

Command	Parameters	Description
`muxstate`	slot number	Querys the IPMC for the current multiplexer (MUX) state and port routing information.
`networkelementid`	Network Element Identifier (optional)	Allows getting or setting the Network Element Identifier.
`poll`		Initiates a poll of the IPM controllers on IPMB-0.
`sel`	IPMB address (optional) number of items (optional)	Shows several most recent items from the System Event Log maintained on the target IPM controller.
`sendamc`	IPMB address AMC address or FRU ID NetFn Command Code Byte1 (optional) ... (optional) ByteN (optional)	Transparently sends an arbitrary IPMI command to an Advanced Management Controller (AMC) that resides behind its correspondent IPM controller in a transparent way.
`sendcmd`	IPMB address Network function Command Code Byte1 (optional) ... (optional) ByteN (optional)	Transparently sends an arbitrary IPMI command to the target IPMC.
`sensor`	IPMB address (optional) sensor name (optional) sensor number (optional)	Shows information about one or a group of sensors; sensors are selected by IPM controller address, number or name.
`sensordata`	IPMB address (optional) sensor name (optional) sensor number (optional)	Shows value information for a specific sensor.
`sensorread`	IPMB address sensor number	Shows raw value information for a specific sensor (ignoring any Sensor Data Record describing the sensor).
`session`		Shows information about active RMCP sessions.
`setacousticlevel`	ETSI NEBS-A NEBS-U	Sets the system acoustic level and fan speed.

TABLE 3-3 Shelf Manager CLI Command Summary *(Continued)*

Command	Parameters	Description
`setbooddev`	IPMB address FRU device ID or AMC address boot device parameter	Sets system boot device parameter.
`setextracted`	IPMB address FRU device ID	Notifies the Shelf Manager that the specified FRU has been physically extracted from the shelf.
`setfanlevel`	IPMB address FRU device ID level	Sets a new level for the fan controlled by the specified FRU.
`setfanpolicy`	IPMB address FRU device ID action to be taken: `ENABLE` or `DISABLE` timeout (optional) site type (optional) site number (optional)	Enables or disables fan trays for cooling management in addition to the Fan Geography record if this one is presented in the Shelf FRU.
`setfruledstate`	IPMB address FRU device ID LED ID or `ALL` LED operation LED color (optional)	Sets the state of a specific LED or all LEDs for the given FRU.
`sethysteresis`	IPMB address sensor name or sensor number hysteresis to be set (`pos` or `neg`) hysteresis value	Sets new hysteresis value for the specified sensor.
`setipmbstate`	IPMB address IPMB bus name (`A` or `B`) IPMB link number (optional) action to be taken	Disables/enables IPMB-A or IPMB-B (or the specific IPMB link) on the target IPM controller.
`setlanconfig`	channel parameter name or number additional parameters	Sets the value of the LAN configuration parameter on the specified channel.
`setlocked`	IPMB address FRU device ID state	Sets the Locked bit for the specified FRU to the specified state (`0` – unlock, `1` – lock).
`setmgmtportroute`	slot number 1 or 0	Sets the management port route, including front or rear access.

TABLE 3-3 Shelf Manager CLI Command Summary *(Continued)*

Command	Parameters	Description
`setmuxconfig`	slot number port1:route value port2:route value	Sets the multiplexer (MUX) configuration information into the shelf midplane records for a specific slot. MUX configuration settings take effect during the blade activation.
`setpefconfig`	parameter name or number set selector (optional) parameter value	Sets a new value of a PEF configuration parameter.
`setpowerlevel`	IPMB address FRU device ID power level or `OFF` `Copy`	Set the power level the specified FRU, turns off power to the FRU, and copies the desired level to the present levels.
`setsensordata`	IPMB address (optional) sensor name assertion events mask (optional) deassertion events mask (optional) event_data (optional)	Changes the reading, assertion or deassertion mask, or event data bytes for the specified sensor.
`setsensoreventenable`	IPMB address sensor name sensor number global flags assertion events mask (optional) deassertion events mask (optional)	Changes the event enable masks for a specific sensor.
`setthreshold`	IPMB address sensor name sensor number threshold type threshold value	Changes a specific threshold value (upper/lower, critical/non-critical/non-recoverable) for a specific sensor.
`setuserlabel`	shelf name slot number name	Configures user assigned names for the shelf and the boards. Board names are assigned to slot numbers.
`shelf`	subcommand, with its parameters	Shows general information about the shelf; several subcommands allow setting shelf attributes and getting additional information about specific areas.
`shelfaddress`	Shelf Address string (optional)	Gets or sets the Shelf Address field of the Address Table within Shelf FRU Information.

TABLE 3-3 Shelf Manager CLI Command Summary *(Continued)*

Command	Parameters	Description
`shmstatus`		Shows the Shelf Manager Active/Backup status.
`showhost`	slot number	Displays version information about the firmware on certain Sun Netra CP3x60 node boards.
`showunhealthy`		Shows the unhealthy components of the shelf.
`switchover`		Initiates a switchover to the backup Shelf Manager.
`terminate`		Terminates the Shelf Manager without rebooting the shelf management card.
`user`	subcommand, with its parameters	Shows information about the RMCP user accounts on the Shelf Manager and provides a simple way to add, delete and modify user accounts.
`userlabel`	shelf slot slot slot-number (optional)	Displays a user assigned name for the shelf and boards. Board names are assigned to slot numbers.
`version`		Shows the Shelf Manager version information.

Most informational commands support brief and verbose modes of execution, differing in the amount of information provided. Brief mode is the default (standard); verbose mode is selected by using the option `-v` in the command line, directly after the command and before the positional arguments. Commands that are executed on the backup Shelf Manager can only access objects (such as sensors, FRUs, IPM controllers) that are local to the backup Shelf Manager.

To help the user to determine whether a specific command is being executed on the active or on the backup Shelf Manager, the following message is issued when a CLI command is executed on the backup Shelf Manager: `Running on the Backup Shelf Manager, with limited functionality`.

Monitoring Your System

The Shelf Manager CLI provides many commands to monitor your system and display system status. This section describes various ways to monitor your system. See "Shelf Manager CLI Commands" on page 89 or Appendix A, "Shelf Manager CLI Commands" on page 157 for more information.

Displaying Board and IPMC Information

Board information includes information about each IPM controller in the range of IPMB addresses allocated to ATCA slots, and about each additional FRU controlled by these controllers. The range of IPMB addresses is 82h-A0h for PICMG 3.0 systems, where boards have IPM controllers on them.

Examples for the following tasks are provided showing the commands used and their outputs.

- To Display Standard Information About All Boards in the Server
- To Display Detailed Information About a Board
- To List the Sensors on a Board
- To Display Data from a Sensor on a Board
- To List All IPMCs in the Server
- To Display Information About a Specific IPM Controller
- To Display Detailed Information About the IPM Controller

● To Display Standard Information About All Boards in the Server

In the example, only the boards in physical slots 3, 4, 5, 7, and 8 are present.

```
# clia board
Pigeon Point Shelf Manager Command Line Interpreter

Physical Slot # 3
92: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x01
    PICMG Version 2.2
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)

92: 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: "NetraCP-3010"

Physical Slot # 4
8e: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x01
    PICMG Version 2.1
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)

8e: 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: "NetraCP-3060"

8e: FRU # 1
    Entity: (0xc1, 0x61)
   Hot Swap State: M4 (Active), Previous: M7 (Communication Lost),
Last State Change Cause: Normal State Change (0x0)
    Device ID String: "SB AMC-HD-A-80X"

Physical Slot # 5
8a: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x00
    PICMG Version 2.1
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)

8a: 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: "NetraCP-3020"

Physical Slot # 7
82: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x01
    PICMG Version 2.1
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)

82: 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: "ATS1160"

82: FRU # 1
    Entity: (0xc0, 0x60)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: "TM1460A RTM"

Physical Slot # 8
84: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x01
    PICMG Version 2.1
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)

84: 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: "CP3140H-BEG"

84: FRU # 1
    Entity: (0xc0, 0x60)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: "XCP3040H-RTC"
#
```

● To Display Detailed Information About a Board

This example shows detailed information about the board in physical slot 14.

```
# clia board -v 4
Pigeon Point Shelf Manager Command Line Interpreter

Physical Slot # 4
8e: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x01
    PICMG Version 2.1
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID: 0x00, Revision: 0, Firmware: 0.210, IPMI ver 1.5
    Manufacturer ID: 00006F or 2A (Sun legacy), Product ID: 0bf4,
Auxiliary Rev: 00000000
    Device ID String: "NetraCP-3060"
    Global Initialization: 0xc, Power State Notification: 0xc,
Device Capabilities: 0x29
    Controller provides Device SDRs
    Supported features: 0x29
        "Sensor Device" "FRU Inventory Device" "IPMB Event
Generator"
    8e: Base Interface (0x00), Channel: 1
        Link: Enabled Ports: 1
        Peer Addr: 0x82, Link Type: PICMG 3.0 Base Interface
10/100/1000 BASE-T, Ext: 0 (10/100/1000 BASE-T)
    8e: Base Interface (0x00), Channel: 2
        Link: Enabled Ports: 1
        Peer Addr: 0x84, Link Type: PICMG 3.0 Base Interface
10/100/1000 BASE-T, Ext: 0 (10/100/1000 BASE-T)
    8e: Fabric Interface (0x01), Channel: 1
        Link: Enabled Ports: 1
        Peer Addr: 0x82, Link Type: PICMG 3.1 Ethernet Fabric
Interface, Ext: 0
    8e: Fabric Interface (0x01), Channel: 2
        Link: Enabled Ports: 1
        Peer Addr: 0x84, Link Type: PICMG 3.1 Ethernet Fabric
Interface, Ext: 0

8e: 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: "NetraCP-3060"
    Site Type: 0x00, Site Number: 04
```

```
    Current Power Level: 0x01, Maximum Power Level: 0x01, Current
Power Allocation: 174.0 Watts

8e: FRU # 1
    Entity: (0xc1, 0x61)
   Hot Swap State: M4 (Active), Previous: M7 (Communication Lost),
Last State Change Cause: Normal State Change (0x0)
    Device Type: "FRU Inventory Device behind management
controller" (0x10), Modifier 0x0
    Device ID String: "SB AMC-HD-A-80X"
    Current Power Level: 0x01, Maximum Power Level: 0x01, Current
Power Allocation: 21.1 Watts
#
```

- **To List the Sensors on a Board**

In this example, a list of sensors on the board at IPMB address `92` is displayed.

```
# clia sensor 92
Pigeon Point Shelf Manager Command Line Interpreter

92: LUN: 0, Sensor # 0 ("FRU 0 Hot Swap")
    Type: Discrete (0x6f), "Hot Swap" (0xf0)
    Belongs to entity: (0xa0, 96) [FRU # 0]

92: LUN: 0, Sensor # 2 ("IPMB Physical")
    Type: Discrete (0x6f), "IPMB Link" (0xf1)
    Belongs to entity: (0xa0, 96) [FRU # 0]

92: LUN: 0, Sensor # 4 ("CPU1 Temp")
    Type: Threshold (0x01), "Temperature" (0x01)
    Belongs to entity: (0x3, 96) [FRU # 0]

92: LUN: 0, Sensor # 5 ("CPU2 Temp")
    Type: Threshold (0x01), "Temperature" (0x01)
    Belongs to entity: (0x3, 96) [FRU # 0]

92: LUN: 0, Sensor # 6 ("Inlet Temp")
    Type: Threshold (0x01), "Temperature" (0x01)
    Belongs to entity: (0x3, 96) [FRU # 0]

92: LUN: 0, Sensor # 7 ("+12.0V")
    Type: Threshold (0x01), "Voltage" (0x02)
    Belongs to entity: (0x14, 96) [FRU # 0]

92: LUN: 0, Sensor # 8 ("-12.0V")
```

```
    Type: Threshold (0x01), "Voltage" (0x02)
    Belongs to entity: (0x14, 96) [FRU # 0]

92: LUN: 0, Sensor # 9 ("+5.0V VCC")
    Type: Threshold (0x01), "Voltage" (0x02)
    Belongs to entity: (0x14, 96) [FRU # 0]

92: LUN: 0, Sensor # 10 ("+3.3V Main")
    Type: Threshold (0x01), "Voltage" (0x02)
    Belongs to entity: (0x14, 96) [FRU # 0]

92: LUN: 0, Sensor # 11 ("+3.3V StandBy")
    Type: Threshold (0x01), "Voltage" (0x02)
    Belongs to entity: (0x14, 96) [FRU # 0]

92: LUN: 0, Sensor # 12 ("VBAT")
    Type: Threshold (0x01), "Voltage" (0x02)
    Belongs to entity: (0x14, 96) [FRU # 0]

92: LUN: 0, Sensor # 13 ("VDD Core0")
    Type: Threshold (0x01), "Voltage" (0x02)
    Belongs to entity: (0x14, 96) [FRU # 0]

92: LUN: 0, Sensor # 14 ("VDD Core1")
    Type: Threshold (0x01), "Voltage" (0x02)
    Belongs to entity: (0x14, 96) [FRU # 0]

92: LUN: 0, Sensor # 15 ("VTT 1.25V")
    Type: Threshold (0x01), "Voltage" (0x02)
    Belongs to entity: (0x14, 96) [FRU # 0]

92: LUN: 0, Sensor # 16 ("VDD 1.2V")
    Type: Threshold (0x01), "Voltage" (0x02)
    Belongs to entity: (0x14, 96) [FRU # 0]

    Belongs to entity: (0x14, 96) [FRU # 0]

92: LUN: 0, Sensor # 17 ("VCC TM 2.5V")
    Type: Threshold (0x01), "Voltage" (0x02)
    Belongs to entity: (0x14, 96) [FRU # 0]

92: LUN: 0, Sensor # 18 ("VDD +2.5V")
    Type: Threshold (0x01), "Voltage" (0x02)
    Belongs to entity: (0x14, 96) [FRU # 0]
```

```
92: LUN: 0, Sensor # 19 ("VDD +1.5V")
    Type: Threshold (0x01), "Voltage" (0x02)
    Belongs to entity: (0x14, 96) [FRU # 0]

92: LUN: 0, Sensor # 20 ("System Event")
    Type: Discrete (0x6f), "System Event" (0x12)
    Belongs to entity: (0xa0, 96) [FRU # 0]

92: LUN: 0, Sensor # 3 ("BMC Watchdog")
    Type: Discrete (0x6f), "Watchdog 2" (0x23)
    Belongs to entity: (0x3, 96) [FRU # 0]

92: LUN: 0, Sensor # 21 ("RTM Presence")
    Type: Discrete (0x6f), "Entity Presence" (0x25)
    Belongs to entity: (0xa0, 96) [FRU # 0]
#
```

● To Display Data from a Sensor on a Board

In the example, information about sensor number 3 (CPU2 Temp) on the board at IPMB address 92 is displayed.

```
# clia sensor 92 9
Pigeon Point Shelf Manager Command Line Interpreter

92: LUN: 0, Sensor # 9 ("+5.0V VCC")
    Type: Threshold (0x01), "Voltage" (0x02)
    Belongs to entity: (0x14, 96) [FRU # 0]
```

● To List All IPMCs in the Server

This example shows the typical output for the ipmc command.

```
# clia ipmc
Pigeon Point Shelf Manager Command Line Interpreter

10: Entity: (0xf0, 0x60) Maximum FRU device ID: 0x08
    PICMG Version 2.1
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)

20: Entity: (0xf0, 0x1) Maximum FRU device ID: 0x08
    PICMG Version 2.1
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
```

```
82: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x00
    PICMG Version 2.1
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)

88: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x00
   Hot Swap State: M7 (Communication Lost), Previous: M4 (Active),
Last State Change Cause: Communication Lost (0x4)

92: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x00
    PICMG Version 2.1
   Hot Swap State: M4 (Active), Previous: M7 (Communication Lost),
Last State Change Cause: Communication Lost (0x4)

96: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x00
    Hot Swap State: M7 (Communication Lost), Previous: M6
(Deactivation In Progress), Last State Change Cause: Communication
Lost (0x4)
20: Entity: (0xf0, 0x1) Maximum FRU device ID: 0x08
    PICMG Version 2.1
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
```

- **To Display Information About a Specific IPM Controller**

In the example, basic information about the IPM controller at address 9C is displayed.

```
# clia ipmc 9c
Pigeon Point Shelf Manager Command Line Interpreter

9c: Entity: (0xd0, 0x0) Maximum FRU device ID: 0x08
    PICMG Version 2.0
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
#
```

● To Display Detailed Information About the IPM Controller

In the example, detailed information about the IPM controller at address 9C is displayed.

```
# clia ipmc -v 9c
Pigeon Point Shelf Manager Command Line Interpreter

9c: Entity: (0xd0, 0x0) Maximum FRU device ID: 0x08
    PICMG Version 2.0
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID: 0x00, Revision: 0, Firmware: 1.01, IPMI ver 1.5
    Manufacturer ID: 00315a (PICMG), Product ID: 0000, Auxiliary
Rev: 01ac10ac
    Device ID String: "PPS Sentry 6"
    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"
#
```

Displaying FRU Information

You can display information about all the FRUs in the system by issuing the `clia fru` command without any parameters, or you can display information for specific FRU by supplying the FRU's address, and optionally the FRU ID. See "Physical Address to Logical Slot Mapping" on page 13 for a mapping of chassis slot number to physical address to IMPB address.

In the following example, in the line "`20: FRU # 1`", `20` is the midplane's IPMB address and `1` is the FRU device ID.

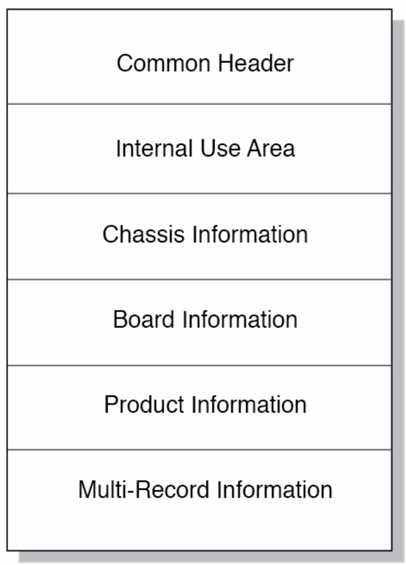
```
# clia fru
Pigeon Point Shelf Manager Command Line Interpreter

20: FRU # 1
 Entity: (0xf2, 0x60)
Hot Swap State: M4 (Active), Previous: M3 (Activation In Process),
Last State Change Cause: Normal State Change (0x0)
 Device ID String: "Shelf EEPROM 1"
```

IPMI FRU Information Layout

FIGURE 3-1 shows how the IPMI FRU information is organized. Each partition provides specific types of data.

- The Common Header contains area offsets.
- The Internal Use area is used to store proprietary data.
- The Chassis Information area contains chassis type, part number, serial number.
- The Board Information area contains manufacturer timestamp, manufacturer, product name, part/serial number.
- The Product Information area contains manufacturer, product name, part/serial number, version.
- The MultiRecord area contains dynamic data.

FIGURE 3-1 IPMI FRU Information Layout



Environment FRUs

Environment FRUs include midplane, shelf management cards, fans, power entry modules (PEMs), and the shelf alarm panel (SAP). All the environment FRUs have IPMI FRU information only, which are provided and programmed by third-party vendors.

The midplane FRU information includes the Sun part number as well as the third-party part number. Sun also adds additional system information such as slot, vlan, vtag, and other data, in the multirecord area of the midplane FRU information. The midplane FRU information is stored in two identical EEPROMs. Any change in one EEPROM is automatically made in the other EEPROM.

Blade FRUs

The Sun Netra CT900 server switch boards (located in slots 7 and 8) have IPMI FRU information only. The Sun node boards have two separate EEPROMS; one contains IPMI FRU information and the other contains Sun FRU information.

Examples

Examples for the following tasks are provided showing the commands used and their outputs.

- To Display Standard Information About All FRUs in the Shelf
- To Display Standard Information About All FRUs at Address 9C
- To Display Detailed Information About FRU #1 in the Shelf, Address 20
- To Display Detailed Information About FRU #0 on a Node Board at Address 8e.
- To Display FRU Information in Raw Form
- To Display FRU Information in User-Friendly Format

● To Display Standard Information About All FRUs in the Shelf

This example shows the standard FRU information displayed by the `fru` command with no arguments.

```
# clia fru
Pigeon Point Shelf Manager Command Line Interpreter

10: FRU # 0
    Entity: (0xf0, 0x60)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: "ShMM-500"

12: FRU # 0
    Entity: (0xf0, 0x60)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: "ShMM-500"

20: FRU # 0
    Entity: (0xf0, 0x1)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: "PPS BMC"

20: FRU # 1
    Entity: (0xf2, 0x60)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: "Shelf EEPROM 1"

20: FRU # 2
    Entity: (0xf2, 0x61)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: "Shelf EEPROM 2"

20: FRU # 3
    Entity: (0x1e, 0x0)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: "Fan Tray 0"

20: FRU # 4
    Entity: (0x1e, 0x1)
```

```
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: “Fan Tray 1”

20: FRU # 5
    Entity: (0x1e, 0x2)
    Device ID String: “Fan Tray 1” Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: “Fan Tray 2”

20: FRU # 6
    Entity: (0xa, 0x60)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: “PEM A”

20: FRU # 7
    Entity: (0xa, 0x61)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: “PEM B”

20: FRU # 8
    Entity: (0x7, 0x6f)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: “SAP Board”

82: FRU # 0
    Entity: (0xa0, 0x60)
    Hot Swap State: M4 (Active), Previous: M4 (Active, Last State
Change Cause: Normal State (0x0)
    Device ID String: “NetraCP-3140”

84: FRU # 0
    Entity: (0xa0, 0x60)
    Hot Swap State: M4 (Active), Previous: M4 (Active, Last State
Change Cause: Normal State Change (0x0)
  Device ID String: "NetraCP-3140"

8e: FRU # 0
    Entity: (0xa0, 0x60)
 Hot Swap State: M4 (Active), Previous: M4 (Active), Last State
Change Cause: Normal State Change (0x0)
    Device ID String: "NetraCP-3010
```

```
9a: 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: “NetraCP-3020”
```

● To Display Standard Information About All FRUs at Address 9C

In this example, only FRU information is displayed for all FRUs at physical address 9C.

```
# clia fru 9a
Pigeon Point Shelf Manager Command Line Interpreter

9a: FRU # 0
    Entity: (0ad0, 0x60)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: "NetraCP-3020"
#
```

● To Display Detailed Information About FRU #1 in the Shelf, Address 20

In this example, detailed FRU information is displayed for FRU device ID 1 on the chassis, physical address 20.

```
# clia fruinfo 20 1
Pigeon Point Shelf Manager Command Line Interpreter

20: FRU # 1, FRU Info
Common Header:    Format Version = 1

Internal Use Area:
    Version = 1
Chassis Info Area:
    Version   = 1
    Chassis Type              = (23)
    Chassis Part Number       = 11592-450
    Chassis Serial Number     =

Board Info Area:
    Version     = 1
    Language Code             = 25
```

```
   Mfg Date/Time          = Jun 16 00:00:00 2005 (4973760 minutes
since 1996)
    Board Manufacturer       = Schroff
    Board Product Name       = ShMM-ACB-III Shelf Manager (Radial
IPMB)
    Board Serial Number      = 0000001
    Board Part Number        = 21593-251
    FRU Programmer File ID   = Schroff_11592450_AA.inf

Product Info Area:
    Version     = 1
    Language Code            = 25
    Manufacturer Name        = Schroff
    Product Name             = 12U 14-Slot ATCA Chassis
    Product Part / Model#    = 11592-450
    Product Version          = Dual Star (Radial IPMB)
    Product Serial Number    = 0000001
    Asset Tag                =
    FRU Programmer File ID   = Schroff_11592450_AA.inf

Multi Record Area:
    PICMG Shelf Manager IP Connection Record (ID=0x13)
        Version = 1

    Record Type              = Management Access Record
        Version = 2
    Sub-Record Type: Component Name (0x05)

    PICMG Address Table Record (ID=0x10)
        Version = 0

    PICMG Backplane Point-to-Point Connectivity Record (ID=0x04)
        Version = 0

    PICMG Backplane Point-to-Point Connectivity Record (ID=0x04)
        Version = 0

    PICMG Shelf Activation And Power Management Record (ID=0x12)
        Version = 0

    PICMG Shelf Power Distribution Record (ID=0x11)
        Version = 0

    PICMG Radial IPMB-0 Link Mapping Record (ID=0x15)
        Version = 0
```

```
    Record Type              = 0xf0 OEM Defined Record
        Version = 2
    Manufacturer ID = 0x303833
```

To Display Detailed Information About FRU #0 on a Node Board at Address 8e.

In this example, detailed FRU information is displayed for FRU device ID `0` on a node board installed at physical address `8e`.

```
# clia fruinfo 8e 0
Pigeon Point Shelf Manager Command Line Interpreter

8e: FRU # 0, FRU Info
Common Header:    Format Version = 1

Board Info Area:
    Version     = 1
    Language Code            = 25
   Mfg Date/Time          = Mar 20 00:00:00 2006 (5372640 minutes
since 1996)
    Board Manufacturer       = Sun Microsystems, Inc.
    Board Product Name       = NetraCP-3060
    Board Serial Number      = 00000000000000000001
    Board Part Number        = 00000000000005017313
    FRU Programmer File ID   = fru-info.inf

Product Info Area:
    Version     = 1
    Language Code            = 25
    Manufacturer Name        = Sun Microsystems, Inc.
    Product Name             = NetraCP-3060
    Product Part / Model#    = 00000000000005017313
    Product Version          = Rev 1.00TEST
    Product Serial Number    = 00000000000000000001
    Asset Tag                =
    FRU Programmer File ID   = fru-info.inf

Multi Record Area:
    PICMG Board Point-to-Point Connectivity Record (ID=0x14)
        Version = 0

    AMC Carrier Information Table Record (ID=0x1a)
        Version = 0
```

```
    AMC Carrier Activation and Current Management Record (ID=0x17)
        Version = 0

    AMC Carrier Point-to-Point Connectivity Record (ID=0x18)
        Version = 0

    AMC Carrier Point-to-Point Connectivity Record (ID=0x19)
        Version = 0

    AMC Carrier Point-to-Point Connectivity Record (ID=0x19)
        Version = 0
```

● To Display FRU Information in Raw Form

In this example, FRU information is displayed in raw form for all FRUs and for a specific FRU.

```
# clia frudata
Pigeon Point Shelf Manager Command Line Interpreter

10: FRU # 0 Raw FRU Info Data
    FRU Info size: 435
12: FRU # 0 Raw FRU Info Data
    FRU Info size: 435
20: FRU # 0 Raw FRU Info Data
    FRU Info size: 152
20: FRU # 1 Raw FRU Info Data
    FRU Info size: 8192
20: FRU # 2 Raw FRU Info Data
    FRU Info size: 8192
20: FRU # 3 Raw FRU Info Data
    FRU Info size: 2048
20: FRU # 4 Raw FRU Info Data
    FRU Info size: 2048
20: FRU # 5 Raw FRU Info Data
    FRU Info size: 2048
20: FRU # 6 Raw FRU Info Data
    FRU Info size: 2048
20: FRU # 7 Raw FRU Info Data
    FRU Info size: 2048
20: FRU # 8 Raw FRU Info Data
    FRU Info size: 2048
20: FRU # 254 Raw FRU Info Data
    FRU Info size: 3068
```

```
9a: FRU # 0 Raw FRU Info Data
    FRU Info size: 512
#
# clia frudata 20 1 0
Pigeon Point Shelf Manager Command Line Interpreter

20: FRU # 1 Block # 0 Raw FRU Info Data
    FRU Info size: 8192
 01 01 22 24  31 3E 00 49  01 A0 A1 A2  A3 A4 A5 A6
 A7 A8 A9 AA  AB AC AD E0  E1 E2 E3 E4  E5 E6 D0 D1
```

- **To Display FRU Information in User-Friendly Format**

This example shows a user-friendly version of the FRU information.

```
# clia fruinfo 20 1
Pigeon Point Shelf Manager Command Line Interpreter

20: FRU # 1, FRU Info
Common Header:    Format Version = 1

Internal Use Area:
    Version = 1
Chassis Info Area:
    Version   = 1
    Chassis Type             = (23)
    Chassis Part Number      = 11592-450
    Chassis Serial Number    =

Board Info Area:
    Version     = 1
    Language Code            = 25
   Mfg Date/Time          = Jun 16 00:00:00 2005 (4973760 minutes
since 1996)
    Board Manufacturer       = Schroff
    Board Product Name       = ShMM-ACB-III Shelf Manager (Radial
IPMB)
    Board Serial Number      = 0000001
    Board Part Number        = 21593-251
    FRU Programmer File ID   = Schroff_11592450_AA.inf

Product Info Area:
    Version     = 1
    Language Code            = 25
    Manufacturer Name        = Schroff
    Product Name             = 12U 14-Slot ATCA Chassis
```

```
    Product Part / Model#     = 11592-450
    Product Version           = Dual Star (Radial IPMB)
    Product Serial Number     = 0000001
    Asset Tag                 =
    FRU Programmer File ID    = Schroff_11592450_AA.inf

Multi Record Area:
    PICMG Shelf Manager IP Connection Record (ID=0x13)
        Version = 1

    Record Type               = Management Access Record
        Version = 2
    Sub-Record Type: Component Name (0x05)

    PICMG Address Table Record (ID=0x10)
        Version = 0

    PICMG Backplane Point-to-Point Connectivity Record (ID=0x04)
        Version = 0

    PICMG Backplane Point-to-Point Connectivity Record (ID=0x04)
        Version = 0

    PICMG Shelf Activation And Power Management Record (ID=0x12)
        Version = 0

    PICMG Shelf Power Distribution Record (ID=0x11)
        Version = 0

    PICMG Radial IPMB-0 Link Mapping Record (ID=0x15)
        Version = 0

    Record Type               = 0xf0 OEM Record
        Version = 2
    UNKNOWN Manufacturer ID = 0x303833
```

Displaying Shelf Information

You can use the `clia shelf` command with one of its valid parameters to display FRU information for key shelf FRUs. You can also get current operating data for the shelf and modify some fields in the Shelf FRU information. The valid `shelf` parameters are:

- `cooling_state` or `cs`
- `fans_state` or `fs`
- `address_table` or `at`
- `power_distribution` or `pd`
- `power_management` or `pm`
- `pci_connectivity` or `pcic`
- `ha_connectivity` or `ha`
- `h110_connectivity` or `h1110c`
- `point-to-point_connectivity` or `ppc`

See "Display Shelf FRU Information" on page 308 for more information.

Examples

Examples for the following tasks are provided showing the commands used and their outputs.

- To Display Shelf Cooling Status
- To Display Shelf Fan Status
- To Display Address Table
- To Display Power Management Information
- To Display Power Distribution Information

To Display Shelf Cooling Status

This example shows the commands and outputs for displaying the shelf's cooling status.

```
# clia shelf cooling_state
Pigeon Point Shelf Manager Command Line Interpreter

    Cooling state: "Normal"

# clia shelf -v cooling_state
Pigeon Point Shelf Manager Command Line Interpreter

    Cooling state: "Normal"
    Sensor(s) at this state: (0x9a,4,0) (0x9a,5,0) (0x10,2,0) (0x9a,3,0)
                           (0x20,120,0) (0x20,121,0) (0x20,122,0) (0x20,123,0)
                           (0x20,200,0) (0x20,201,0) (0x20,240,0) (0x20,241,0)
                             (0x20,242,0)
```

To Display Shelf Fan Status

The commands to display the fan status and the outputs are in this example.

```
# clia shelf fans_state
Pigeon Point Shelf Manager Command Line Interpreter

    Fans state: "Normal"

# clia shelf -v fans_state
Pigeon Point Shelf Manager Command Line Interpreter

    Fans state: "Normal"
    Sensor(s) at this state: (0x10,7,0) (0x10,8,0) (0x10,9,0) (0x10,10,0)
                             (0x10,11,0) (0x10,12,0)
#
```

To Display Address Table

This example shows the command and output for displaying the shelf address table.

```
# clia shelf address_table
Pigeon Point Shelf Manager Command Line Interpreter

    PICMG Address Table Record (ID=0x10)
        Version = 0
    Shelf Address            = 1
    Address Table Entries#   = 16
        Hw Addr: 41, Site # 7, Type: "AdvancedTCA Board" 00
        Hw Addr: 42, Site # 8, Type: "AdvancedTCA Board" 00
        Hw Addr: 43, Site # 6, Type: "AdvancedTCA Board" 00
        Hw Addr: 44, Site # 9, Type: "AdvancedTCA Board" 00
        Hw Addr: 45, Site # 5, Type: "AdvancedTCA Board" 00
        Hw Addr: 46, Site # 10, Type: "AdvancedTCA Board" 00
        Hw Addr: 47, Site # 4, Type: "AdvancedTCA Board" 00
        Hw Addr: 48, Site # 11, Type: "AdvancedTCA Board" 00
        Hw Addr: 49, Site # 3, Type: "AdvancedTCA Board" 00
        Hw Addr: 4a, Site # 12, Type: "AdvancedTCA Board" 00
        Hw Addr: 4b, Site # 2, Type: "AdvancedTCA Board" 00
        Hw Addr: 4c, Site # 13, Type: "AdvancedTCA Board" 00
        Hw Addr: 4d, Site # 1, Type: "AdvancedTCA Board" 00
        Hw Addr: 4e, Site # 14, Type: "AdvancedTCA Board" 00
        Hw Addr: 08, Site # 1, Type: "Dedicated ShMC" 03
        Hw Addr: 09, Site # 2, Type: "Dedicated ShMC" 03
```

To Display Power Management Information

An example of the command and its output are shown in this example.

```
# clia shelf pm
Pigeon Point Shelf Manager Command Line Interpreter

PICMG Shelf Activation And Power Management Record (ID=0x12)
        Version = 0
    Allowance for FRU Activation Readiness: 20 seconds
    FRU Activation and Power Description Count: 19
    Hw Address: 41, FRU ID: 0xfe, Maximum FRU Power Capabilities: 200 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 42, FRU ID: 0xfe, Maximum FRU Power Capabilities: 200 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds
```

```
Hw Address: 43, FRU ID: 0xfe, Maximum FRU Power Capabilities: 200 Watts
    Shelf Manager Controlled Activation: Enabled
    Delay Before Next Power On: 0.0 seconds

Hw Address: 44, FRU ID: 0xfe, Maximum FRU Power Capabilities: 200 Watts
    Shelf Manager Controlled Activation: Enabled
    Delay Before Next Power On: 0.0 seconds

Hw Address: 45, FRU ID: 0xfe, Maximum FRU Power Capabilities: 200 Watts
    Shelf Manager Controlled Activation: Enabled
    Delay Before Next Power On: 0.0 seconds

Hw Address: 46, FRU ID: 0xfe, Maximum FRU Power Capabilities: 200 Watts
    Shelf Manager Controlled Activation: Enabled
    Delay Before Next Power On: 0.0 seconds

Hw Address: 47, FRU ID: 0xfe, Maximum FRU Power Capabilities: 200 Watts
    Shelf Manager Controlled Activation: Enabled
    Delay Before Next Power On: 0.0 seconds

Hw Address: 48, FRU ID: 0xfe, Maximum FRU Power Capabilities: 200 Watts
    Shelf Manager Controlled Activation: Enabled
    Delay Before Next Power On: 0.0 seconds

Hw Address: 49, FRU ID: 0xfe, Maximum FRU Power Capabilities: 200 Watts
    Shelf Manager Controlled Activation: Enabled
    Delay Before Next Power On: 0.0 seconds

Hw Address: 4a, FRU ID: 0xfe, Maximum FRU Power Capabilities: 200 Watts
    Shelf Manager Controlled Activation: Enabled
    Delay Before Next Power On: 0.0 seconds

Hw Address: 4b, FRU ID: 0xfe, Maximum FRU Power Capabilities: 200 Watts
    Shelf Manager Controlled Activation: Enabled
    Delay Before Next Power On: 0.0 seconds

Hw Address: 4c, FRU ID: 0xfe, Maximum FRU Power Capabilities: 200 Watts
    Shelf Manager Controlled Activation: Enabled
    Delay Before Next Power On: 0.0 seconds

Hw Address: 4d, FRU ID: 0xfe, Maximum FRU Power Capabilities: 200 Watts
    Shelf Manager Controlled Activation: Enabled
    Delay Before Next Power On: 0.0 seconds

Hw Address: 4e, FRU ID: 0xfe, Maximum FRU Power Capabilities: 200 Watts
    Shelf Manager Controlled Activation: Enabled
    Delay Before Next Power On: 0.0 seconds

Hw Address: 44, FRU ID: 0xfe, Maximum FRU Power Capabilities: 200 Watts
```

```
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 45, FRU ID: 0xfe, Maximum FRU Power Capabilities: 200 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 08, FRU ID: 0xfe, Maximum FRU Power Capabilities: 24 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 09, FRU ID: 0xfe, Maximum FRU Power Capabilities: 24 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 20, FRU ID: 0xfe, Maximum FRU Power Capabilities: 100 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

#
```

- **To Display Power Distribution Information**

The example shows the command and output for getting the shelf's power distribution information.

```
# clia shelf pd
Pigeon Point Shelf Manager Command Line Interpreter

PICMG Shelf Power Distribution Record (ID=0x11)
        Version = 0
    Feed count: 8
    Feed 00:
        Maximum External Available Current: 25.0 Amps
        Maximum Internal Current: 25.0 Amps
        Minimum Expected Operating Voltage: -40.5 Volts
        Actual Power Available: 1012.500 Watts
        Currently Used Power: 30.000 Watts
        Feed-to-FRU Mapping entries count: 3
           FRU Addr: 49, FRU ID: 0xfe
           FRU Addr: 4b, FRU ID: 0xfe
           FRU Addr: 4d, FRU ID: 0xfe
    Feed 01:
        Maximum External Available Current: 25.0 Amps
        Maximum Internal Current: 25.0 Amps
        Minimum Expected Operating Voltage: -40.5 Volts
        Actual Power Available: 1012.500 Watts
```

```
        Currently Used Power: 30.000 Watts
        Feed-to-FRU Mapping entries count: 3
           FRU Addr: 49, FRU ID: 0xfee
           FRU Addr: 4b, FRU ID: 0xfe
           FRU Addr: 4d, FRU ID: 0xfe
    Feed 02:
        Maximum External Available Current: 25.0 Amps
        Maximum Internal Current: 25.0 Amps
        Minimum Expected Operating Voltage: -40.5 Volts
        Actual Power Available: 1012.500 Watts
        Currently Used Power: 40.000 Watts
        Feed-to-FRU Mapping entries count: 4
           FRU Addr: 41, FRU ID: 0xfe
           FRU Addr: 43, FRU ID: 0xfe
           FRU Addr: 45, FRU ID: 0xfe
           FRU Addr: 47, FRU ID: 0xfe
    Feed 03:
        Maximum External Available Current: 25.0 Amps
        Maximum Internal Current: 25.0 Amps
        Minimum Expected Operating Voltage: -40.5 Volts
        Actual Power Available: 1012.500 Watts
        Currently Used Power: 40.000 Watts
        Feed-to-FRU Mapping entries count: 4
           FRU Addr: 41, FRU ID: 0xfe
           FRU Addr: 43, FRU ID: 0xfe
           FRU Addr: 45, FRU ID: 0xfe
           FRU Addr: 47, FRU ID: 0xfe
        Maximum External Available Current: 25.0 Amps
        Maximum Internal Current: 25.0 Amps
        Minimum Expected Operating Voltage: -40.5 Volts
        Actual Power Available: 1012.500 Watts
        Currently Used Power: 40.000 Watts
        Feed-to-FRU Mapping entries count: 4
           FRU Addr: 42, FRU ID: 0xfe
           FRU Addr: 44, FRU ID: 0xfe
           FRU Addr: 46, FRU ID: 0xfe
           FRU Addr: 48, FRU ID: 0xfe
    Feed 05:
        Maximum External Available Current: 25.0 Amps
        Maximum Internal Current: 25.0 Amps
        Minimum Expected Operating Voltage: -40.5 Volts
        Actual Power Available: 1012.500 Watts
        Currently Used Power: 40.000 Watts
        Feed-to-FRU Mapping entries count: 4
           FRU Addr: 42, FRU ID: 0xfe
```

```
            FRU Addr: 44, FRU ID: 0xfe
            FRU Addr: 46, FRU ID: 0xfe
            FRU Addr: 48, FRU ID: 0xfe
     Feed 06:
         Maximum External Available Current: 25.0 Amps
         Maximum Internal Current: 25.0 Amps
         Minimum Expected Operating Voltage: -40.5 Volts
         Actual Power Available: 1012.500 Watts
         Currently Used Power: 100.000 Watts
         Feed-to-FRU Mapping entries count: 6
            FRU Addr: 08, FRU ID: 0xfe
            FRU Addr: 09, FRU ID: 0xfe
            FRU Addr: 20, FRU ID: 0xfe
            FRU Addr: 4a, FRU ID: 0xfe
            FRU Addr: 4c, FRU ID: 0xfe
            FRU Addr: 4e, FRU ID: 0xfe
     Feed 07:
         Maximum External Available Current: 25.0 Amps
         Maximum Internal Current: 25.0 Amps
         Minimum Expected Operating Voltage: -40.5 Volts
         Actual Power Available: 1012.500 Watts
         Currently Used Power: 100.000 Watts
         Feed-to-FRU Mapping entries count: 6
            FRU Addr: 08, FRU ID: 0xfe
            FRU Addr: 09, FRU ID: 0xfe
            FRU Addr: 20, FRU ID: 0xfe
            FRU Addr: 4a, FRU ID: 0xfe
            FRU Addr: 4c, FRU ID: 0xfe
            FRU Addr: 4e, FRU ID: 0xfe
#
```

Re-initializing the Shelf Manager

This section describes how to re-initialize the U-Boot environment variables, the file system in Flash memory, and the login password on the shelf management card.

Re-initialize the U-Boot Environment

The U-Boot environment variables are stored in the shelf management card EEPROM. To restore the factory defaults for the U-Boot environment variables, you must first erase the environment variables stored in EEPROM and reset (or power cycle) the shelf management card.

Caution – U-boot re-initialization can cause the backup ShMM to continually reboot because some critical variables are being reset. To prevent this, make a copy of existing variables before re-initialization and apply them after re-initialization.

For lists of variables that need to be restored, see the list the variables that should only be changed for debugging by advanced users and the list variables should not be changed list in "Recommendations for Changing U-Boot Variables" on page 24.

▼ To Re-initializing the U-Boot Environment

1. **Erase the EEPROM by entering the following command from the U-Boot prompt:**

```
shmm500 eeprom write 80400000 0 1000

EEPROM @0x50 write: addr 80400000  off 0000  count 4096 ... done
```

2. **Reset the shelf management card as follows:**

```
shmm500 reset
U-Boot 1.1.2 (Nov 27 2005 - 19:17:09)

CPU: Au1550 324 MHz, id: 0x02, rev: 0x00
Board: ShMM-500
S/N: 8000041
DRAM:  128 MB
Flash: 64 MB
*** Warning - bad CRC, using default environment

In:    serial
Out:   serial
Err:   serial
Net:   Au1X00 ETHERNET
Hit any key to stop autoboot:  0

YOU MUST HIT ANY KEY NOW TO BREAK THE BOOT PROCESS
```

3. **Set default for the Sun environment**

```
shmm500 setenv rc2 /etc/rc.acb3
```

4. **Save these environment settings using the following commands:**

   The `saveenv` command must be issued twice as shown.

```
shmm500 saveenv
shmm500 saveenv
```

5. **Reboot the shelf manager card using the following command:**

```
shmm500 boot
```

Re-initializing the File System

The file system is stored within the Flash memory and can be reset to factory defaults quite easily. U-Boot has an environment variable called `flash_reset`. Setting this variable to `y` and then booting the system will re-initialized the file system to the factory defaults.

```
# setenv flash_reset y
# boot
```

The `flash_reset` variable is automatically set to `n` at system startup after the Flash memory is re-initialized. The `bootcmd` command will begin booting the Linux kernel. It is during this process that the file system is re-initialized. The following output will be shown on the console.

```
/etc/rc: Mounted /dev/pts
/etc/rc: Flash erase requested via U-BOOT var
/etc/rc: erasing mtdchar1 -> /etc
Erased 1024 Kibyte @ 0 -- 100% complete.
/etc/rc: erasing mtdchar0 -> /var
Erased 1536 Kibyte @ 0 -- 100% complete.
/etc/rc: Mounted /dev/mtdblock3 to /var
/etc/rc: /var/log mounted as FLASH disk
/etc/rc: Started syslogd and klogd
/etc/rc: /var/tmp mounted as RAM disk
/etc/rc: hostname demo
/etc/rc: /dev/mtdblock2 appears to be empty ... restoring from
factory /etc...
```

Resetting the Login Password

The factory default login for the shelf management card is userid `root` and password `sunct900`. Users are encouraged to change the password when configuring the Shelf Manager. In the event that the new password is forgotten, the password can be reset to its factory default via the `password_reset` U-Boot variable. Setting this variable to `y` and then booting the system will remove the `root` password.

```
# setenv password_reset y
# boot
```

The following output will be displayed on the console, during boot.

```
/etc/rc: hostname demo
```

Reprogramming the Shelf Management Card

Reprogramming the shelf management card involves reprogramming several firmware images stored on it. The set of images can be conceptually divided into two groups, depending on the method of reprogramming them. These groups are:

- U-Boot, Linux kernel, and Linux root file system (RFS) images

  These images are normally reprogrammed using the reliable upgrade procedure. Alternatively, the kernel and RFS can be reprogrammed from the U-Boot prompt by loading them from a TFTP server.

- Complex Programmable Logic Device (CPLD) image

  This image is reprogrammed from the shelf management card command line, using a special command-line utility `cpldtool`.

Detailed instructions for reprogramming these images are given below

Firmware Reliable Upgrade Procedure

The Monterey Linux operating system provides a reliable upgrade procedure for the firmware images on a running and functioning shelf management card. The procedure supports upgrade of the U-Boot firmware, the Linux kernel, and the Linux RFS (or an arbitrary combination of these three images). If a software upgrade attempt fails (for instance, due to installation of a faulty U-Boot firmware image, incapable of booting the shelf management card or a Shelf Manager that can't start) the reliable upgrade procedure automatically falls back to the previous version of the firmware in persistent Flash memory.

The shelf management card Flash memory is divided into two areas. When a stable set of firmware is established in one of these areas, it is designated the *persistent* area. When new firmware is installed, it goes in the other area, which is initially designated *provisional*. Once a new set of firmware in the provisional area is validated, that area is designated the persistent area and continues in use until a future upgrade cycle starts the process over.

The reliable upgrade hardware mechanisms ensure that no matter what is installed to the provisional Flash, the shelf management card always manages to boot from a software copy that is either fully functional or sufficiently sane to determine that there has been a failure in the upgrade session and consequently take appropriate corrective actions to revert to the safe software copy in persistent Flash.

At a higher level, the reliable upgrade hardware mechanisms are assisted by a software protocol based on logging of the status of the upgrade session to a non-volatile file in `/var/upgrade/status` (see "Reliable Upgrade Procedure Status File" on page 129). The software protocol ensures that the reliable upgrade does not finish until all the required actions, including those defined by custom *hook* scripts that might be needed for a specific application, have completed successfully.

The reliable upgrade procedure implemented in Monterey Linux is neutral to the embedded application running on the shelf management card. The procedure provides a sufficient set of "hooks" allowing a specific application running on the shelf management card to ensure that custom actions are carried out at appropriate points of the reliable upgrade. The remainder of this section focuses on provisions for reliable upgrade of the Pigeon Point Shelf Manager firmware that have been implemented using these hooks.

Flash Partitioning

The shelf management card provides a hardware mechanism that allows swapping of the lower and upper halves of the Flash in the system memory map under control of software running on the MIPS processor. This capability is implemented in support of the reliable upgrade procedure for software images in Flash. The reliable software upgrade procedure assumes that the Flash device contains two copies of the software, located in the lower and upper halves of Flash. All shelf management cards are shipped with the partitioning, where the Flash device is divided onto two equal parts, each dedicated to holding one copy of the Shelf Manager software.

The U-Boot environment variable `reliable_upgrade` (see "U-Boot Environment Variables" on page 19) is used by the Linux layers to determine whether or not the reliable upgrade procedure is enabled. This variable must be set to `y` and is passed to the Linux kernel in the `bootargs` kernel parameters string. The Linux Flash MTD layer checks the `reliable_upgrade` parameter at Flash partitions initialization time and, depending on the parameter value (as well as the size of the Flash device installed on the shelf management card), partitions the Flash device in an appropriate way.

This section assumes that the shelf management card is configured to support the reliable upgrade, including the two separate Flash regions. TABLE 3-4 provides a summary of the Flash partitions maintained on the shelf management card in this configuration (`reliable_upgrade=y`):

TABLE 3-4 Flash Partitions for 16MB Flash

Offset in Flash (in MBytes)	Size (in MBytes)	Device Node	Mounted As (on Startup)	Content
0	0.5	`/dev/mtdchar10`, `/dev/mtdblock10`	`/var/upgrade`	The second half of the `/var/upgrade` JFFS2 file system
0.5 +(FLASH_SIZE –16)/2	1.5	`/dev/mtdchar5`, `/dev/mtdblock5`	Not mounted	The other `/var` JFFS2 file system
FLASH_SIZE/2 – 62	1	`/dev/mtdchar6`, `/dev/mtdblock6`	Not mounted	The other `/etc` JFFS2 file system
FLASH_SIZE/2 – 53	1	`/dev/mtdchar7`	Not mounted	The other Linux kernel image
FLASH_SIZE/2 – 44	0.25	`/dev/mtdchar8`	Not mounted	The other U-Boot firmware image
FLASH_SIZE/2 – 3.754.25	3.75	`/dev/mtdchar9`	Not mounted	The other Linux root file system (rfs) image
FLASH_SIZE/28	0.5	`/dev/mtdchar10`, `/dev/mtdblock10`	`/var/upgrade`	The first half of the `/var/upgrade` JFFS2 file system
FLASH_SIZE – 7.58.5	1	`/dev/mtdchar0`, `/dev/mtdblock0`	`/var`	The `/var` JFFS2 file system
FLASH_SIZE – 610	1	`/dev/mtdchar1`, `/dev/mtdblock1`	`/etc`	The `/etc` JFFS2 file system
FLASH_SIZE – 5.11	1	`/dev/mtdchar2`	Not mounted	The Linux kernel image
FLASH_SIZE – 412	0.25	`/dev/mtdchar3`	Not mounted	The U-Boot firmware image
FLASH_SIZE – 3.712.255	3.75	`/dev/mtdchar4`	Not mounted	The Linux root file system (rfs) image

TABLE 3-5 provides a summary of the Flash partitions maintained on the shelf management card for the 64MB Flash devices:

TABLE 3-5 Flash Partitions for 64MB Flash

Offset in Flash (in MBytes)	Size (in MBytes)	Device Node	Mounted As (on Startup)	Content
0	0.5	/dev/mtdchar10, /dev/mtdblock10	/var/upgrade	The second half of the /var/upgrade JFFS2 file system
0.5	1	/dev/mtdchar7	Not mounted	The "other" Linux kernel image
1.5	1	/dev/mtdchar6, /dev/mtdblock6	Not mounted	The "other" /etc JFFS2 file system
2.5	1.75	/dev/mtdchar5, /dev/mtdblock5	Not mounted	The "other" /var JFFS2 file system
4.25	15.75	/dev/mtdchar9	Not mounted	The "other" Linux root file system (rfs) image
20	8	/dev/mtdchar12, /dev/mtdblock12	Not mounted	The second half of the app1_jffs application-specific JFFS2 partition
28	0.25	/dev/mtdchar8	Not mounted	The "other" U-Boot firmware image
28.25	3.75	/dev/mtdchar11, /dev/mtdblock11	Not mounted	The second half of the app_jffs application-specific JFFS2 partition
32	0.5	/dev/mtdchar10, /dev/mtdblock10	/var/upgrade	The first half of the /var/upgrade JFFS2 file system
32.5	1	/dev/mtdchar2	Not mounted	The Linux kernel
33.5	1	/dev/mtdchar1, /dev/mtdblock1	/etc	The /etc JFFS2 file system
34.5	1.75	/dev/mtdchar0, /dev/mtdblock0	/var	The /var JFFS2 file system
36.25	15.75	/dev/mtdchar4	Not mounted	The Linux root file system (rfs) image
52	8	/dev/mtdchar12, /dev/mtdblock12	Not mounted	The first half of the app1_jffs application-specific JFFS2 partition
60	0.25	/dev/mtdchar3	Not mounted	The U-Boot firmware image
60.25	3.75	/dev/mtdchar11, /dev/mtdblock11	Not mounted	The second half of the app_jffs application-specific JFFS2 partition

The `/var/upgrade` File System

As documented in "Firmware Reliable Upgrade Procedure" on page 125, if `reliable_upgrade` is `y`, Monterey Linux mounts a 1-MByte partition as a JFFS2 file system at `/var/upgrade`. This file system is used to host the reliable upgrade procedure status file (see "Reliable Upgrade Procedure Status File" on page 129).

It is important to note that the `/var/upgrade` JFFS2 partition is composed of two non-contiguous Flash blocks (0.5 MByte each), one residing in the lower half and one in upper half of the Flash device. Monterey Linux takes advantage of the ability of the Linux MTD and JFFS2 layers to support a file system in non-contiguous Flash sectors in order to implement `/var/upgrade` this way.

Another feature of the JFFS2 file system that makes `/var/upgrade` work for the purposes of the reliable upgrade procedure is that the JFFS2 internal structures do not create any dependencies (such as linked lists) based on Flash sector numbers or absolute offsets in Flash. Instead, when mounting a file system on a partition, the JFFS2 scans all the Flash sectors making up the partition and recreates the logical content of a file system in an internal in-RAM representation. This feature ensures that regardless of which half of the Flash the ShMM has booted from, Linux is able to mount `/var/upgrade` as a JFFS2 file system and make use of the previous content of the file system.

Reliable Upgrade Procedure Status File

The software reliable upgrade procedure maintains the status of the most recent upgrade procedure session in the file `/var/upgrade/status` residing in a dedicated file system (`/var/upgrade`), which is mounted by Linux regardless of which Flash the ShMM has booted from. If the file exists, it contains the status of an upgrade procedure session that either is in progress presently or has recently completed.

`/var/upgrade/status` is an ASCII-format file that contains one or more new line-terminated records, each describing the status of a particular step in the upgrade procedure. The format of a record line is as follows:

<*step*>: <*status*>

where *step* is an integer ranging from 1 to 14 (with Step 14 corresponding to a completed upgrade session) and *status* is a human-readable string describing status of the current step of the upgrade procedure session.

The status file is used by the reliable upgrade utility (see "Reliable Upgrade Utility" on page 130) to maintain a software protocol atop the reliable upgrade procedure hardware mechanisms to reliably determine the status of the upgrade procedure and proceed as appropriate.

Reliable Upgrade Utility

A special user-space, reliable upgrade utility is provided for carrying out the reliable upgrade procedure as well as checking the status of the most recent upgrade.

The utility can be called only from the superuser (`root`) account. Any attempt to run the utility from a non-superuser account is rejected.

As a first step in its execution, the utility checks that the `reliable_upgrade` U-Boot environment variable (see "Flash Partitioning" on page 126), as passed by U-Boot to the Linux kernel in the kernel parameters string, is set to `y`. If this check fails, the utility immediately terminates and exits with an appropriate error code.

If called with any of the `-s`, `-c`, or `-f` options, the utility is being used to carry out the reliable upgrade procedure. While in the upgrade procedure, the utility logs to `/var/upgrade/status` the status of each action it performs as it proceeds through the steps of the upgrade procedure. If the utility detects a failure, the reliable upgrade procedure is terminated by adding to `/var/upgrade/status` a record indicating an unsuccessful completion of the upgrade procedure and exiting with an appropriate error code.

The utility prints any informational messages to `stdout`. Providing `-v` specifier to any option that supports it increases the verbosity of the informational messages. The utility prints any error messages to `stderr`.

The utility has the following syntax:

- `rupgrade_tool -s {--dst=src}...` [`--proto=protocol`] [`-d`] [`--hook=args`] [`-v`]
- `rupgrade_tool -c [-v]`
- `rupgrade_tool -f [--hook=args] [-v]`
- `rupgrade_tool -w [-f]`
- `rupgrade_tool -S [-v]`
- `rupgrade_tool -u`
- `rupgrade_tool -h`

where the parameters are defined as follows:

`-s {--dst=src}... [--proto=protocol] [--hook=args] [-v]`

Initiate the reliable upgrade procedure. As delivered with Shelf Manager support, this step includes the following actions:

- Obtaining the images to copy, locally or via the network
- Copying the images to the provisional Flash
- Terminating the Shelf Manager instance running on the ShMM, if any
- Copying non-volatile data to the provisional Flash

- Resetting the shelf management card and instructing it to boot from the provisional Flash

  Because of the last step, an invocation of `rupgrade_tool -s` typically does not return and instead resets the shelf management card. If `rupgrade_tool -s` does return, it indicates that the reliable upgrade procedure has failed and was terminated before proceeding to reset the shelf management card in order to boot from the provisional Flash.

  Before the first step of the upgrade procedure is initiated by the utility, it removes the `/var/upgrade/status` file (see "Reliable Upgrade Procedure Status File" on page 129). In other words, the status of the previous upgrade procedure session (if any) is lost and overwritten by the status of the new upgrade procedure session as soon as `rupgrade_tool -s` is called.

  There can be one or more *--dst=src* specifiers in a call to `rupgrade_tool -s`. Each such specifier defines the name of a to-be-installed upgrade image file and where the file is to be installed in the Flash of the shelf management card.

  dst defines the destination of a newly installed upgrade image and can be any of the following:

  - `u` – Upgrade the U-Boot image in the provisional U-Boot firmware image partition (`/dev/mtdchar3`).
  - `k` – Upgrade the Linux kernel image in the provisional Linux kernel image partition (`/dev/mtdchar2).`
  - `r` – Upgrade the root file system image in the provisional root file system image partition (`/dev/mtdchar4`).

  src specifies an upgrade image file to be copied to the provisional Flash partition specified by *dst*.

  The image upgrade works as follows. For each of the specified *src* images, the image is copied to the shelf management card using the specified copy protocol. If no `-d` specifier is supplied, the image is first copied to the RAM file system of the shelf management card (specifically, the copy is to the `/tmp` directory) and then moved to Flash (that is, copied to the destination partition in Flash and then removed from the RAM file system). If a `-d` specifier is supplied in the call to `rupgrade_tool -s`, the intermediate copy to the `/tmp` directory is skipped and the image is copied directly to its destination in the Flash. Use of this specifier is intended for a scenario where there is insufficient runtime memory on the shelf management card for an intermediate copy to the RAM file system.

  If no `-d` specifier is supplied, the reliable upgrade procedure invokes a special script, the main purpose of which is to validate images after they are copied to the `/tmp` directory. If `-d` specifier is present, no such validation is performed.

  Currently, the script `/etc/upgrade/step4vshm` supplied with the Shelf Manager does not perform specific image validation steps, but does take responsibility for filling in the Flash partitions for which no images are provided in the current call to `rupgrade_tool` (as would happen in a partial upgrade scenario). These partitions are copied from the current persistent Flash to the provisional Flash. For example, if the current partial upgrade provides only a new RFS image, the script copies the U-Boot and kernel partitions from the persistent Flash to the provisional Flash.

As soon as the first image has been installed to its destination, the utility proceeds to the second image (if there is one), and so on, until all the supplied image files have been successfully installed to Flash. A failure to successfully install an image immediately terminates the upgrade procedure (vs. skipping a failing image and proceeding to the next one).

This approach enables the user to separately upgrade the three parts of shelf management card firmware (U-Boot, kernel, and RFS image). However, bear in mind that the parts that are not explicitly updated will be copied from persistent Flash.

It is recommended that you use of one of the following upgrade approaches:

- Explicitly upgrade all three partitions.
- When fewer than three partitions are explicitly upgraded, omit the `-d` specifier; in that case, the special script mentioned above will automatically ensure that every upgrade is effectively a full upgrade covering all three partitions.

protocol specifies a file copy protocol used to pull each of the specified *src* files to the Shelf Manager and can be any of the following:

- `no` – No copy is performed. This protocol assumes that all of the specified *src* files were pushed to the `/tmp` directory prior to start of the reliable upgrade procedure. This protocol choice cannot be used in conjunction with the `-d` option.
- `cp:`*dir* – Simple copy. This protocol assumes that all of the specified *src* files are to be copied from the specified directory in the Shelf Manager local file system by the `cp` command. This protocol can be useful, for instance, for installation of upgrade images from an NFS-mounted file system or even from a JFFS2 file system.
- ftp:*server*:*dir*:*user*[:*pwd*] – Copy from a remote FTP server. This protocol assumes that all of the specified *src* files are to be copied to the shelf management card from the FTP server host specified as the host name or the IP address by server. All the images must reside in the directory specified by *dir* on the remote FTP server. The FTP connection is made using the account specified by the user parameter, with the password specified by the optional `pwd` parameter. If no `pwd` is supplied, the utility will prompt for a password.

A failure in copying an image to the Shelf Manager causes the utility to terminate the upgrade procedure (vs. skipping a failing image and proceeding to the next one).

For each provisional Flash partition upgraded by the `-s` option, the partition to be upgraded is given write permissions after the validity of the image has been checked and immediately before the *src* image is moved to the Flash. Write permissions are removed from the partition immediately after the full image has been moved to Flash. Combined with the fact that all the partitions containing the U-Boot, Linux kernel, and root file system images are read-only on boot of the Shelf Manager, this ensures that applications cannot accidentally erase the critical boot images.

After all the specified images have been installed to their respective destinations in Flash memory, the utility invokes a *hook* script that enables custom actions required by an application at the point where the upgrade images have been already installed in Flash but the upgrade procedure has not yet initiated the hardware mechanisms of the reliable upgrade procedure by enabling the ShMM's upgrade watchdog timer (WDT).

The hook script, `/etc/upgrade/step4hshm`, is supplied with the Shelf Manager. It performs the following actions:

- Terminates the Shelf Manager, performing a switchover to the backup ShMM without restarting the shelf; the ATCA watchdog timer is stopped.
- Mounts the provisional `/etc` and `/var` Flash partitions and erases all files on them.
- Optionally copies the current contents of the `/etc` directory to the provisional `/etc` Flash partition.
- Optionally copies the current non-volatile Shelf Manager information from the directory `/var/nvdata` to the provisional `/var` file system; or optionally copies the whole `/var` directory to the provisional `/var` Flash partition.
- Temporarily (until next boot) sets the boot delay to `0`; this is done to minimize the time of the next boot and prevent the reliable upgrade watchdog timer from premature expiration.

This script is invoked as a sub-shell and given a single parameter, which is either the string specified by *args* or no *args* (an empty string). The given parameter defines the script's operational mode which determines operations like the copying of non-volatile information from the persistent Flash partitions to the provisional Flash partitions. The script takes the following *args* values, and perform the corresponding actions:

- No parameter supplied – the script erases both the provisional `/etc` and provisional `/var` directories, then copies Shelf Manager non-volatile information from the directory `/var/nvdata` to the provisional `/var` partition. This is the default mode of operation; in this case, the non-volatile data will be preserved but the Shelf Manager configuration file will be taken from the new RFS image.
- `erase` – the script erases both the provisional `/etc` and provisional `/var` directories; they will be restored from the RFS default values during the next boot; the current Shelf Manager non-volatile data and configurations are not preserved.

- etc_copy – the script erases both the provisional /etc and provisional /var directories, then it copies the contents of /etc and the non-volatile information from the directory /var/nvdata to the provisional Flash partitions. In this case, both the non-volatile data and the Shelf Manager configuration file are preserved.
- copy – the script erases both the provisional /etc and provisional /var directories, then copies the full contents of the /etc and /var directories onto the provisional partition. In this case, not only the configuration, but also the executable files placed to /var/bin will be copied and will override executable files with the same name from the RFS image. This mode of operation is useful if the directory /var/bin contains some special executable files (for example, a special version of the Shelf Manager or other utilities) that must be preserved across the upgrade.

The script returns 0 on success and a non-zero value for failure. If a non-zero value is returned, the upgrade procedure is terminated.

The utility starts the upgrade WDT with a 12.8-second time-out period. This time-out period is considered sufficient for any software that will boot after the reset to proceed to the point where it is able to call rupgrade_tool -c (which strobes the upgrade WDT in case it is active) without having to strobe the upgrade WDT in the interim. The utility performs a strobe of the upgrade WDT just before resetting the ShMM.

- -c [-v]

Proceed with the reliable upgrade procedure after the ShMM is booted from the provisional Flash. The invocation of rupgrade_tool -c is done from the /etc/rc script. As described below, certain situations discovered by rupgrade_tool -c imply a failure in the upgrade procedure and require corrective actions, including those resulting in the need to soft reset the ShMM. This means that an invocation of rupgrade_tool -c may not return and instead may result in reset of the ShMM. If a reset takes place, it reverts the ShMM to the software installed in the persistent Flash.

If the upgrade WDT is active and has fired at any step prior to invocation of rupgrade -c, this means that the ShMM already reverted to the software in the persistent Flash. In this scenario, the utility disables the upgrade WDT and returns to the use of persistent Flash and terminates the upgrade procedure.

If the upgrade WDT is active but has not fired, this means that the ShMM successfully booted (up to this point) from the provisional Flash. The utility strobes the upgrade WDT and exits with the return code of 0, indicating that there is an upgrade procedure session in progress.

If the upgrade WDT is not active but the content of the /var/upgrade/status file indicates that the upgrade procedure is still in progress, this means that the ShMM rebooted due to a power-cycle at one of the steps of the upgrade procedure. In this scenario the utility performs the same corrective actions as for the situation when the upgrade WDT is active and has fired.

Finally, if the upgrade WDT is not active and `/var/upgrade/status` is either not present or indicates that the upgrade procedure has finished (either successfully or unsuccessfully), the utility exits with the return value of `1`, indicating that there is no upgrade procedure in progress.

- `-f [--hook=args] [-v]`

  Complete the upgrade procedure. The invocation of `rupgrade_tool -f` is done from inside the Shelf Manager after the Shelf Manager successfully completes its initialization. If the Shelf Manager is not started automatically, that invocation is done at the end of the `/etc/rc` script.

  As soon as it is invoked, `rupgrade_tool -f` strobes the upgrade WDT and proceeds with establishing the new persistent Flash and disabling the upgrade WDT.

  Before completion, the utility updates `/var/upgrade/status` with a record indicating a successful completion of the upgrade procedure then exits with a value of `0`.

- `-w [-f]`

  Print the current status of the most recent upgrade procedure. Essentially, this option dumps the content of the `/var/upgrade/status` file to `stdout`.

  `rupgrade_tool -w` returns a value of `0` if the upgrade procedure has completed successfully, `1` if the upgrade procedure was unsuccessful, and an error code if there is no `/var/upgrade/status`.

  If the `-f` specifier is supplied, `rupgrade_tool -w` removes the `/var/upgrade/status` file before exiting.

- `-S [-v]`

  Strobe the upgrade WDT. `rupgrade_tool -S` is intended as a shell-level interface for use by newly installed software that is validating its sanity.

  `rupgrade_tool -S` returns a value of `0`.

- `-u`

  Undo a successful upgrade session, reverting to the previous persistent Flash device.

  `rupgrade_tool -u` causes the ShMM to reboot.

- `-h`

  Print help to `stdout`.

Reliable Upgrade Utility Use Scenarios

It is intended that the reliable upgrade utility will be used in the following sequence to carry out an upgrade of the ShMM:

1. The user makes a call to `rupgrade_tool -s` to initiate the upgrade procedure. The call can be made either locally from the ShMM serial console or remotely over the network via `telnet`, `rsh`, `ssh`, or any equivalent utility.

2. The user waits for `rupgrade_tool -s` to reboot the ShMM. If the user is connected to the serial console locally, the status of the reboot is obvious from the messages printed by the U-Boot firmware and Linux to the serial console. If the connection to the ShMM is remote, the status of the reboot is less obvious. For instance, a Telnet connection will time out on the reboot of the shelf management card. The user can either assume that the upgrade procedure has been carried out successfully or wait for a certain amount of time required for the upgrade session to complete and then make a call to `rupgrade_tool -w` (again, remotely, over any of the remote shell tools mentioned above) to find out the status of the upgrade session. The amount of time to wait depends on the size of the upgrade images and the copy protocol used to pull the images to the shelf management card as well as actions performed by the image validation script.

3. On the shelf management card, the startup script `/etc/rc` unconditionally makes a call to `rupgrade_tool -c`. If the call returns a value of `1`, indicating that there is no upgrade in progress or an error code value indicating that the upgrade session has failed, the startup scripts proceed with the normal mode boot sequence. However, if a value of `0` is returned, indicating that there is an upgrade session in progress, the startup scripts proceed with validation of the sanity of the newly installed software, calling `rupgrade_tool -S` in the middle of the operation to strobe the upgrade WDT in case the validation takes longer than the upgrade WDT timeout period, and finally start the Shelf Manager to perform final validation. The watchdog timer interval is set to 12.8 seconds, so the processing times in the `/etc/rc` script between the call to `rupgrade_tool -c` and strobing the WDT and between strobing the WDT and starting the Shelf Manager must not exceed 12.8 seconds each.

4. During initialization, the Shelf Manager strobes the upgrade WDT once again, before trying to establish a network connection with the peer Shelf Manager. Establishing a network connection may take up to six seconds. After that, and after successfully finishing the initialization (which indicates validity of the new configuration), the Shelf Manager makes a call to `rupgrade_tool -f`, which completes the upgrade procedure.

5. The user optionally calls `rupgrade_tool -w` to find out the status of the upgrade session. As explained above, this option may be especially useful for a remote upgrade session where the progress of the upgrade cannot be observed directly from the messages printed to the serial console, as is the case for a local upgrade.

After the completion of the reliable upgrade, the user can revert to the original images if the new images are not acceptable for any reason. To do this, the user calls `rupgrade_tool -u`.

If necessary, the above sequence can be easily automated by developing a simple script designed to run on a remote network host. Alternatively, an operator can carry out the reliable upgrade manually, either locally from the serial console or remotely over the network.

Reliable Upgrade Examples

Example 1:

This example shows a reliable upgrade of all three components (U-Boot, kernel, and RFS image), copying `/etc` and `/var/nvdata` non-volatile directories to the provisional Flash. All images are taken from the local `/tmp` (which implies that they have already been copied there in some unspecified way). The U-boot image is taken from `/tmp/u-boot.bin`, the kernel image is taken from `/tmp/sentry.kernel`, and the RFS image is taken from `/tmp/sentry.rfs`. The upgrade procedure is started from the serial console. Comments are interspersed in the console log to provide additional background on the steps of the upgrade procedure.

First, `rupgrade_tool` is started from the command prompt. The parameters show that all three Flash images are to be updated, with the Shelf Manager non-volatile data and configuration file preserved as well.

```
# rupgrade_tool -s --k=sentry.kernel --r=sentry.rfs
--u=u-boot.bin --hook=etc_copy -v
rupgrade_tool: PLB is 5
rupgrade_tool: EEPROM page saved
rupgrade_tool: persistent flash is 0
rupgrade_tool: provisional flash is 1
rupgrade_tool: copying image(s)
```

The upgrade utility attempts to invoke a validation script to check the images in `/tmp` currently supplied. If any of the specified file designators is not found in `/tmp`, the utility stops and a message like the following is produced.

```
rupgrade_tool: cannot open /tmp/u-boot.bin for reading.
rupgrade_tool: failed to copy images to flash
```

The utility proceeds to copy the images to the specified destinations in provisional Flash.

```
rupgrade_tool: invoking scripts (step4v*) [--u=u-boot.bin --k=
sentry.kernel --r=sentry.rfs --hook=etc_copy]
rupgrade_tool: copying u-boot.bin from /tmp to /dev/mtdchar8 using
'cp' protocol
rupgrade_tool: copying sentry.kernel from /tmp to /dev/mtdchar7
using 'cp' protocol
rupgrade_tool: copying sentry.rfs from /tmp to /dev/mtdchar9 using
'cp' protocol
rupgrade_tool: invoking scripts (step4h*) [etc_copy]
```

At this point, the `step4hshm` hook script is invoked; it stops the Shelf Manager and copies non-volatile information to the provisional Flash.

```
/etc/upgrade/step4hshm: Stopping Shelf Manager...
/etc/upgrade/step4hshm: Erasing /var and /etc, copying
/var/nvdata...
/etc/upgrade/step4hshm: Operation: copy /etc and /var/nvdata.
/etc/upgrade/step4hshm: Copying completed.
rupgrade_tool: image(s) copy OK
rupgrade_tool: watchdog started
rupgrade_tool: selected provisional flash
rupgrade_tool: reboot
Restarting system.
```

Here, the reliable upgrade procedure resets the ShMM. This causes U-boot to start from the provisional Flash.

```
* Resetting Integrated Peripherals

U-Boot 1.1.2 (May 12 2005 - 21:27:13)

CPU: Au1550 324 MHz, id: 0x02, rev: 0x00
Board: ShMM-500
S/N: 8000044
DRAM:  128 MB
Flash: 64 MB
In:    serial
Out:   serial
Err:   serial
Net:   Au1X00 ETHERNET
Hit any key to stop autoboot:  0
## Booting image at bfb00000 ...
   Image Name:   MIPS Linux-2.4.26
```

```
   Created:      2005-06-24  13:29:50 UTC
   Image Type:   MIPS Linux Kernel Image (gzip compressed)
   Data Size:    844843 Bytes = 825 kB
   Load Address: 80100000
   Entry Point:  802bc040
   Verifying Checksum ... OK
   Uncompressing Kernel Image ... OK
## Loading Ramdisk Image at bfc40000 ...
   Image Name:   sentry RFS Ramdisk Image
   Created:      2005-04-22   9:10:41 UTC
   Image Type:   MIPS Linux RAMDisk Image (gzip compressed)
   Data Size:    2465924 Bytes =  2.4 MB
   Load Address: 00000000
   Entry Point:  00000000
   Verifying Checksum ... OK

Starting kernel ...

init started:  BusyBox v0.60.5 (2005.06.15-14:45+0000) multi-call
binary
/etc/rc: Mounted /proc
/etc/rc: Mounting filesystems...
/etc/rc: Mounted /dev/pts
/etc/rc: Mounted /dev/mtdblock0 to /var
/etc/rc: Mounted /dev/mtdblock10 to /var/upgrade
```

At this point, the `rc` script invokes `rupgrade_tool -c` to check if a reliable upgrade is in progress. The tool returns `0`, confirming that an upgrade is in progress. Given that result, the `rc` script continues with the startup process.

```
/etc/rc: Checking the reliable upgrade watchdog timer
/etc/rc: Mounted ram disk to /var/log
/etc/rc: Started syslogd and klogd
/etc/rc: Mounted ram disk to /var/tmp
/etc/rc: Setting hostname shmm+193
```

Since a reliable upgrade is in progress, the watchdog timer is strobed once more in the `rc` script.

```
/etc/rc: Strobing the reliable upgrade watchdog timer
/etc/rc: Mounted /dev/mtdblock1 to /etc
/etc/rc: Calling /etc/rc.carrier3
Board Hardware Address: 0xFE
/etc/netconfig: /etc/hosts has valid  192.168.1.193 entry
/etc/netconfig: Updating /etc/profile.sentry with IP settings
```

```
/etc/netconfig: ifconfig eth0 192.168.1.193
/etc/netconfig: ifconfig eth1 192.168.0.193
/etc/netconfig: route add default gw 192.168.1.253
/etc/netconfig: Starting inetd...
/etc/rc.carrier3: Starting up IPMBs ...
/etc/rc.carrier3: Updating /etc/profile.sentry with specific
settings
/etc/rc.carrier3: Starting snmpd...
/etc/rc.carrier3: Starting httpd...
/etc/rc.carrier3: Starting Shelf Manager ...
<I> 02:48:08.463   [171] Pigeon Point Shelf Manager ver. 2.0.0.
Built on Jun 27 2005 14:48:57
<*> 02:48:08.469   [171] Limits: code=(400000:506f0), end_data=
10062000, start_stack=7fff7e30, esp=7fff78a0, eip=6F or 2A (Sun
legacy)b0d2e4
<*> 02:48:08.469   [171] Stack limits: curr=1ff000, max=7fffffff
<*> 02:48:08.470   [171] Data limits: curr=7fffffff, max=7fffffff
<*> 02:48:08.900   [171] *** Lock log print buffer at 1003b7f0 ***
<*> 02:48:08.900   [171] *** Pthread lock log print buffer at
1003f820 ***
```

The Shelf Manager starts and finalizes the reliable upgrade by executing `rupgrade_tool -f`.

```
eth0: link up
eth1: link up
eth1: going to full duplex

shmm+193 login:root

BusyBox v0.60.5 (2005.05.12-22:46+0000) Built-in shell (msh)
```

The user then checks the status of the reliable upgrade by issuing the `rupgrade_tool -w` command.

```
# rupgrade_tool -w
Recent upgrade status:
1:PLB is 5
1:EEPROM page saved
2:persistent flash is 1
3:provisional flash is 0
4:copying image(s)
4:invoking scripts (step4v*) [--u=u-boot.bin --k=sentry.kernel --
r=sentry.rfs  --hook=etc_copy]
4:copying u-boot.bin from /tmp to /dev/mtdchar8 using 'cp'
protocol
```

```
4:copying sentry.kernel from /tmp to /dev/mtdchar7 using 'cp'
protocol
4:copying sentry.rfs from /tmp to /dev/mtdchar9 using 'cp'
protocol
4:invoking scripts (step4h*) [etc_copy]
4:image(s) copy OK
5:watchdog started
6:selected provisional flash
7:reboot
9:WDT not fired, upgrade in progress.
11:provisional flash 0, updating EEPROM
12:EEPROM updated
13:upgrade WDT disabled
13:invoking scripts (step13h*) []
14:upgrade completed successfully
#
```

Example 2:

This example shows a reliable upgrade of the RFS image only, copying `/etc` and `/var/nvdata` non-volatile directories to provisional Flash. The RFS image is taken from an FTP server at the IP address `192.168.1.253`. The path to the RFS image on the FTP server is `/tftpboot/ru-mips/sentry.mips.rfs`. The upgrade procedure is started from a Telnet session.

Note – Since only the RFS image is explicitly updated, the U-Boot and kernel images are automatically copied from the persistent Flash partition to the provisional Flash partition.

The local system must have access to the FTP server over the network (that is, its network adapter must be up and configured and a route must exist from the ShMM to the FTP server). In the example, the ShMM is configured with the network address `192.168.1.174` (which is in the same network with the FTP server):

```
# telnet 192.168.1.174
Trying 192.168.1.174...
Connected to 192.168.1.174.
Escape character is '^]'.

BusyBox on shmm+174 login: root



BusyBox v0.60.5 (2005.05.07-17:27+0000) Built-in shell (msh)
```

The parameters to `rupgrade_tool -s` indicate that only the RFS is being upgraded and that the copy protocol is FTP, accessing a specified IP address and file, with user `admin` and no password supplied.

```
# rupgrade_tool -s --r=sentry.mips.rfs
--proto=ftp:192.168.1.253:/tftpboot/ru-mips:admin --hook=etc_copy
-v
rupgrade_tool: PLB is 5
rupgrade_tool: EEPROM page saved
rupgrade_tool: persistent flash is 1
rupgrade_tool: provisional flash is 0
rupgrade_tool: copying image(s)
rupgrade_tool: copying sentry.rfs from
192.168.1.253:/tftpboot/ru-mips:admin to /tmp using 'ftp' protocol
220 hydra FTP server (Version wu-2.4.2-academ[BETA-17](1) Tue Jun
9 10:43:14 EDT 1998) ready.
USER admin
```

The user is asked for a password to the FTP site; that password is entered manually.

```
331 Password required for admin.
Password:xxxx
PASS *****
230 User admin logged in.
TYPE I
200 Type set to I.
PASV
227 Entering Passive Mode (192,168,1,253,9,20)
RETR /tftpboot/ru-mips/sentry.mips.rfs
150 Opening BINARY mode data connection for
/tftpboot/ru-mips/sentry.mips.rfs (2465988 bytes).
226 Transfer complete.
QUIT
221 Goodbye.
```

In the next step, a special script `step4vshm` is invoked, that copies the U-Boot and kernel images from the persistent Flash to the provisional Flash, After that, the upgrade utility proceeds to copy the RFS image to its designated position in provisional Flash.

```
rupgrade_tool: invoking scripts (step4v*) [--r=sentry.mips.rfs --
proto=ftp:192.168.1.253:/tftpboot/ru-mips:admin --hook=etc_copy]
/etc/upgrade/step4vshm: Erasing /dev/mtdchar7...Done
etc/upgrade/step4vshm: Copying Kernel from /dev/mtdchar2 to
/dev/mtdchar7...Done
```

```
/etc/upgrade/step4vshm: Erasing /dev/mtdchar8...Done
/etc/upgrade/step4vshm: Copying U-Boot from /dev/mtdchar3 to
/dev/mtdchar8...Done
rupgrade_tool: copying sentry.mips.rfs from /tmp to /dev/mtdchar9
using 'cp' protocol
```

The `step4hshm` hook script is invoked, which stops the Shelf Manager and preserves the non-volatile data. The utility then starts the upgrade WDT and reboots.

```
rupgrade_tool: invoking scripts (step4h*) [etc_copy]
/etc/upgrade/step4hshm: Stopping Shelf Manager...
/etc/upgrade/step4hshm: Erasing /var and /etc, copying
/var/nvdata..
/etc/upgrade/step4hshm: Operation: copy /etc and /var/nvdata.
/etc/upgrade/step4hshm: Copying completed.
rupgrade_tool: image(s) copy OK
rupgrade_tool: watchdog started
rupgrade_tool: selected provisional flash
rupgrade_tool: reboot
Restarting system.
Connection closed by foreign host.
```

At this point, the Telnet session is closed after a certain inactivity period; after several seconds, it is possible to reconnect to the target again and check the status of the reliable upgrade, by invoking `rupgrade_tool -w`.

```
# telnet 192.168.1.174
Trying 192.168.1.174...
Connected to 192.168.1.174.
Escape character is '^]'.

BusyBox on shmm+174 login: root

BusyBox v0.60.5 (2005.05.07-17:27+0000) Built-in shell (msh)
#
# rupgrade_tool -w
Recent upgrade status:
1:PLB is 5
1:EEPROM page saved
2:persistent flash is 1
3:provisional flash is 0
4:copying image(s)
4:copying sentry.mips.rfs from 192.168.1.253:/tftpboot/ru-
mips:admin to /tmp using 'ftp' protocol
4:invoking scripts (step4v*) [--r=sentry. rfs --hook=etc_copy]
```

```
4:copying sentry.mips.rfs from /tmp to /dev/mtdchar9 using 'cp'
protocol
4:invoking scripts (step4h*) [etc_copy]
4:image(s) copy OK
5:watchdog started
6:selected provisional flash
7:reboot
9:WDT not fired, upgrade in progress.
11:provisional flash 0, updating EEPROM
12:EEPROM updated
13:upgrade WDT disabled
13:invoking scripts (step13h*) []
14:upgrade completed successfully
#
```

Example 3:

This example shows an unsuccessful reliable upgrade. Power is turned off after the boot from the provisional Flash, but before the reliable upgrade is finalized. After turning the power back on, the rollback to the persistent Flash occurs. This reliable upgrade is initiated from the serial console. All three images are assumed to be already in `/tmp`.

```
# rupgrade_tool -s --k=sentry.kernel --r=sentry.rfs --u=u-boot.bin
--hook=etc_copy -v
rupgrade_tool: PLB is 5
rupgrade_tool: EEPROM page saved
rupgrade_tool: persistent flash is 0
rupgrade_tool: provisional flash is 1
rupgrade_tool: copying image(s)
rupgrade_tool: invoking scripts (step4v*) [--u=u-boot.bin --k=
sentry.kernel --r=sentry.rfs --hook=etc_copy]
rupgrade_tool: copying u-boot.bin from /tmp to /dev/mtdchar8 using
'cp' protocol
rupgrade_tool: copying sentry.kernel from /tmp to /dev/mtdchar7
using 'cp' protocol
rupgrade_tool: copying sentry.rfs from /tmp to /dev/mtdchar9 using
'cp' protocol
rupgrade_tool: invoking scripts (step4h*) [etc_copy]
Stopping Shelf Manager...

Pigeon Point Shelf Manager Command Line Interpreter

Terminating the
```

```
Erasing /var and /etc, copying /var/nvdata...
Operation: copy /etc and /var/nvdata.
Copying completed.
rupgrade_tool: image(s) copy OK
rupgrade_tool: watchdog started
rupgrade_tool: selected provisional flash
rupgrade_tool: reboot
Restarting system.
```

The reliable upgrade procedure resets the ShMM here and starts U-boot from the provisional Flash.

```
** Resetting Integrated Peripherals


U-Boot 1.1.2 (Nov 11 2005 - 15:16:25)

CPU: Au1550 324 MHz, id: 0x02, rev: 0x00
Board: ShMM-500
S/N: 8000044
DRAM:  128 MB
Flash: 64 MB
In:    serial
Out:   serial
Err:   serial
Net:   Au1X00 ETHERNET
Hit any key to stop autoboot:  0
## Booting image at bfb00000 ...
   Image Name:   MIPS Linux-2.4.26
   Created:      2005-04-11  10:35:08 UTC
   Image Type:   MIPS Linux Kernel Image (gzip compressed)
   Data Size:    843129 Bytes = 823.4 kB
   Load Address: 80100000
   Entry Point:  802bc040
   Verifying Checksum ... OK
   Uncompressing Kernel Image ... OK
## Loading Ramdisk Image at bfc40000 ...
   Image Name:   sentry RFS Ramdisk Image
   Created:      2005-04-22   9:10:41 UTC
   Image Type:   MIPS Linux RAMDisk Image (gzip compressed)
   Data Size:    2400736 Bytes =  2.3 MB
   Load Address: 00000000
   Entry Point:  00000000
   Verifying Checksum ... OK
```

Power is turned off here. After some time, power is turned back on. Assignment of provisional Flash has been lost because of the power loss, so the system reverts back to the persistent Flash.

```
U-Boot 1.1.2 (Nov 11 2005 - 15:16:25)

CPU: Au1550 324 MHz, id: 0x02, rev: 0x00
Board: ShMM-500
S/N: 8000048
DRAM:  128 MB
Flash: 64 MB
In:    serial
Out:   serial
Err:   serial
Net:   Au1X00 ETHERNET
Hit any key to stop autoboot:  0
## Booting image at bfb00000 ...
   Image Name:   MIPS Linux-2.4.26
   Created:      2005-04-11  10:35:08 UTC
   Image Type:   MIPS Linux Kernel Image (gzip compressed)
   Data Size:    843129 Bytes = 823.4 kB
   Load Address: 80100000
   Entry Point:  802bc040
   Verifying Checksum ... OK
   Uncompressing Kernel Image ... OK
## Loading Ramdisk Image at bfc40000 ...
   Image Name:   sentry RFS Ramdisk Image
   Created:      2005-04-11  18:27:17 UTC
   Image Type:   MIPS Linux RAMDisk Image (gzip compressed)
   Data Size:    2372311 Bytes =  2.3 MB
   Load Address: 00000000
   Entry Point:  00000000
   Verifying Checksum ... OK

Starting kernel ...

init started:  BusyBox v0.60.5 (2005.02.07-16:45+0000) multi-call
binary
hub.c: new USB device AU1550-1, assigned address 2
usb0: ? speed config #1: Ethernet Gadget
usb1: register usbnet usb-AU1550-1, Linux Device
serial#=8000048: not found
/etc/rc: Mounted /proc
/etc/rc: Mounting filesystems...
/etc/rc: Mounted /dev/pts
```

```
/etc/rc: Mounted /dev/mtdblock0 to /var
/etc/rc: Mounted /dev/mtdblock10 to /var/upgrade
```

The next step in the rc script is to call `rupgrade_tool -c` to check whether a reliable upgrade is in progress. The check determines that an attempted reliable upgrade failed. The message `restoring ADM1060 EEPROM to RAM` refers to the ShMM system supervisory device (an ADM1060), which supervises the ShMM boot process and implements some of the hardware aspects of the reliable upgrade support. This message indicates that key variables affecting the boot process are being restored to their state before the reliable upgrade was attempted.

```
/etc/rc: Checking the reliable upgrade watchdog timer
rupgrade_tool: Watchdog not active.
rupgrade_tool: restoring ADM1060 EEPROM to RAM
rupgrade_tool: upgrade failed
/etc/rc: Rupgrade -c Ret: 255
/etc/rc: Mounted ram disk to /var/log
/etc/rc: Started syslogd and klogd
/etc/rc: Mounted ram disk to /var/tmp
/etc/rc: Setting hostname shmm+173
/etc/rc: Mounted /dev/mtdblock1 to /etc
/etc/rc: Calling /etc/rc.carrier3
Board Hardware Address: 0xFE
/etc/netconfig: /etc/hosts has valid  192.168.1.173 entry
/etc/netconfig: Updating /etc/profile.sentry with IP settings
/etc/netconfig: Starting inetd...
/etc/rc.carrier3: Starting up IPMBs ...
/etc/rc.carrier3: Updating /etc/profile.sentry with specific
settings
/etc/rc.carrier3: RC2 daemons not started by request
```

Connecting to a Node Board Console

The Shelf Manager provides the capability to connect to node boards and open console sessions from the active shelf management card (ShMM). This feature is often referred to as *NetConsole*. You begin by logging in to the active ShMM through the Ethernet port. Use of the ShMM's serial port is not recommended for this function.

Note – The shelf management card must be the **active** shelf management card to use the NetConsole feature. A switch card must also be installed in slot 7 of the server's midplane.

Once a console session with a node board is established, you can run system administration commands, such as `passwd`, read status and error messages, or halt the board in that particular slot.

Note – When a console or serial cable is connected to the node board's serial port, the console output goes to the cabled console rather than the console session on the ShMM even if the ShMM's console session was active when the cable is plugged in.

Establishing Console Sessions Between the Shelf Manager and Node Boards

Once you have configured your system for console use, you can log in to the active ShMM and open a console for a slot. The Sun Netra CT system allows one console session per node board slot. TABLE 3-6 shows the Shelf Manager CLI console-related commands that can be executed from the current login session on the shelf management card.

TABLE 3-6 Shelf Manager CLI Console-Related Commands

Command	Description
`clia console [-f]` *slot-number*	Enter console mode and connect to the node board in *slot-number*, where *slot-number* is the slot number where the node board resides. The `-f` option forces a new console session by terminating an existing console session before starting the new console session.
`~q` or `~.`	Disconnect from the current console session.

▼ To Start a Console Session From the Shelf Manager

1. **Log in to the ShMM.**

   You can log in to the ShMM through a terminal attached to either the serial port connection on the SAP or the Ethernet port connection.

2. **Check to see if the ShMM is the active ShMM.**

   Once you are logged in, use the `clia shmstatus` command to verify that you are logged onto the *active* ShMM before continuing. If you are logged onto the *standby* ShMM, either connect to the other serial port on the SAP or use the `clia switchover` command to change the ShMM to `Active`. (See `shmstatus` and `switchover` in "Shelf Manager CLI Commands" on page 157 for more information.)

3. **Open a console session to a node board.**

```
# clia console slot-number
```

where *slot-number* is 1 -6 and 9 -14. For example, to open a console to the board in slot 4, enter the following:

```
# clia console 4
```

You now have access to the node board in slot 4. Depending on the state of the board in that particular slot, and whether the previous user logged out of the shell, you see one of several prompts:

- `console login%` (Oracle Solaris level)
- `#` (Oracle Solaris level, previous user logged in as superuser, and did not log out before disconnecting from the console)
- `ok` (OpenBoot PROM level)
- `#` (Monta Vista Linux)

▼ To End the Console Session

1. **(Optional) Log out of the OS shell.**

2. **At the prompt, disconnect from the console by entering the escape sequence** `~q` **or** `~.` **(tilde period):**

```
prompt ~q
```

Disconnecting from the console does not automatically log you out from the remote host. Unless you log out from the remote host, the next console user who connects to that board sees the shell prompt of your previous session.

Note – Always logout of the console session when your are done.

Changing Console Escape Character

Normally the `~.` or `~q` escape sequences are used to disconnect a console session. The tilda (~) is the default console escape character. The escape character can be modified by changing the value of the `Escape_Char` variable in the `/etc/netconsole.conf` file. For example, `Escape_Char="#"` changes the escape character to the pound sign (`#`) and the console disconnect escape sequences to `#.` or `#q`. All active console sessions must be restarted to implement the change.

The escape character will default to ~ if the `Escape_Char` variable is missing or undefined.

Displaying Board's User Label During Console Session

Administrators can assign names (or user labels) to the shelf and to slots using the ShMM CLI `setuserlabel` command (see "`setuserlabel`" on page 306 and "`userlabel`" on page 348). User labels are displayed as *shelf-name*:*slot-name*.

During a console session, the user can display the user label for the session by typing the escape character and `?` or `L`. Using the default escape character ~ (tilde), the escape sequence would be `~?` or `~L`.

For example:

```
# clia console 6
console login
. . . . . . . . . . . . . . .
# ~?
Connected to "ATCA02:CP3020-06" console.
#
```

Manual Graceful Shutdown of Node Boards

For Sun Netra CP3010, CP3020, and CP3060 node boards, a manual graceful shutdown procedure is provided to prevent the accidental removal of a node boards or FRUs. Graceful shutdown means to quiesce (shut down) all the applications and OS running on the node board's payload, and the payload itself in a non-abrupt manner.

Note – The Sun Netra CP3220, CP3250, and CP3260 node boards automatically execute a graceful shutdown when the hot-swap latch on the board is opened provided deactivation is enabled.

Before hot-swapping or removing a one of these Sun Netra node boards, the applications and operating system running on the node board should be gracefully shut down. Using the Shelf Manager's NetConsole feature, the system administrator can start a console session on the node board and perform a graceful shutdown according to the site's shutdown procedures.

Once the shutdown is complete, the system administrator enables deactivation of the FRU (or node board) and opens the hot-swap latch on the board. After the node board is replaced or reinstalled, the administrator disables deactivation of the FRU (or node board) to prevent an ungraceful shutdown when the hot-swap latch is open.

The steps involved in this process are detailed in the following procedures.

To Shutdown a Node Board

This procedure requires that the hot-swap latch on the node board be closed.

Note – If the hot-swap latch is opened, the blue hot-swap LED will continue blinking and the board will not become hot-swap ready (as indicated by a steadily lit blue LED). To correct this condition, close the hot-swap latch before starting this procedure.

---

1. **Log in to the active ShMM.**
2. **Start a console session to the node board.**

   See "To Start a Console Session From the Shelf Manager" on page 149.
3. **Shut down the application and operating system on the node board.**

   Follow the graceful shutdown procedures for your site. When the shutdown is complete, close the console session (see To End the Console Session).
4. **Enable the shelf manager controlled deactivation on the node board using the following Shelf Manager CLI command.**

```
# clia shelf deactivation hardware-addr fru-id 0
```

For example:

```
# clia shelf deactivation 0x41 0xfe 0
Pigeon Point Shelf Manager Command Line Interpreter

Updating Shelf FRU Info, address: 0x41, FRU ID # 254
    Cached information updated
    Wrote Information to the Shelf FRU
#
```

5. **Verify that the shelf manager controlled deactivation is enabled using the following Shelf Manager CLI command.**

```
# clia shelf pm
```

Look for the `Shelf Manager Controlled Deactivation: Enabled` message.

For example:

```
# clia shelf pm
Pigeon Point Shelf Manager Command Line Interpreter

PICMG Shelf Activation And Power Management Record (ID=0x12)
        Version = 1
    Allowance for FRU Activation Readiness: 10 seconds
    FRU Activation and Power Description Count: 16
    Hw Address: 41, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Shelf Manager Controlled Deactivation: Enabled
        Delay Before Next Power On: 0.0 seconds
```

6. **Open the hot-swap latch on the node board.**

7. **When the node board blue hot-swap LED is on steady, remove or replace the node board.**

Caution – Always wear a grounded antistatic wrist strap when handling modules.

---

8. **When the node board is replaced or reinstalled, disable the shelf manager controlled deactivation on the node board using the following Shelf Manager CLI command.**

```
# clia shelf deactivation hardware-addr fru-id 1
```

For example:

```
# clia shelf deactivation 0x41 0xfe 1
Pigeon Point Shelf Manager Command Line Interpreter

Updating Shelf FRU Info, address: 0x41, FRU ID # 254
    Cached information updated
    Wrote Information to the Shelf FRU
#
```

APPENDIX **A**

Shelf Manager CLI Commands

This chapter describes each CLI command and provides the syntax and usage of each of the available commands. The CLI supports AdvancedTCA shelf contexts.

As a convenience, key types of shelf components can be referenced in the following way, as an alternative to a reference notation based on the IPMB address and numerical FRU identifier:

- `board` *n* | `b` *n*
- `fan_tray` *n* | `ft` *n*
- `shm 1` | `2`

Note – The notation `shm 1` and `shm 2` can be used to access the redundant Shelf Managers that are described in the address table in the Shelf FRU. In this manual, `shm 1` relates to the Shelf Manager with the numerically smaller hardware address, and `shm 2` relates to the Shelf Manager with the numerically greater hardware address.

In redundant configurations, not all commands listed below are supported by the backup Shelf Manager. The backup Shelf Manager recognizes only the following commands:

- `debuglevel`
- `localaddress`
- `shmstatus`
- `switchover`

Most informational commands support brief and verbose modes of execution, differing in the amount of information provided. Brief mode is the default (standard); verbose mode is selected by using the option `-v` in the command line, directly after the command and before the positional arguments.

In the command syntax below, optional elements are enclosed in square brackets ([,]), and variable elements in the command line (for example, IPMB address and FRU device ID) are printed in *italics*. A vertical bar (|) separates parameter alternatives.

SNMP is supported for reading and setting TELCO alarms (minor, major and critical).

activate

Syntax:

`activate` *IPMB-address fru-id*
`activate board` *n*
`activate shm` *n*

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. FRU device ID 0 designates the IPM controller proper in PICMG 3.0 contexts.

In the PICMG 3.0 context, this command is primarily useful for those FRUs that were not listed in the power management table in the Shelf FRU Information, or for which the `Shelf Manager Controlled Activation` attribute is set to `0 (DISABLED)`. These FRUs are not automatically activated by the Shelf Manager and stay in the state M2 (Activation Request). The Shelf Manager automatically activates other FRUs once they reach state M2. Attempting to activate a FRU that is not in state M2 does nothing.

Example:

Activate the IPM controller proper at address 9C.

```
# clia activate 9c 0
Pigeon Point Shelf Manager Command Line Interpreter
Command issued via IPMB, status = 0 (0x0)
Command executed successfully
#
```

alarm

Syntax:

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

Purpose:

This command provides access to the Telco alarm outputs. Parameters `minor`, `major`, and `critical` allow the user to set the corresponding alarm output. These actions are cumulative; that is, after the commands `clia alarm minor` and `clia alarm major`, both minor and major alarms will be set. The action `clear` clears the minor and major alarm outputs; critical alarm output cannot be cleared. The action `info` displays information about the last alarm that occurred in the shelf.

Command invocation without parameters will return the status of Telco alarm outputs.

Examples:

```
# clia alarm
Pigeon Point Shelf Manager Command Line Interpreter

    alarm mask: 0x00
# clia alarm major
Pigeon Point Shelf Manager Command Line Interpreter

Returned completion code: 0
# clia alarm
Pigeon Point Shelf Manager Command Line Interpreter

    alarm mask: 0x02
      Major Alarm
# clia alarm clear
Pigeon Point Shelf Manager Command Line Interpreter

Returned completion code: 0
# clia alarm
Pigeon Point Shelf Manager Command Line Interpreter

alarm mask: 0x00
```

```
# clia alarm info
Pigeon Point Shelf Manager Command Line Interpreter

Last saved alarm information:
 Alarm mask: 0x02
 Alarm date/time: Wed May 10 10:54:04 2007
 Alarm source: Remote request
 Alarm reason: On-demand setting alarms mask: 0x02
#
```

airfilterreplaced

Syntax:

`airfilterreplaced` [*dd.mm.yyyy*]

Purpose:

A fan tray air filter change date can be maintained in a Pigeon Point defined multirecord in the Shelf FRU Information. This multirecord contains the following information related to air filter changes:

- `Air Filter Replaced` – Indicates the date when the Air Filter was last replaced. The format is *dd.mm.yyyy*
- `Air Filter To Be Replaced` – Indicates the date when the Air Filtrer must be replaced again (the filter expiration date). The format is *dd.mm.yyyy*.

After an operator has replaced the air filter, he or she should use the Shelf Manager CLI tool to change the dates above in the Shelf FRU Information.

The date *dd.mm.yyyy* specified as the command parameter indicates the filter expiration date (that is, when the Air Filter shall be replaced again). If the date is omitted, the default expiration time is 6 months from the current date.

After executing the command, the Shelf Manager updates the Shelf FRU Information as follows: the field `Air Filter Replaced` contains the current calendar date and the field `Air Filter To Be Replaced` contains the expiration date - either the date supplied with the command or the default date - 6 months in the future.

Note – This command requires special carrier-specific support and is not implemented for all ShMM carriers. If the command is not implemented for the current carrier, an error message is shown when this command is used.

Example:

Set the date for the next air filter replcement in the Shelf FRU information.

```
# clia airfilterreplaced 25.10.2007
Pigeon Point Shelf Manager Command Line Interpreter

Command executed successfully
#
```

amcportstate

Syntax:

amcportstate [-v] *IPMB-address* [*fru-id* | amc *amc-number*]
amcportstate [-v] board *n* [*fru-id* | amc *amc-number*]
amcportstate [-v] shm *n* [*fru-id* | amc *amc-number*]

Purpose:

This command shows AMC port state information for a specific AMC. If a *fru-id* or *amc-number* is omitted, the AMC port state information is reported for all active AMCs for the designated IPM controller.

Examples:

```
# clia amcportstate 98

Pigeon Point Shelf Manager Command Line Interpreter

98: FRU # 1 (AMC # 5)
    Channel 0:
        Link 1 configuration:
            lane mask 03, type 07, type extension 02, grouping ID
00, status 0 (Disabled)
        Link 2 configuration:
            lane mask 01, type 07, type extension 02, grouping ID
00, status 1 (Enabled)
        Link 3 configuration:
            lane mask 02, type 07, type extension 02, grouping ID
00, status 0 (Disabled)
```

```
98: FRU # 2 (AMC # 6)
    Channel 0:
        Link 1 configuration:
            lane mask 03, type 07, type extension 02, grouping ID
00, status 0 (Disabled)
        Link 2 configuration:
            lane mask 01, type 07, type extension 02, grouping ID
00, status 1 (Enabled)
        Link 3 configuration:
            lane mask 02, type 07, type extension 02, grouping ID
00, status 0 (Disabled)

# clia amcportstate 9c 2

Pigeon Point Shelf Manager Command Line Interpreter

9C: FRU # 2 (AMC # 6)
    Channel 0:
        Link 1 configuration:
            lane mask 0f, type 05, type extension 01, grouping ID
00, status
1 (Enabled)
    Channel 1:
        Link 1 configuration:
            lane mask 0f, type 05, type extension 01, grouping ID
00, status
1 (Enabled)
    Channel 2:
        Link 1 configuration:
            lane mask 01, type f0, type extension 00, grouping ID
00, status
1 (Enabled)

# clia amcportstate 88 amc 6

Pigeon Point Shelf Manager Command Line Interpreter

88: FRU # 2 (AMC # 6)
    Channel 0:
        Link 1 configuration:
            lane mask 01, type 07, type extension 01, grouping ID
00, status 1 (Enabled)
```

```
    Channel 1:
        Link 1 configuration:
            lane mask 01, type 07, type extension 01, grouping ID
00, status 0 (Disabled)
```

board

Syntax:

`board` [`-v`] [*physical-slot-address*]

Purpose:

This command and the `boardreset` command are different from the rest of the command set in that they work with ATCA boards and take as arguments physical slot numbers, rather than IPM controller addresses and FRU device IDs.

The command `board` shows information about each IPM controller in the range of IPMB addresses allocated to ATCA slots, and about each additional FRU controlled by these controllers. The list of items to be shown is given in sections "`fru`" on page 180 and "`ipmc`" on page 245. The range of IPMB addresses is 82h-A0h for PICMG 3.0 systems, where boards have IPM controllers on them.

The physical address should be specified as a decimal number. For PICMG 3.0 systems, the correspondence between physical addresses and IPMB addresses is specified in the Shelf FRU Information. If the Shelf FRU information does not contain an address table, the following mapping table (mapping of logical slot numbers) is used.

Slot Number	IPMB Address
1	9A
2	96
3	92
4	8E
5	8A
6	86
7	82
8	84
9	88

Slot Number	IPMB Address
10	8C
11	90
12	94
13	98
14	9C

Examples:

Get standard information about all boards in the system (where only the boards in physical slots 4, 5, and 8 are present).

```
# clia board
Pigeon Point Shelf Manager Command Line Interpreter

Physical Slot # 4
8e: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x01
    PICMG Version 2.1
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)

8e: 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: "NetraCP-3060"

8e: FRU # 1
    Entity: (0xc1, 0x61)
   Hot Swap State: M4 (Active), Previous: M7 (Communication Lost),
Last State Change Cause: Normal State Change (0x0)
    Device ID String: "SB AMC-HD-A-80X"

Physical Slot # 5
8a: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x00
    PICMG Version 2.1
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)

8a: 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: "NetraCP-3020"

Physical Slot # 8
84: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x01
    PICMG Version 2.1
    Hot Swap State: M4 (Active), Previous: M3 (Activation In 
Process), Last State Change Cause: Normal State Change (0x0)

84: 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: "CP3140H-BEG"

84: FRU # 1
    Entity: (0xc0, 0x60)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In 
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: "XCP3040H-RTC"
#
```

Get verbose information about a board in physical slot 4.

```
# clia board -v 4
Pigeon Point Shelf Manager Command Line Interpreter

Physical Slot # 4
8e: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x01
    PICMG Version 2.1
    Hot Swap State: M4 (Active), Previous: M3 (Activation In 
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID: 0x00, Revision: 0, Firmware: 0.210, IPMI ver 1.5
    Manufacturer ID: 00006F or 2A (Sun legacy), Product ID: 0bf4, 
Auxiliary Rev: 00000000
    Device ID String: "NetraCP-3060"
    Global Initialization: 0xc, Power State Notification: 0xc, 
Device Capabilities: 0x29
    Controller provides Device SDRs
    Supported features: 0x29
        "Sensor Device" "FRU Inventory Device" "IPMB Event 
Generator"
    8e: Base Interface (0x00), Channel: 1
        Link: Enabled Ports: 1
        Peer Addr: 0x82, Link Type: PICMG 3.0 Base Interface 
10/100/1000 BASE-T, Ext: 0 (10/100/1000 BASE-T)
    8e: Base Interface (0x00), Channel: 2
```

```
        Link: Enabled Ports: 1
        Peer Addr: 0x84, Link Type: PICMG 3.0 Base Interface
10/100/1000 BASE-T, Ext: 0 (10/100/1000 BASE-T)
    8e: Fabric Interface (0x01), Channel: 1
        Link: Enabled Ports: 1
        Peer Addr: 0x82, Link Type: PICMG 3.1 Ethernet Fabric
Interface, Ext: 0
    8e: Fabric Interface (0x01), Channel: 2
        Link: Enabled Ports: 1
        Peer Addr: 0x84, Link Type: PICMG 3.1 Ethernet Fabric
Interface, Ext: 0

8e: 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: "NetraCP-3060"
    Site Type: 0x00, Site Number: 04
    Current Power Level: 0x01, Maximum Power Level: 0x01, Current
Power Allocation: 174.0 Watts

8e: FRU # 1
    Entity: (0xc1, 0x61)
    Hot Swap State: M4 (Active), Previous: M7 (Communication Lost),
Last State Change Cause: Communication Lost due to local failure
(0x5)
    Device Type: "FRU Inventory Device behind management
controller" (0x10), Modifier 0x0
    Device ID String: "SB AMC-HD-A-80X"
    Current Power Level: 0x01, Maximum Power Level: 0x01, Current
Power Allocation: 21.1 Watts
#
```

boardreset

Syntax:

`boardreset` *physical-slot-address*

Purpose:

This command resets the board in the specified physical slot, sending it the IPMI command `FRU Control (Cold Reset)`.

The physical address should be specified as a decimal number. For PICMG 3.0 systems, correspondence between physical addresses and IPMB addresses is specified in the Shelf FRU Information. If the Shelf FRU information does not contain an address table, the following mapping table (mapping of logical slot numbers) is used. FRU device ID is 0.

Slot Number	IPMB Address
1	9A
2	96
3	92
4	8E
5	8A
6	86
7	82
8	84
9	88
10	8C
11	90
12	94
13	98
14	9C

Example:

Reset the board in physical slot 14 (IPMB address 9C, FRU 0).

```
# clia boardreset 14
Pigeon Point Shelf Manager Command Line Interpreter

Board 14 reset, status returned 0
#
```

busres

Syntax:

`busres` *subcommand*

The following subcommands are supported:

- `info [`*resource*`]`
- `release` *resource*
- `force` *resource*
- `lock` *resource*
- `unlock` *resource*
- `query [-v]` *resource* `[`*target* `[noupdate]]`
- `setowner` *resource target*
- `sendbusfree` *resource target*

Purpose:

This command shows information about the current state of the Bused E-Keying-managed resources and allows changing that state.

All subcommands accept a resource ID as one of the parameters. The resource ID is either a 0-based resource number or a short resource name. The following resource names and numbers are defined

Number	Short Name	Description
0	mtb1	Metallic Test Bus pair 1
1	mtb2	Metallic Test Bus pair 2

Number	Short Name	Description
2	clk1	Synch Clock group 1
3	clk2	Synch Clock group 2
4	clk3	Synch Clock group 3

The following subsections describe the syntax of the `busres` command for several key uses.

Display the State of Bused E-Keying-Managed Resources

Syntax:

`busres info [`*resource*`]`

Purpose:

This command displays information about the current state of the specified resource or all resources if the resource ID is not specified.

The parameter *resource* is the resource ID. The list of supported resource IDs is provided in "busres" on page 168.

Example:

Get information about the state of Metallic Test Bus pair 2

```
# clia busres info mtb2
Pigeon Point Shelf Manager Command Line Interpreter

Metalic Test Bus pair 2 (ID 1): Owned by IPMC 0x82, Locked
#
```

Release a Specified Resource

Syntax:

`busres release | force` *resource*

Purpose:

This command sends the `Bused Resource Control` request to the current owner of the resource, instructing it to release the resource. If the command syntax is `busres release` *resource*, the `Bused Resource Control (Release)` command is sent. If the command syntax is `busres force` *resource*, the `Bused Resource Control (Force)` command is sent. Refer to section 3.7.3.4 of the PICMG 3.0 R1.0 specification for a detailed description of these ATCA commands.

The parameter *resource* is the resource ID. The list of supported resource IDs is provided in "busres" on page 168.

Example:

Force releasing Metallic Test Bus pair 2 by the current owner.

```
# clia busres force mtb2
Pigeon Point Shelf Manager Command Line Interpreter

Force operation succeeded
#
```

Lock/Unlock the Specified Resource

Syntax:

`busres lock | unlock` *resource*

Purpose:

This command locks (`busres lock` *resource*) or unlocks (`busres unlock` *resource*) the specified resource. If the resource is locked, when another IPM controller sends the `Bused Resource Control (Request)` command to the Shelf Manager, the Shelf Manager responds with the Deny status. If the resource is unlocked, when another IPM controller sends the `Bused Resource Control (Request)` command to the Shelf Manager, the Shelf Manager responds with Busy status and sends the `Bused Resource Control (Release)` to the current owner. If the current owner releases the resource, on the next request, this resource will be granted to the requestor.

Only the resources that are owned by an IPM controller can be locked. As soon as the current owner releases the resource, the lock is also removed from this resource.

The parameter *resource* is the resource ID. The list of supported resource IDs is provided in "busres" on page 168.

Example:

Lock Synch Clock group 3.

```
# clia busres lock clk3
Pigeon Point Shelf Manager Command Line Interpreter

Lock operation succeeded
#
```

Send Bused Resource Control (Query) Command

Syntax:

`busres [-v] query` *resource* `[`*target* `[noupdate]]`

Purpose:

This command sends the `Bused Resource Control (Query)` request to the specified IPM controller. If the IPM controller is not specified in the command line, the request is sent to the current owner of the resource. Upon receiving the response, appropriate changes are made in the resource table (for example, if the IPM controller that is believed to be the current owner responds with the No Control status, the table is modified to reflect that fact), unless the `noupdate` flag is provided. If this flag is passed in the command line, no changes to the resource table are made based on the received information.

The parameter *resource* is the resource ID. The list of supported resource IDs is provided in "busres" on page 168.

The parameter *target* specifies the IPMB address of the IPM controller to which the request will be sent.

The flag `noupdate`, if present, indicates that the information received in response to the Query request should not be used to update the resource table.

In the current revision of the Shelf Manager, no additional information is provided if `-v` flag is specified.

Example:

Send query for Metallic Test Bus pair 1 to the IPM controller with address 0x82. Do not update the resource table based on the response.

```
# clia busres query mtb1 0x82 noupdate
Pigeon Point Shelf Manager Command Line Interpreter

No Control: IPMC 0x82 is not the owner of resource 0
#
```

Set Owner For the Resource

Syntax:

`busres setowner` *resource target*

Purpose:

Caution – This command is for experienced users and should be used at one's discretion.

This command directly sets the owner of the specified resource in the resource table. It does not send any Bused Resource Control commands, even if the resource had a different owner before executing the command. This is a low-level command that should be used for testing and recovery purposes only.

The parameter *resource* is the resource ID. The list of supported resource IDs is provided in "busres" on page 168.

The parameter *target* specifies the IPMB address of the IPM controller that is set as the owner of the resource. Use 0 as the IPMB address to specify that the resource is not owned by any IPM controller.

Example:

Set board 1 as the new owner for Metallic Test Bus pair 1.

```
# clia busres setowner mtb1 board 1
Pigeon Point Shelf Manager Command Line Interpreter

New owner is set successfully
#
```

Send `Bused Resource Control (Bus Free)` Command

Syntax:

`busres sendbusfree` *resource target*

Purpose:

Caution – This command is for experienced users and should be used at one's discretion.

This command sends the `Bused Resource Control (Bus Free)` request to the specified IPM controller. No operation is performed on the resource before sending the request even if a different IPM controller owns it. However, the resource table is updated based on the response to this request. That is, if the IPM controller accepts ownership of the resource, it is set as the new owner in that table. This is a low-level command that should be used for testing and recovery purposes only.

The parameter *resource* is the resource ID. The list of supported resource IDs is provided in "`busres`" on page 168.

The parameter *target* specifies the IPMB address of the IPM controller, to which the request is sent. Use 0 as the IPMB address to specify that the resource is not owned by any IPM controller.

Example:

Send Bus Free request for Metallic Test Bus pair 1 to the IPM controller with address 0x82.

```
# clia busres sendbusfree mtb1 0x82
Pigeon Point Shelf Manager Command Line Interpreter

IPMC rejected ownership of the resource
#
```

console

Syntax:

`console` *slot-number* `[-f]`

Purpose:

This command establishes a console terminal session on the node board in the specified physical slot. The shelf manager allows one console session per node board. Valid slot numbers are from 1-6 and 9-14.

The `-f` option forces a new console session by terminating an existing console session before starting the new console session.

Note – The shelf management must be the **active** shelf management card to use the console feature. A switch card must also be installed in slot 7 of the server's midplane.

Once a console session with a node board is established, you can run system administration commands, such as `passwd`, read status and error messages, or halt the board in that particular slot.

Note – When a console or serial cable is connected to the node board's serial port, the console output goes to the cabled console rather than the console session on the ShMM even if the ShMM's console session was active when the cable is connected.

To disconnect from the current console session, enter `~q` or `~.` (tilda dot).

Example:

Start a console session on the node board in physical slot 4.

```
# clia console 4
prompt
```

deactivate

Syntax:

deactivate *IPMB-address fru-id*
deactivate board *n*
deactivate shm *n*

Purpose:

This command sends the IPMI command `Set FRU Activation (Deactivate FRU)` to the specified FRU. The FRU is specified using the IPMB address of the owning IPM controller and the FRU device ID. FRU device ID 0 designates the IPM controller proper in PICMG 3.0 contexts.

Example:

Deactivate the IPM controller proper at address 9C.

```
# clia deactivate 9c 0
Pigeon Point Shelf Manager Command Line Interpreter

Command issued via IPMB, status = 0 (0x0)
    Command executed successfully
#
```

debuglevel

Syntax:

`debuglevel [`*new-value*`]`

Purpose:

This command shows the current debug level for the Shelf Manager, or sets it to a new value if a new value is specified.

The debug level is a hexadecimal number in the range 0x0000 to 0x00FF that is treated as a bit mask. Each bit in the mask, when set, enables debug output of a specific type:

- 0x0001 – Error messages
- 0x0002 – Warning messages
- 0x0004 – Informational messages
- 0x0008 – Verbose informational messages
- 0x0010 – Trace messages
- 0x0020 – Verbose trace messages
- 0x0040 – Messages displayed for important commands sent to the IPM controllers during their initialization
- 0x0080 – Verbose messages about acquiring and releasing internal locks

Starting with Shelf Manager release 2.4.4, separate debug levels can be set for Shelf Manager output to the system log versus output to the console. This makes it possible, for example, to reserve the system console for only serious error messages, while preserving the normal verbosity of the Shelf Manager output to the system log.

This command, when issued without parameters, shows the current debug level values for both system log and console. If both levels have the same value, only a single line of output produced.

This command, when issued with a single parameter *new-value*, sets the specified debug level for output to both the system log and the console.

If this command is invoked with two parameters, the first parameter specifies the debug level for system log output and the second parameter specifies the debug level for console output.

The default debug level for the Shelf Manager is 0x0007 for both the system log and the console, but this value can be overridden in the Shelf Manager configuration file (separately for the system log and the console), or during Shelf Manager startup, using the `-v` option in the command line for both the system log and the console.

This command can also be issued on the backup Shelf Manager.

Examples:

Get current debug levels, and then set both to 0x001F.

```
# clia debuglevel
Pigeon Point Shelf Manager Command Line Interpreter

Debug Mask is 0x0007

# clia debuglevel 1f
Pigeon Point Shelf Manager Command Line Interpreter

Debug Mask is 0x001f

# clia debuglevel
Pigeon Point Shelf Manager Command Line Interpreter
#
Debug Mask is 0x001f
```

Set the system log debug level mask to 0x0007 (informational) and the console maskto 0x0003 (errors and warnings only).

```
# clia debuglevel 7 3
Pigeon Point Shelf Manager Command Line Interpreter

Debug Mask is 0x0007
Console Debug Mask is set to 0x0003

# clia debuglevel
Pigeon Point Shelf Manager Command Line Interpreter

Debug Mask is 0x0007
Console Debug Mask is 0x0003
#
```

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
#
```

fans

Syntax:

fans [-v] [*IPMB-address* [*FRU-device-ID*]]
fans fan_tray *n*

Purpose:

This command shows information about the specified fan FRUs. If FRU device ID is omitted, the command shows information about all fan FRUs controlled by the IPM controller at the specified address. If the IPMB address is also omitted, the command shows information about all fan FRUs known to the Shelf Manager. The following information is shown:

- IPMB address and FRU device ID
- Minimum Speed Level
- Maximum Speed Level
- Maximum Sustained Speed Level
- Current Level (Override and Local Control levels, if both are available)

Example:

Get fan information about all fan FRUs at IPMB address 20.

```
# clia fans 20
Pigeon Point Shelf Manager Command Line Interpreter

20: FRU # 3
 Current Level: 9
 Minimum Speed Level: 0, Maximum Speed Level: 15
20: FRU # 4
 Current Level: 9
 Minimum Speed Level: 0, Maximum Speed Level: 15
20: FRU # 5
 Current Level: 9
 Minimum Speed Level: 0, Maximum Speed Level: 15
#
```

`flashupdate`

Syntax:

`flashupdate` *slot-number* `-s` *server-ip-address* `-f` *fw-image-pathname*

Purpose:

This command is used to update the Sun Netra CP3x60 board system firmware with the firmware image from a location that you specify. This command is only valid for Sun Netra CP3x60 boards installed in the Sun Netra CT900 server. The Sun Netra CP3x60 board system firmware that is updated includes the ILOM, ALOM-CMT/ILOM, Hypervisor, OBP, POST, and VBSC firmware.

Note – You can find the links to the Sun Netra CP3x60 firmware download sites at: `http://www.sun.com/downloads/`

To use this command, you need to know the following:

- IP address of the FTP server from which you want to download the firmware image
- User name and password of the FTP server, to enter at the prompts
- Path at which the image is stored

The *slot-number* contains the slot number of a Sun Netra CP3x60 board, the **-s** *server-ip-addresss* argument specifies the IP address of the server to download the firmware image from and the `-f` *fw-image-pathname* specifies the full pathname where the firmware image is located.

Examples:

Downloading and updating system firmware on a Sun Netra CP3x60 board. Note that this process can take several minutes before completion. Upon successful completion, reset the board using the `boardreset` command.

```
# clia flashupdate 2 -s 123.45.67.89
  -f /sysfw/System_Firmware-6_2_5-Netra_CP3060.bin
Username: username
Password: *******
...............................................................
...............................................................
....................................
Update complete. Reset device to use new software.

# clia boardreset slot-number
```

fru

Syntax:

`fru [-v] [`*addr* `[id=`*fru_id* `|` `type=`*site_type*`]] | [type=`*site_type*
`[/`*site_number*`]]`
`fru board` *n*
`fru shm` *n*
`fru fan_tray` *n*

Purpose:

This command shows information about a specific FRU. If the FRU device ID is omitted, the command shows information about all FRUs controlled by the IPM controller at the specified address. If the IPMB address is also omitted, the command shows information about all FRUs known to the Shelf Manager.

Additionally, the site type can select FRUs. Site type should be specified in command parameters in hexadecimal. Associations between FRUs and their site types are stored in the Shelf FRU information. Site types are defined in the PICMG 3.0 specification as follows:

- 00h – AdvancedTCA Board
- 01h – Power Entry Module
- 02h – Shelf FRU Information
- 03h – Dedicated ShMM
- 04h – Fan Tray
- 05h – Fan Filter Tray
- 06h – Alarm
- 07h – AdvancedTCA Module (Mezzanine)
- 08h – PMC
- 09h – Rear Transition Module
- C0h–CFh – OEM defined
- All other values reserved.

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

Examples:

Get standard information about all FRUs at address 9C.

```
# clia fru 9c 0
Pigeon Point Shelf Manager Command Line Interpreter

9c: FRU # 0
    Entity: (0xd0, 0x0)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: "PPS Sentry 6"
#
```

Get verbose information about all FRUs at address 9C.

```
# clia fru -v 9c 0
Pigeon Point Shelf Manager Command Line Interpreter

9c: FRU # 0
    Entity: (0xd0, 0x0)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: "PPS Sentry 6"
    Site Type: 0x00, Site Number: 14
    Current Power Level: 0x01, Maximum Power Level: 0x01, Current
Power Consumption: 20.0 Watts
#
```

Get verbose information about FRU 1 at address 20.

```
# clia fru -v 20 id=1
Pigeon Point Shelf Manager Command Line Interpreter

20: FRU # 1
    Entity: (0xf2, 0x60)
```

```
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device Type: "FRU Inventory Device behind management
controller" (0x10), Modifier 0x0
    Device ID String: "Shelf EEPROM 1"
   Current Power Level: 0x01, Maximum Power Level: 0x01, Current
Power Consumption: 5.0 Watts
#
```

frucontrol

Syntax:

`frucontrol` *IPMB-address fru-id command options*

`frucontrol board` *n command options*
`frucontrol shm` *n command options*
`frucontrol fan_tray` *n command options*

Purpose:

This command sends the FRU Control command to the specified FRU, performing the specified operation on the FRU payload. The FRU is specified using the IPMB address of the owning IPM controller and the FRU device ID. FRU device ID 0 designates the IPM controller proper in PICMG 3.0 contexts.

The *command* parameter specifies the FRU Control command to be used. It can be specified as one of the following symbolic values:

- `cold_reset` (abbreviated as `cr`) – perform cold reset of the FRU payload
- `warm_reset` (abbreviated as `wr`) – perform warm reset of the FRU payload
- `diagnostic_interrupt` (abbreviated as `di`) – issue the diagnostic interrupt
- `graceful_reboot` (abbreviated as `gr`) – perform graceful reboot of the FRU payload and set the timer value and the time out action parameters.

  The *options* parameter is used to set the timer options for the `graceful_reboot` command. The options are specified in hex and must be in the order shown:

  timer_use timer_action pre-time_out countdown_LSB countdown_MSB

  Refer to the section 8.3.1.1 in the *ATCA Payload Graceful Shutdown Design* document for detailed information.

Note – The `warm_reset`, `diagnostic_interrupt`, and `graceful_reboot` are optional and blade specific. Also `warm_reset` and `diagnostic_interrupt` are not supported on the Sun Netra CP3xxx blades.

Examples:

Issue a cold reset command to the 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
#
```

Issue a cold reset command to the board in slot 13.

```
# clia frucontrol board 13 cr
Pigeon Point Shelf Manager Command Line Interpreter

    FRU Control: Controller 0x98, FRU ID # 0, command 0x00, status
0(0x0)
     Command executed successfully
#
```

Issue a graceful_reboot command to FRU 0 at IPMB address 92.

```
# clia frucontrol 92 0 gr 03 01 0 b0 04
Pigeon Point Shelf Manager Command Line Interpreter

    FRU control: Controller 0x92, FRU ID # 0, command 0x00, status
0(0x0)
     Command executed successfully
#
```

Issue a graceful_reboot command to board in slot 13.

```
# clia frucontrol board 13 gr 03 01 0 b0 04
Pigeon Point Shelf Manager Command Line Interpreter

    FRU control: Controller 0x98, FRU ID # 0, command 0x02, status
0(0x0)
     Command executed successfully
#
```

frudata

Syntax:

frudata [*addr* [*fru_id* [*block_offset*]]]
frudata *addr fru_id byte_offset byte 1* [*byte2* ... [*byte 16*] ...]

addr fru_id can be replaced with the following:

board *n*
shm *n*
fan_tray *n*

Purpose:

This command provides access to the FRU Information in raw form. Depending on the command format, it is used to read or write the FRU Information.

In the read format, the command takes an optional 32-byte block number.

In the write format it requires a byte offset parameter. The user can modify up to 65535 bytes of FRU Information.

frudataw and frudatar are variants of the frudata command. frudataw allows the user to write a files on the ShMM Flash file system into the FRU information storage on a specific FRU in the shelf (see "frudatar" on page 187). frudatar allows the user to transfer the contents of the FRU Information storage for a specific FRU into a file on the ShMM Flash file system (see "frudatar" on page 187).

Examples:

Display standard FRU data for all FRUs.

```
# clia frudata
Pigeon Point Shelf Manager Command Line Interpreter

20: FRU # 0    Failure status: 203 (0xcb)
    Requested data not present
20: FRU # 1 Raw FRU Info Data
    FRU Info size: 529
20: FRU # 2    Failure status: 203 (0xcb)
    Requested data not present
82: FRU # 0 Raw FRU Info Data
    FRU Info size: 160
9c: FRU # 0 Raw FRU Info Data
```

```
    FRU Info size: 160
fc: FRU # 0 Raw FRU Info Data
    Requested data not present
. . .
#
```

This example shows how to display FRU data and ways to write data to a FRU.

```
# clia frudata 20 1 0
Pigeon Point Shelf Manager Command Line Interpreter

20: FRU # 1 Block # 0 Raw FRU Info Data
    FRU Info size: 529
 01 00 01 05  0E 18 00 D3  01 04 01 02  55 AA 83 55
 AA 55 C1 00  00 00 00 00  00 00 00 00  00 00 00 00
#
# clia frudata 20 1 1 0xfc 0xfe
Pigeon Point Shelf Manager Command Line Interpreter

Writing 2 bytes to IPM 0x20, FRU # 1, offset: 1, status = 0(0x0)
#
# clia frudata 20 1 0
Pigeon Point Shelf Manager Command Line Interpreter

20: FRU # 1 Block # 0 Raw FRU Info Data
    FRU Info size: 529
 01 FC FE 05  0E 18 00 D3  01 04 01 02  55 AA 83 55
 AA 55 C1 00  00 00 00 00  00 00 00 00  00 00 00 00
#
# clia frudata 20 1 1 0 1
Pigeon Point Shelf Manager Command Line Interpreter

Writing 2 bytes to IPM 0x20, FRU # 1, offset: 1, status = 0(0x0)
#
# clia frudata 20 1 0
Pigeon Point Shelf Manager Command Line Interpreter

20: FRU # 1 Block # 0 Raw FRU Info Data
    FRU Info size: 529
 01 00 01 05  0E 18 00 D3  01 04 01 02  55 AA 83 55
 AA 55 C1 00  00 00 00 00  00 00 00 00  00 00 00 00
#
```

frudatar

Syntax:

frudatar *addr fru_id file_name*
frudatar *addr fru_id byte_offset byte 1* [*byte2* ... [*byte 16*] ...]

addr fru_id can be replaced with the following:

board *n*
shm *n*
fan_tray *n*

Purpose:

This command reads FRU Information from the specified FRU and stores it in a file on the ShMM Flash file system in a raw format (in other words, uploads FRU Information from the specified FRU to a Flash file). The parameter *file name* specifies the path to the destination file. The number of bytes read from the FRU and written to the destination file is equal to the number of bytes returned in the response to the IPMI command `Get FRU Inventory Area Info` for the specified FRU.

Example:

Read FRU data for a specific FRU and store the data in the named file.

```
# clia frudatar 20 2 /var/tmp/20.2.bin
Pigeon Point Shelf Manager Command Line Interpreter

20: FRU # 2 Raw FRU Info Data
 FRU Info size: 176
01 00 00 01  09 00 00 F5  01 08 19 84  C0 42 C7 53
63 68 72 6F  66 66 D9 53  68 4D 4D 2D  41 43 42 2D
46 43 20 53  68 65 6C 66  20 4D 61 6E  61 67 65 72
86 10 04 41  10 14 01 89  D2 04 65 58  13 51 17 00
00 C0 C1 00  00 00 00 EA  01 0D 19 C7  53 63 68 72
6F 66 66 DD  46 61 6E 20  43 6F 6E 74  72 6F 6C 6C
65 72 20 6F  6E 20 53 68  4D 4D 2D 41  43 42 2D 46
43 89 D2 04  65 58 13 51  17 00 00 C9  52 65 76 2E
20 31 2E 30  30 86 10 04  41 10 14 01  C0 DF 2F 76
61 72 2F 6E  76 64 61 74  61 2F 66 61  6E 2D 66 72
75 2D 69 6E  66 6F 72 6D  61 74 69 6F  6E C1 00 26
#
```

frudataw

Syntax:

frudataw *addr fru_id file_name*
frudataw *addr fru_id byte_offset byte 1* [*byte2* ... [*byte 16*] ...]

addr fru_id can be replaced with the following:

board *n*
shm *n*
fan_tray *n*

Purpose:

This command downloads FRU Information to the specified FRU from a file on the ShMM flash file system. The file contains the raw binary image of the FRU Information. The parameter *file name* specifies the path to the source file.

Example:

Write FRU data from a file to the named FRU.

```
# clia frudataw 20 2 /var/tmp/20.2.bin
Pigeon Point Shelf Manager Command Line Interpreter

 Writing 16 bytes to IPM 0x20, FRU # 2, offset: 0, status = 0(0x0)
 Writing 16 bytes to IPM 0x20, FRU # 2, offset: 16, status = 0(0x0)
 Writing 16 bytes to IPM 0x20, FRU # 2, offset: 32, status = 0(0x0)
 Writing 16 bytes to IPM 0x20, FRU # 2, offset: 48, status = 0(0x0)
 Writing 16 bytes to IPM 0x20, FRU # 2, offset: 64, status = 0(0x0)
 Writing 16 bytes to IPM 0x20, FRU # 2, offset: 80, status = 0(0x0)
 Writing 16 bytes to IPM 0x20, FRU # 2, offset: 96, status = 0(0x0)
 Writing 16 bytes to IPM 0x20, FRU # 2, offset: 112, status = 0(0x0)
 Writing 16 bytes to IPM 0x20, FRU # 2, offset: 128, status = 0(0x0)
 Writing 16 bytes to IPM 0x20, FRU # 2, offset: 144, status = 0(0x0)
 Writing 16 bytes to IPM 0x20, FRU # 2, offset: 160, status = 0(0x0)
 File "/var/tmp/20.2.orig.bin" has been written to the FRU 20#2
#
```

fruinfo

Syntax:

`fruinfo [-v] [-x]` *addr fru_id*

addr fru_id can be replaced by the following:

`board` *n*
`shm` *n*
`fan_tray` *n*

Purpose:

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

Example:

Display FRU information for a particular FRU.

```
# clia fruinfo 20 0
Pigeon Point Shelf Manager Command Line Interpreter

20: FRU # 0, FRU Info
 Failure status: 203 (0xcb
 Requested data not present
#

# clia fruinfo 20 1
Pigeon Point Shelf Manager Command Line Interpreter
20: FRU # 1, FRU Info
Common Header:    Format Version = 1
Chassis Info Area:
    Version   = 1
    Chassis Type             = (1)
    Chassis Part Number      = 0x55 0xAA
    Chassis Serial Number    = 5I:5
Board Info Area:
    Version     = 1
Mfg Date/Time            = Jun 16 15:37:00 2011 (8129737 minutes
since 1996)
Board Manufacturer       = Pigeon Point Systems
Board Serial Number      = PPS0000000
    Board Part Number        = A
```

```
    FRU Programmer File ID   =
Product Info Area:
    Version     = 1
    Language Code            = 25
    Manufacturer Name        = Pigeon Point Systems
    Product Name             = Shelf Manager
    Product Part / Model#    = 000000
    Product Version          = Rev. 1.00
    Product Serial Number    = PPS0000000
    Asset Tag                =
    FRU Programmer File ID   =
Multi Record Area:
    Record Type              = Management Access Record
        Version = 2
    Sub-Record Type: Component Name (0x05)
    PICMG Address Table Record (ID=0x10)
        Version = 1
    PICMG Backplane Point-to-Point Connectivity Record (ID=0x04)
        Version = 0
    PICMG Shelf Power Distribution Record (ID=0x11)
        Version = 0
    PICMG Shelf Activation And Power Management Record (ID=0x12)
        Version = 0
#
# clia fruinfo -v -x 20 1

Pigeon Point Shelf Manager Command Line Interpreter

20: FRU # 1, FRU Info
Common Header:    Format Version = 1
 01 00 01 05  0E 18 00 D3

Chassis Info Area:
    Version   = 1
    Chassis Type             = (1)
    Chassis Part Number      = 0x55 0xAA
    Chassis Serial Number    = 5I:5
    Custom Chassis Info      =
 01 04 01 02  55 AA 83 55  AA 55 C1 00  00 00 00 00
 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 61

Board Info Area:
    Version     = 1
    Language Code            = 25
```

```
   Mfg Date/Time          = Jun 16 15:37:00 2011 (8129737 minutes
since 1996)
    Board Manufacturer       = Pigeon Point Systems
    Board Product Name       = Shelf Manager
    Board Serial Number      = PPS0000000
    Board Part Number        = A
    FRU Programmer File ID   =
    Custom Board Info        =
 01 09 19 C9  0C 7C D4 50  69 67 65 6F  6E 20 50 6F
 69 6E 74 20  53 79 73 74  65 6D 73 D6  53 68 65 6C
 66 20 4D 61  6E 61 67 65  72 20 20 20  20 20 20 20
 20 20 CA 50  50 53 30 30  30 30 30 30  30 C2 41 20
 C0 C1 00 00  00 00 00 A0

Product Info Area:
    Version     = 1
    Language Code            = 25
    Manufacturer Name        = Pigeon Point Systems
    Product Name             = Shelf Manager
    Product Part / Model#    = 000000
    Product Version          = Rev. 1.00
    Product Serial Number    = PPS0000000
    Asset Tag                =
    FRU Programmer File ID   =
    Custom Product Info      =
 01 0A 19 D4  50 69 67 65  6F 6E 20 50  6F 69 6E 74
 20 53 79 73  74 65 6D 73  D6 53 68 65  6C 66 20 4D
 61 6E 61 67  65 72 20 20  20 20 20 20  20 20 20 C6
 30 30 30 30  30 30 C9 52  65 76 2E 20  31 2E 30 30
 CA 50 50 53  30 30 30 30  30 30 30 C0  C0 C1 00 6A

Multi Record Area:
    Record Type              = Management Access Record
        Version = 2
    Sub-Record Type: Component Name (0x05)
    Sub-Record Data:         = ShMM
 03 02 05 A6  50 05 53 68  4D 4D

    PICMG Address Table Record (ID=0x10)
        Version = 1
    Shelf Address            =
    Address Table Entries#   = 16
        Hw Addr: 41, Site # 1, Type: "AdvancedTCA Board" 00
        Hw Addr: 42, Site # 2, Type: "AdvancedTCA Board" 00
        Hw Addr: 43, Site # 3, Type: "AdvancedTCA Board" 00
```

```
        Hw Addr: 44, Site # 4, Type: "AdvancedTCA Board" 00
        Hw Addr: 45, Site # 5, Type: "AdvancedTCA Board" 00
        Hw Addr: 46, Site # 6, Type: "AdvancedTCA Board" 00
        Hw Addr: 47, Site # 7, Type: "AdvancedTCA Board" 00
        Hw Addr: 48, Site # 8, Type: "AdvancedTCA Board" 00
        Hw Addr: 49, Site # 9, Type: "AdvancedTCA Board" 00
        Hw Addr: 4a, Site # 10, Type: "AdvancedTCA Board" 00
        Hw Addr: 4b, Site # 11, Type: "AdvancedTCA Board" 00
        Hw Addr: 4c, Site # 12, Type: "AdvancedTCA Board" 00
        Hw Addr: 4d, Site # 13, Type: "AdvancedTCA Board" 00
        Hw Addr: 4e, Site # 14, Type: "AdvancedTCA Board" 00
        Hw Addr: 4f, Site # 15, Type: "AdvancedTCA Board" 00
        Hw Addr: 50, Site # 16, Type: "AdvancedTCA Board" 00
 C0 02 4B 44  AF 5A 31 00  10 01 00 00  00 00 00 00
 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 10
 41 01 00 42  02 00 43 03  00 44 04 00  45 05 00 46
 06 00 47 07  00 48 08 00  49 09 00 4A  0A 00 4B 0B
 00 4C 0C 00  4D 0D 00 4E  0E 00 4F 0F  00 50 10 00

    PICMG Backplane Point-to-Point Connectivity Record (ID=0x04)
        Version = 0
    P2P Slot Descriptor:
        Channel Type         = 0x0B PICMG®3.0 Base Interface
        LocalSlot/HW Address = 0x41
        Channel Count        = 0x0F
        Channel Descriptor   = LocalChannel 2, RemoteChannel 2,
RemoteSlot 0x42
        Channel Descriptor   = LocalChannel 3, RemoteChannel 1,
RemoteSlot 0x43
        Channel Descriptor   = LocalChannel 4, RemoteChannel 1,
RemoteSlot 0x44
        Channel Descriptor   = LocalChannel 5, RemoteChannel 1,
RemoteSlot 0x45
        Channel Descriptor   = LocalChannel 6, RemoteChannel 1,
RemoteSlot 0x46
        Channel Descriptor   = LocalChannel 7, RemoteChannel 1,
RemoteSlot 0x47
        Channel Descriptor   = LocalChannel 8, RemoteChannel 1,
RemoteSlot 0x48
        Channel Descriptor   = LocalChannel 9, RemoteChannel 1,
RemoteSlot 0x49
        Channel Descriptor   = LocalChannel 10, RemoteChannel 1,
RemoteSlot 0x4A
        Channel Descriptor   = LocalChannel 11, RemoteChannel 1,
RemoteSlot 0x4B
```

```
        Channel Descriptor   = LocalChannel 12, RemoteChannel 1,
RemoteSlot 0x4C
        Channel Descriptor   = LocalChannel 13, RemoteChannel 1,
RemoteSlot 0x4D
        Channel Descriptor   = LocalChannel 14, RemoteChannel 1,
RemoteSlot 0x4E
        Channel Descriptor   = LocalChannel 15, RemoteChannel 1,
RemoteSlot 0x4F
        Channel Descriptor   = LocalChannel 16, RemoteChannel 1,
RemoteSlot 0x50
    P2P Slot Descriptor:
        Channel Type         = 0x0B PICMG®3.0 Base Interface
        LocalSlot/HW Address = 0x42
        Channel Count        = 0x0F
        Channel Descriptor   = LocalChannel 2, RemoteChannel 2,
RemoteSlot 0x41
        Channel Descriptor   = LocalChannel 3, RemoteChannel 2,
RemoteSlot 0x43
        Channel Descriptor   = LocalChannel 4, RemoteChannel 2,
RemoteSlot 0x44
        Channel Descriptor   = LocalChannel 5, RemoteChannel 2,
RemoteSlot 0x45
        Channel Descriptor   = LocalChannel 6, RemoteChannel 2,
RemoteSlot 0x46
        Channel Descriptor   = LocalChannel 7, RemoteChannel 2,
RemoteSlot 0x47
        Channel Descriptor   = LocalChannel 8, RemoteChannel 2,
RemoteSlot 0x48
        Channel Descriptor   = LocalChannel 9, RemoteChannel 2,
RemoteSlot 0x49
        Channel Descriptor   = LocalChannel 10, RemoteChannel 2,
RemoteSlot 0x4A
        Channel Descriptor   = LocalChannel 11, RemoteChannel 2,
RemoteSlot 0x4B
        Channel Descriptor   = LocalChannel 12, RemoteChannel 2,
RemoteSlot 0x4C
        Channel Descriptor   = LocalChannel 13, RemoteChannel 2,
RemoteSlot 0x4D
        Channel Descriptor   = LocalChannel 14, RemoteChannel 2,
RemoteSlot 0x4E
        Channel Descriptor   = LocalChannel 15, RemoteChannel 2,
RemoteSlot 0x4F
        Channel Descriptor   = LocalChannel 16, RemoteChannel 2,
RemoteSlot 0x50
 C0 02 65 2B  AE 5A 31 00  04 00 0B 41  0F 42 42 00
 43 61 00 44  81 00 45 A1  00 46 C1 00  47 E1 00 48
```

```
 01 01 49 21  01 4A 41 01  4B 61 01 4C  81 01 4D A1
 01 4E 01 01  4F E1 01 50  01 02 0B 42  0F 41 42 00
 43 62 00 44  82 00 45 A2  00 46 C2 00  47 E2 00 48
 02 01 49 22  01 4A 42 01  4B 62 01 4C  82 01 4D A2
 01 4E C2 01  4F E2 01 50  02 02

    PICMG Shelf Power Distribution Record (ID=0x11)
        Version = 0
    Feed count: 1
    Feed:
        Maximum External Available Current: 50.0 Amps
        Maximum Internal Current: Not specified
        Minimum Expected Operating Voltage: -40.5 Volts
        Feed-to-FRU Mapping entries count: 16
           FRU Addr: 41, FRU ID: 0xfe
           FRU Addr: 42, FRU ID: 0xfe
           FRU Addr: 43, FRU ID: 0xfe
           FRU Addr: 44, FRU ID: 0xfe
           FRU Addr: 45, FRU ID: 0xfe
           FRU Addr: 46, FRU ID: 0xfe
           FRU Addr: 47, FRU ID: 0xfe
           FRU Addr: 48, FRU ID: 0xfe
           FRU Addr: 49, FRU ID: 0xfe
           FRU Addr: 4a, FRU ID: 0xfe
           FRU Addr: 4b, FRU ID: 0xfe
           FRU Addr: 4c, FRU ID: 0xfe
           FRU Addr: 4d, FRU ID: 0xfe
           FRU Addr: 4e, FRU ID: 0xfe
           FRU Addr: 4f, FRU ID: 0xfe
           FRU Addr: 50, FRU ID: 0xfe
 C0 02 2C A7  6B 5A 31 00  11 00 01 F4  01 FF FF 51
 10 41 FE 42  FE 43 FE 44  FE 45 FE 46  FE 47 FE 48
 FE 49 FE 4A  FE 4B FE 4C  FE 4D FE 4E  FE 4F FE 50
 FE

    PICMG Shelf Activation And Power Management Record (ID=0x12)
        Version = 0
    Allowance for FRU Activation Readiness: 10 seconds
    FRU Activation and Power Description Count: 16
   Hw Address: 41, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds
```

```
    Hw Address: 42, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
         Shelf Manager Controlled Activation: Enabled
         Delay Before Next Power On: 0.0 seconds

    Hw Address: 43, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
         Shelf Manager Controlled Activation: Enabled
         Delay Before Next Power On: 0.0 seconds

    Hw Address: 44, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
         Shelf Manager Controlled Activation: Enabled
         Delay Before Next Power On: 0.0 seconds

    Hw Address: 45, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
         Shelf Manager Controlled Activation: Enabled
         Delay Before Next Power On: 0.0 seconds

    Hw Address: 46, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
         Shelf Manager Controlled Activation: Enabled
         Delay Before Next Power On: 0.0 seconds

    Hw Address: 47, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
         Shelf Manager Controlled Activation: Enabled
         Delay Before Next Power On: 0.0 seconds

    Hw Address: 48, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
         Shelf Manager Controlled Activation: Enabled
         Delay Before Next Power On: 0.0 seconds

    Hw Address: 49, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
         Shelf Manager Controlled Activation: Enabled
         Delay Before Next Power On: 0.0 seconds

    Hw Address: 4a, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
         Shelf Manager Controlled Activation: Enabled
         Delay Before Next Power On: 0.0 seconds
```

```
    Hw Address: 4b, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 4c, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 4d, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 4e, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 4f, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 50, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

 C0 82 57 81  E6 5A 31 00  12 00 0A 10  41 FE 96 00
 40 42 FE 96  00 40 43 FE  96 00 40 44  FE 96 00 40
 45 FE 96 00  40 46 FE 96  00 40 47 FE  96 00 40 48
 FE 96 00 40  49 FE 96 00  40 4A FE 96  00 40 4B FE
 96 00 40 4C  FE 96 00 40  4D FE 96 00  40 4E FE 96
 00 40 4F FE  96 00 40 50  FE 96 00 40

#
```

getacousticlevel

Syntax:

`getacousticlevel` *ETSI NEBS-A NEBS-U*

Purpose:

This command shows the system acoustic level and fan speed. This command is available for use with the server chassis. It allows users to get the current system acoustic level. When you set an acoustic level, the variable will be changed and the default cooling algorithm will set the fan speeds accordingly. This change will take effect dynamically, is persistant, and no ShMM reboot is required. See also `setacousticlevel`.

Note – These commands are applicable only if you are using the SUNCT900 cooling algorithm.

Example:

Get the system acoustic level and fan speed.

```
# clia getacousticlevel ETSI
Pigeon Point Shelf Manager Command Line Interpreter

System acoustic level - ETSI
fan speed             -  5#
```

getbootdev

Syntax:

`getbootdev` *IPMB-0-address* `[`*FRU-device-ID* | *IPMB-L-address*`]`

Purpose:

This command shows the system boot parameters for a designated IPM controller. If AdvancedMC access is not targeted, the second parameter should be set to `0` or omitted. The IPMB-L address for an AMC address is used if the second parameter exceeds 70h. Otherwise, the second parameter is treated as a FRU ID and converted to an IPMB-L address via AMC address mapping.

Examples:

Get the system boot options for IPM controller at IPMB address 82h.

```
# clia getbootdev 82
Pigeon Point Shelf Manager Command Line Interpreter

Get boot device option: status = 0x0 (0)
Response data (raw): 01 05 00 00 00 00 00
Decoded:
        Parameter version: 1
        Parameter valid = TRUE
        Boot option selector: 5
        Boot flags valid = FALSE
        Boot device selector: 0 (No override)
#
```

Get the system boot options for an AMC, where the carrier has IPMB-0 address 90h and the MMC has address IPMB-L address 72h.

```
# clia getbootdev 90 72
Pigeon Point Shelf Manager Command Line Interpreter

Get boot device option: status = 0x0 (0)
Response data (raw): 01 05 80 04 00 00 00
Decoded:
        Parameter version: 1
        Parameter valid = TRUE
        Boot option selector: 5
        Boot flags valid = TRUE
        Boot device selector: 1 (Force PXE)
#
```

getfanlevel

Syntax:

`getfanlevel` *IPMB-address fru-id*
`getfanlevel fan_tray` *n*

Purpose:

This command shows the current level of the fan controlled by the FRU specified in the command parameters.

Example:

Get fan level for the fan residing at FRU #2 at IPMB address 0x20.

```
# clia getfanlevel 20 2
Pigeon Point Shelf Manager Command Line Interpreter

20: FRU # 2 Override Fan Level: 1, Local Fan Level: 255
#
```

getfanpolicy

Syntax:

getfanpolicy [*fan-tray-addr* [*fan-tray-fru_id*]] [-s *addr*|*site_type* [*fru_id* | *site_number*]]

Purpose:

This retrieves information about Fan Tray(s) control mode and/or FRUs coverage by the specified Fan Tray(s). Notice that this command returns two different pieces of data: whether or not the site(s) are enabled/disabled for autonomous control by the Shelf Manager (based on `Set Fan Policy` commands), and whether or not the FRU site(s) are covered by the fans (according to the Fan Geography record).

The parameters *fan-tray-addr* and *fan-tray-fru_id* specify a fan tray. The command may accept no parameters, in this case the information about all Fan Trays and FRUs will be received.

If a numeric argument is expected to be treated as a hexadecimal, the "0x" prefix should be used, otherwise the error will be returned.

The flag `-s` precedes the parameters that define a site covered by the fan tray.

The *site_type* parameter can accept one of the following values: `Board`, `PEM`, `ShelfFRU`, `ShelfManager`, `FanTray`, `FanFilterTray`, `Alarm`, `Mezzanine`, `PMC`, `RTM`.

Examples:

Get fan policy for the fan tray at IPMB address 20h, FRU ID 3.

```
# clia getfanpolicy 0x20 3
Pigeon Point Shelf Manager Command Line Interpreter

Fan Tray: 0x20, FRU Id # 3
     Policy Type: Any Site
     Policy Timeout: 20 seconds
     Policy Applied: Tue Oct  17 02:32:06 2006
#
```

Get fan policy for the fan tray at IPMB address 20h, FRU ID 3, applied to the site at IPMB address 20h, FRU ID 1.

```
# clia getfanpolicy 0x20 3 -s 0x20 1
Pigeon Point Shelf Manager Command Line Interpreter

Fan Tray: 0x20, FRU Id # 3
    Policy Type: Per Site
    Policy Timeout: 20 seconds
    Policy Applied: Tue Oct  17 02:39:06 2006
    Site Type: Dedicated ShMC, Site Number: 1
    Site Covered: TRUE
#
```

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.

Information can be shown about a specific LED or all LEDs for the given FRU. IPMB address and FRU ID of the target LED can also be omitted. If FRU ID is omitted, information is shown about all LEDs on all FRUs of the given IPM controller. If IPMB address is also omitted, information is shown about all known LEDs in the shelf.

Examples:

Show LED state for all LEDs on the IPM controller at IPMB address FCh.

```
# clia getfruledstate fc
Pigeon Point Shelf Manager Command Line Interpreter

fc: FRU # 0, Led # 0 ("BLUE LED"):
    Local Control LED State: LED OFF

fc: FRU # 0, Led # 1 ("LED 1"):
    Local Control LED State: LED OFF
```

```
fc: FRU # 0, Led # 2 ("LED 2"):
    Local Control LED State: LED OFF

fc: FRU # 0, Led # 3 ("LED 3"):
    Local Control LED State: LED OFF

fc: FRU # 0, Led # 4 ("Application Specific LED# 1"):
    Local Control LED State: LED ON, color: GREEN
```

Show verbose information about LED state for all LEDs on the IPM controller at IPMB address FCh.

```
# clia getfruledstate -v  FC
Pigeon Point Shelf Manager Command Line Interpreter

fc: 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

fc: FRU # 0, Led # 1 ("LED 1"):
    Local Control LED State: LED OFF
    LED's color capabilities:
        Colors supported(0x0C): RED GREEN
        Default LED Color in Local Control State(0x03): GREEN
        Default LED Color in Override State(0x03): GREEN

fc: FRU # 0, Led # 2 ("LED 2"):
    Local Control LED State: LED OFF
    LED's color capabilities:
        Colors supported(0x0C): RED GREEN
        Default LED Color in Local Control State(0x03): GREEN
        Default LED Color in Override State(0x03): GREEN

fc: FRU # 0, Led # 3 ("LED 3"):
    Local Control LED State: LED OFF
    LED's color capabilities:
        Colors supported(0x0C): RED GREEN
        Default LED Color in Local Control State(0x02): RED
        Default LED Color in Override State(0x02): RED

fc: FRU # 0, Led # 4 ("Application Specific LED# 1"):
```

```
    Local Control LED State: LED ON, color: GREEN
    LED's color capabilities:
        Colors supported(0x0C): RED GREEN
        Default LED Color in Local Control State(0x02): RED
        Default LED Color in Override State(0x02): RED
```

Show LED state for FRU #0 of the IPM controller at IPMB address 20h.

```
# clia getfruledstate   20 0
Pigeon Point Shelf Manager Command Line Interpreter

20: FRU # 0, Led # 0 ("BLUE LED"):
    Local Control LED State: LED ON, color: BLUE

20: FRU # 0, Led # 1 ("LED 1"):
    Local Control LED State: LED OFF
```

Show LED state for LED #1 from FRU #0 of the IPM controller at IPMB address 20h.

```
# clia getfruledstate  -v  20 0 1
Pigeon Point Shelf Manager Command Line Interpreter

20: FRU # 0, Led # 1 ("LED 1"):
    Local Control LED State: LED OFF
    LED's color capabilities:
        Colors supported(0x04): RED
        Default LED Color in Local Control State(0x02): RED
        Default LED Color in Override State(0x02): RED
```

gethysteresis

Syntax:

`gethysteresis [`*IPMB-address* `[[`*lun*`:]`*sensor id* | *sensor name*`]]`

Purpose:

This command shows the current hysteresis values for the specified sensor(s). The sensors must be threshold based. Both raw and processed values are shown.

The command allows the user to qualify the sensor number with the Logical Unit Number (LUN) if the targets controller supports sensors on multiple LUNs. If the LUN is omitted, the current hysteresis values for all sensors with the specified sensor number are shown. *lun* can take the value 0, 1, or 3. (LUN 2 is reserved.) Sensor names are not qualified with LUN numbers, since it is assumed that sensor names will normally be unique within the controller. However, if there are several sensors with the same name within the controller, information is shown about all of them. If *IPMB-address* is omitted, the current hysteresis levels for all sensors for the specified IPMB address are shown.

Example:

Show the hysteresis values for sensor # 2 on the IPM controller at IPMB address FCh.

```
# clia gethysteresis FC 2
Pigeon Point Shelf Manager Command Line Interpreter

fc: LUN: 0, Sensor # 2 ("lm75 temp")
    Type: Threshold (0x01), "Temperature"(0x01)
        Positive hysteresis, Raw data: 0x00   Processed data:
0.00000 degrees C
        Negative hysteresis, Raw data: 0x00   Processed data:
0.00000 degrees C
```

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. The state is taken from the sensor data provided by the IPMB Link sensor on the target IPMC (sensor type F1). Information about both buses A and B is printed.

The command works differently in bused and radial environments. In a bused environment, or in a radial environment if the target IPMC is not an IPMB hub, the argument *link* is not used. Information about the state of IPMB-A and IPMB-B on the target IPM controller is shown.

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. The state is taken from the sensor data of the multiple IPMB link sensors on the IPM controller.
- If the *link* is present, the command prints information about the specific radial IPMB link (1 to 95). The state of the link is taken from the state of the corresponding IPMB link sensor on the IPM controller.

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

Examples:

Show the current state of IPMB-0 on the IPMC at IPMB address 92h.

```
# clia getipmbstate 92
Pigeon Point Shelf Manager Command Line Interpreter

92: LUN: 0, Sensor # 1 ("IPMB LINK")
 Bus Status: 0x8  (IPMB-A Enabled, IPMB-B Enabled)
 IPMB A State: 0x8  (LocalControl, No failure)
 IPMB B State: 0x8  (LocalControl, No failure)
```

Show the current state of link 8 for the Shelf Manager in the radial environment.

```
# clia getipmbstate 20 8
Pigeon Point Shelf Manager Command Line Interpreter

20: Link: 8, LUN: 0, Sensor # 12 ("IPMB LINK 8")
 Bus Status: 0x8  (IPMB-A Enabled, IPMB-B Enabled)
 IPMB A State: 0x8  (LocalControl, No failure)
 IPMB B State: 0x8  (LocalControl, No failure)
```

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 A-1 lists names and numbers of LAN configuration parameters supported by the `getlanconfig` command:

TABLE A-1 LAN Configuration Parameters for `getlanconfig`

Parameter Name	Number	Description
`auth_support`	1	An 8-bit value that contains authentication types support flags for the LAN channel.
`auth_enables`	2	Five 8-bit values that contain authentication types 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`).

TABLE A-1 LAN Configuration Parameters for `getlanconfig` *(Continued)*

`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.
`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)

TABLE A-1 LAN Configuration Parameters for `getlanconfig` *(Continued)*

`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:

The following subsections provide more detailed information about each of the supported parameters.

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: 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: 172.16.2.203
IP Address Source: Static Address (Manually Configured) (01)
MAC Address: 90:91:91:91:91:91
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: 02
    Enable BMC-generated Gratuitous Response
Gratuitous ARP Interval: 2.0 seconds
Default Gateway Address: 0.0.0.0
Default Gateway MAC Address: N/A
Backup Gateway Address: 0.0.0.0
Backup MAC Address: N/A
Community String: "public"
Number of Destinations: 16
#
```

auth_support

Syntax:

getlanconfig *channel* auth_support
getlanconfig *channel* 1

Purpose:

This command shows the current value of the LAN parameter `auth_support`. This parameter specifies which authentication types are supported by the Shelf Manager, represented by a single byte, treated as a bit mask with the bits defined as follows:

- 0x01 – None
- 0x02 – MD2
- 0x04 – MD5
- 0x10 – Straight password/key
- 0x20 – OEM proprietary

Other bits are reserved and should be set to 0.

Besides the raw hexadecimal value, symbolic values for the bits that are set are also shown.

Example:

```
# clia getlanconfig 1 auth_support
Pigeon Point Shelf Manager Command Line Interpreter

Authentication Type Support: 0x15 ( None MD5 Straight Password/Key
#
```

auth_enables

Syntax:

getlanconfig *channel* auth_enables
getlanconfig *channel* 2

Purpose:

This command shows the current value of the LAN parameter auth_enables. This parameter specifies which authentication types are currently enabled by the Shelf Manager for each of five supported privilege levels (Callback, User, Administrator, Operator, and OEM), represented by a sequence of five bytes, each corresponding to the respective privilege level, treated as a bit mask with the bits defined as follows:

- 0x01 – None
- 0x02 – MD2
- 0x04 – MD5
- 0x10 – Straight password/key
- 0x20 – OEM proprietary

Other bits are reserved and should be set to 0.

Besides the raw hexadecimal values, symbolic values for the bits that are set are also shown.

Example:

```
# clia getlanconfig 1 auth_enables
Pigeon Point Shelf Manager Command Line Interpreter

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

Syntax:

getlanconfig *channel* ip
getlanconfig *channel* 3

Purpose:

This command shows the current IP address used by the channel, in dotted decimal notation.

Example:

```
# clia getlanconfig 1 ip
Pigeon Point Shelf Manager Command Line Interpreter

IP Address: 172.16.2.203
```

ip_source

Syntax:

getlanconfig *channel* ip_source
getlanconfig *channel* 4

Purpose:

This command shows the current value of the LAN parameter `ip_source`. This parameter specifies the source of the IP Address used by the Shelf Manager, represented by a single byte, which can have one of the following values:

- 0 – Unspecified
- 1 – Static address (manually configured)
- 2 – address obtained by Shelf Manager running DHCP
- 3 – address loaded by BIOS or system software
- 4 – address obtained by Shelf Manager running other address assignment protocol

Other values are reserved.

Besides the raw hexadecimal value, the symbolic value is also shown.

Example:

```
# clia getlanconfig 1 ip_source
Pigeon Point Shelf Manager Command Line Interpreter

IP Address Source: Static Address (Manually Configured) (0x01)
#
```

mac

Syntax:

getlanconfig *channel* mac
getlanconfig *channel* 5

Purpose:

This command shows the current MAC address used by the channel, in the form of six hexadecimal bytes separated by colons.

Example:

```
# clia getlanconfig 1 mac
Pigeon Point Shelf Manager Command Line Interpreter

MAC Address: 90:91:91:91:91:91
#
```

subnet_mask

Syntax:

getlanconfig *channel* subnet_mask
getlanconfig *channel* 6

Purpose:

This command shows the current IP subnet mask used by the channel, in dotted decimal notation.

Example:

```
# clia getlanconfig 1 subnet_mask
Pigeon Point Shelf Manager Command Line Interpreter

Subnet Mask: 255.255.255.0
#
```

ipv4_hdr_param

Syntax:

getlanconfig *channel* ipv4_hdr_param
getlanconfig *channel* 7

Purpose:

This command shows the current IP 4 header parameters. They are represented as three single-byte values in hexadecimal notation, separated with colons. The content of the bytes conforms to section 19.2 of the IPMI 1.5 specification.

Example:

```
# clia getlanconfig 1 ipv4_hdr_param
Pigeon Point Shelf Manager Command Line Interpreter

IPv4 Header Parameters: 0x40:0x40:0x10
#
```

pri_rmcp_port

Syntax:

getlanconfig *channel* pri_rmcp_port
getlanconfig *channel* 8

Purpose:

This command shows the current RMCP primary port used by the channel, in hexadecimal. This is the port used for regular interactions via RMCP.

Example:

```
# clia getlanconfig 1 pri_rmcp_port
Pigeon Point Shelf Manager Command Line Interpreter

Primary RMCP Port Number: 0x026f
#
```

sec_rmcp_port

Syntax:

```
getlanconfig channel sec_rmcp_port
getlanconfig channel 9
```

Purpose:

This command shows the current RMCP secondary port used by the channel, in hexadecimal. This is the port used for secure interactions via RMCP.

Example:

```
# clia getlanconfig 1 sec_rmcp_port
Pigeon Point Shelf Manager Command Line Interpreter

Primary RMCP Port Number: 0x0298
#
```

arp_control

Syntax:

```
getlanconfig channel arp_control
getlanconfig channel 10
```

Purpose:

This command shows the current value of the LAN parameter `arp_control`. This parameter specifies additional ARP support provided by the Shelf Manager, represented by a single byte, treated as a bit mask with the bits defined as follows:

- 1 – Enable Shelf Manager-generated Gratuitous ARPs
- 2 – Enable Shelf Manager-generated ARP responses

Other bits are reserved and should be set to 0.

Besides the raw hexadecimal value, symbolic values for the bits that are set are also shown.

Example:

```
# clia getlanconfig 1 arp_control
Pigeon Point Shelf Manager Command Line Interpreter

BMC-generated ARP Control: 02
    Enable BMC-generated Gratuitous Response
#
```

arp_interval

Syntax:

getlanconfig *channel* arp_interval
getlanconfig *channel* 11

Purpose:

This command shows the current ARP interval used by the channel. The value is shown as the number of seconds in fixed-point numeric format.

Example:

```
# clia getlanconfig 1 arp_interval
Pigeon Point Shelf Manager Command Line Interpreter

Gratuitous ARP Interval: 2.0 seconds
#
```

dft_gw_ip

Syntax:

getlanconfig *channel* dft_gw_ip
getlanconfig *channel* 12

Purpose:

This command shows the IP address of the default gateway used by the channel, in dotted decimal notation.

Example:

```
# clia getlanconfig 1 dft_gw_ip
Pigeon Point Shelf Manager Command Line Interpreter

Default Gateway Address: 0.0.0.0
#
```

dft_gw_mac

Syntax:

getlanconfig *channel* dft_gw_mac
getlanconfig *channel* 13

Purpose:

This command shows the MAC address of the default gateway used by the channel, in the form of six hexadecimal bytes separated by colons.

Example:

```
# clia getlanconfig 1 dft_gw_mac
Pigeon Point Shelf Manager Command Line Interpreter

Default Gateway MAC Address: N/A
#
```

backup_gw_ip

Syntax:

getlanconfig *channel* backup_gw_ip
getlanconfig *channel* 14

Purpose:

This command shows the IP address of the backup gateway used by the channel, in dotted decimal notation.

Example:

```
# clia getlanconfig 1 backup_gw_ip
Pigeon Point Shelf Manager Command Line Interpreter

Backup Gateway Address: 0.0.0.0
#
```

backup_gw_mac

Syntax:

getlanconfig *channel* backup_gw_mac
getlanconfig *channel* 15

Purpose:

This command shows the MAC address of the backup gateway used by the channel, in the form of six hexadecimal bytes separated by colons.

Example:

```
# clia getlanconfig 1 backup_gw_mac
Pigeon Point Shelf Manager Command Line Interpreter

Backup Gateway MAC Address: N/A
#
```

community

Syntax:

getlanconfig *channel* community
getlanconfig *channel* 16

Purpose:

This command shows the community string parameter used in PET traps.

Example:

```
# clia getlanconfig 1 community
Pigeon Point Shelf Manager Command Line Interpreter

Community String: "public"
#
```

destination_count

Syntax:

getlanconfig *channel* destination_count
getlanconfig *channel* 17

Purpose:

This command shows the maximum number of alert destinations available for the channel. This is a configuration parameter for the Shelf Manager and can be changed only through the `shelfman` configuration file.

Example:

```
# clia getlanconfig 1 destination_count
Pigeon Point Shelf Manager Command Line Interpreter

Number of Destinations: 16
#
```

destination_type

Syntax:

getlanconfig *channel* destination_type [*set-selector*]
getlanconfig *channel* 18 [*set-selector*]

Purpose:

This command shows the element of the destination table with the index equal to *set-selector*. Indexes are 0-based. Selector 0 is used to address the volatile destination. The following information is shown about the destination:

- Destination selector
- Alert destination type (PET Trap or OEM destination; whether the alert should be acknowledged)
- Alert acknowledge timeout
- Retry count

If the set selector is omitted, all active destinations are shown, with their numbers.

Examples:

```
# clia getlanconfig 1 destination_type 2
Pigeon Point Shelf Manager Command Line Interpreter

DST Type # 2, Type: Acknowledged PET Trap Destination (0x80), ACK
Timeout / Retry Interval: 3 seconds, Retries: 5

# clia getlanconfig 1 destination_type
Pigeon Point Shelf Manager Command Line Interpreter

DST Type # 0, Type: Acknowledged reserved (0x81), ACK Timeout /
Retry Interval: 2 seconds, Retries: 6
DST Type # 1, Type: Unacknowledged reserved (0x02), ACK Timeout /
Retry Interval: 3 seconds, Retries: 4
DST Type # 2, Type: Acknowledged PET Trap Destination (0x80), ACK
Timeout / Retry Interval: 3 seconds, Retries: 5
#
```

destination_address

Syntax:

`getlanconfig` *channel* `destination_address [`*set-selector*`] getlanconfig` *channel* `19 [`*set-selector*`]`

Purpose:

This command shows the element of the destination address table with the index equal to *set-selector*. Indexes are 0-based. Selector 0 is used to address the volatile destination. The following information is shown about the destination:

- Destination selector
- Address format (IP+MAC by default)
- Destination IP address
- Destination MAC address
- Which gateway to use (default vs. backup).

If the set selector is omitted, all active destination addresses are shown, with their numbers.

Example:

```
# clia getlanconfig 1 destination_address 2
Pigeon Point Shelf Manager Command Line Interpreter

DST Addresses # 2, Address Format: IPv4 IP Address followed by DIX
ethernet / 802.3 MAC Address (0x00)
    Gateway: Default (0x00), Alerting IP: 172.16.2.100, Alerting
MAC: 90:93:93:93:93:93
#
```

getmgmtportroute

Syntax:

`getmgmtportroute` [*slot-number*]

Purpose:

This command displays the management port routing configuration. This information is stored on the midplane as an OEM record.

Use the *slot-number* option to display the management port routing information for a specified slot. If slot number is omitted, information will be returned for all slots. This data will be from the midplane only.

The configuration is persistent and designated for each slot, regardless of blade presence.

- For setting management route configuration, see “`setmgmtportroute`” on page 284 command.
- For querying the IPMC to determine management port state or routing configuration, see “`mgmtportstate`” on page 247 command.
- For configuring management port (front or rear access), see “Set Management Port” on page 363.

Example:

Display the management port routing information for the specified slot.

```
# clia getmgmtportroute 4
Pigeon Point Shelf Manager Command Line Interpreter

```

```
MGMT port Routing Information from Midplane FRU
-------------------------------------------
Slot   MGMT
-------------------------------------------
 4     Front
```

getmuxconfig

Syntax:

`getmuxconfig` [*slot-number*]

Purpose:

This command displays the multiplexer (MUX) configuration information that is stored in the shelf's midplane OEM multi record for a specified slot. If the *slot-number* is not specified, then routing information for all the slots displayed. If the blade doesn't have a MUX controller, the IPMC will ignore the mux-routing information sent by ShMM. The MUX controller is used to route the NIU ports to Zone 2 (backplane) and/or Zone 3 (ARTM). Use the *slot-number* option to display the MUX configuration information for a specified slot.

The Shelf Manager provides the configuration information read from the storage (midplane FRU) to the IPMC and the IPMC has to program the MUX prior to blade activation. See `setmuxconfig` and `muxstate` commands for more information.

Example:

Display the port routing information for the MUX on the node board in physical slot 5.

```
# clia getmuxconfig 5
Pigeon Point Shelf Manager Command Line Interpreter

10GbE NIU-XAUI Routing Information from Midplane FRU
-------------------------------------------
Slot   XAUI-1           XAUI-2
-------------------------------------------
 5     Zone3 (ARTM)    Zone2 (Back Plane)
```

getpefconfig

Syntax:

```
getpefconfig
getpefconfig parameter-name [additional-parameters]
getpefconfig parameter-number [additional-parameters]
```

Purpose:

This command shows the value of the specified PEF configuration parameter. If neither the configuration parameter name nor the parameter-number is specified, all PEF configuration parameters are shown.

TABLE A-2 lists names and numbers of PEF configuration parameters:

TABLE A-2 PEF Configuration Parameters

Parameter Name	Number	Description
`control`	1	An 8-bit value that represents control flags for PEF (enable PEF, enable PEF startup delay, etc.).
`action_control`	2	An 8-bit value that represents PEF action global control flags (enable reset, enable power down, etc.).
`startup_delay`	3	Time to delay PEF after system power-ups and resets, in seconds.
`alert_startup_delay`	4	Time to delay alerts after system power-ups and resets, in seconds.
`event_filter_count`	5	Maximum number of event filters.
`event_filter`	6	An event filter table entry identified by the specified set selector. If no set selector is given, all active event filters are shown.
`event_filter_data1`	7	The first byte of the event filter table entry identified by the specified set selector. If no set selector is given, all active event filters are shown.
`alert_policy_count`	8	Maximum number of alert policies.
`alert_policy`	9	An alert policy table entry identified by the specified set selector. If no set selector is given, all active alert policies are shown.
`system_guid`	10	A GUID used to fill in the GUID field in the PET trap.
`alert_string_count`	11	Maximum number of alert strings.

TABLE A-2 PEF Configuration Parameters *(Continued)*

alert_string_key	12	An alert string key identified by the specified set selector. If no set selector is given, all alert string keys are shown.
alert_string	13	An alert string identified by the specified set selector. If no set selector is given, all alert strings are shown.
oem_filter_count	96	Maximum number of OEM filters.
oem_filter	97	An OEM filter table entry identified by the specified set selector. If no set selector is given, all active event filters are shown.

Example:

Get and show the whole PEF parameter table.

```
# clia getpefconfig
Pigeon Point Shelf Manager Command Line Interpreter

PEF parameters:
    PEF control: 0x00
    PEF Action Global Control: 0x00
    PEF Startup Delay: 60 seconds
    PEF Alert Startup Delay: 60 seconds
    PEF Number of Event Filters: 64
    PEF Number of OEM Filters: 16
    Active Event Filters:
        None
    Active event filter data:
        None
    Alert Policies Count: 64
    Policy:
        None
    PEF GUID: Using the system GUID
Alert Strings Count: 64
    Alert string key:
        None
    Alert Strings:
        None
#
```

The following subsections provide more detailed information about each of the supported parameters.

control

Syntax:

```
getpefconfig control
getpefconfig 1
```

Purpose:

This command shows the current value of the PEF parameter `control`. This parameter is a single byte, treated as a bit mask with the bits defined as follows:

- 0x01 – Enable PEF
- 0x02 – Enable generation of event messages for PEF actions
- 0x04 – Enable PEF startup delays on system power-ups and resets
- 0x08 – Enable PEF Alert Startup delays

Other bits are reserved and should be set to 0.

Example:

```
# clia getpefconfig control
Pigeon Point Shelf Manager Command Line Interpreter

PEF control: 0x07
        Enable PEF
        Enable Event Message for PEF Actions
        Enable PEF Startup Delay
#
```

action_control

Syntax:

```
getpefconfig action_control
getpefconfig 2
```

Purpose:

This command shows the current value of the PEF parameter `action_control`. This parameter is a single byte, treated as a bit mask with the bits defined as follows:

- 0x01 – Enable alert action
- 0x02 – Enable power down action
- 0x04 – Enable reset action
- 0x08 – Enable power cycle action
- 0x10 – Enable OEM action
- 0x20 – Enable diagnostic interrupt

Other bits are reserved and should be set to 0.

Example:

```
# clia getpefconfig action_control
Pigeon Point Shelf Manager Command Line Interpreter

PEF Action Global Control: 0x3f
        Enable Alert Action
        Enable Power Down Action
        Enable Reset Action
        Enable Power Cycle Action
        Enable OEM Action
        Enable Diagnostic Interrupt
#
```

startup_delay

Syntax:

```
getpefconfig startup_delay
getpefconfig 3
```

Purpose:

This command shows the current value of the PEF parameter `startup_delay`. This parameter is a single byte, representing the number of seconds that the PEF facility delays at startup.

Example:

```
# clia getpefconfig startup_delay
Pigeon Point Shelf Manager Command Line Interpreter

    PEF Startup Delay: 60 seconds
#
```

alert_startup_delay

Syntax:

```
getpefconfig startup_delay
getpefconfig 4
```

Purpose:

This command shows the current value of the PEF parameter `alert_startup_delay`. This parameter is a single byte, representing the number of seconds that the alerting facility delays at startup.

Example:

```
# clia getpefconfig alert_startup_delay
Pigeon Point Shelf Manager Command Line Interpreter

PEF Alert Startup Delay: 60 seconds
#
```

event_filter_count

Syntax:

```
getpefconfig event_filter_count
getpefconfig 5
```

Purpose:

This command shows the current value of the PEF parameter `event_filter_count`. This read-only value is the size of the event filter table. This value is a configuration parameter for the Shelf Manager and can be changed only through the `shelfman` configuration file.

Example:

```
# clia getpefconfig event_filter_count
Pigeon Point Shelf Manager Command Line Interpreter

PEF Number of Event Filters: 64
#
```

event_filter

Syntax:

```
getpefconfig event_filter [set-selector]
getpefconfig 6 [set-selector]
```

Purpose:

This command shows the element of the event filter table with index equal to *set-selector*. Indexes are 1-based. The following information is shown about each event filter:

- Filter configuration: whether the filter is software configured or manufacturer pre-configured
- Event filter action mask
- Alert policy number
- Event severity
- Event source address to match (255 = any address)
- Source Channel/LUN to match (255 = match any source channel/LUN)
- Sensor type to match
- Sensor number to match
- Event trigger (event/reading type) to match
- Event offset mask
- AND, Compare 1 (CMP1), and Compare 2 (CMP2) masks for event data bytes 1, 2, and 3.

If the set selector is omitted, all active event filter table entries are shown, with their numbers.

Example:

```
# clia getpefconfig event_filter 2
Pigeon Point Shelf Manager Command Line Interpreter

    Active Event Filters:
    0x02: Software Configurable Filter
        Action Mask: 0x01
        Policy Number: 1, Severity: Critical Condition
        Source Address: 0x20, LUN: 3, Channel: 15
        Sensor Type: Hot Swap (0xf0), Sensor # 255 (ANY)
        Event Trigger: 0xff (ANY), Event Offset Mask: 0xffff
        0: AND: 0x0f, CMP1: 0xff, CMP2: 0x00
        1: AND: 0x00, CMP1: 0x00, CMP2: 0x00
        2: AND: 0xff, CMP1: 0xff, CMP2: 0x00
#
```

event_filter_data1

Syntax:

`getpefconfig event_filter_data1 [`*set-selector*`]`
`getpefconfig 7 [`*set-selector*`]`

Purpose:

This command shows the first byte of the element of the event filter table with the index equal to *set-selector*. Indexes are 1-based. This byte is shown in hexadecimal. Bits in this byte have the following meaning:

- 0x80 – This filter is enabled.
- 0x40 – This filter is pre-configured by the manufacturer and should not be altered by software.

Other bits are reserved and should be 0.

If the set selector is omitted, first byte for each of the active event filter table entries is shown, with the corresponding filter numbers.

Example:

```
# clia getpefconfig event_filter_data1 2
Pigeon Point Shelf Manager Command Line Interpreter

    Active event filter data:
        0x02: 0x80 Enabled 1, Configuration: 0 ("Software
Configurable Filter")
#
```

alert_policy_count

Syntax:

```
getpefconfig alert_policy_count
getpefconfig 8
```

Purpose:

This command shows the current value of the PEF parameter `alert_policy_count`. This read-only value is the size of the alert policy table. This value is a configuration parameter for the Shelf Manager and can be changed only through the `shelfman` configuration file.

Example:

```
# clia getpefconfig alert_policy_count
Pigeon Point Shelf Manager Command Line Interpreter

    Alert Policies Count: 64
#
```

alert_policy

Syntax:

getpefconfig alert_policy [*set-selector*]
getpefconfig 9 [*set-selector*]

Purpose:

This command shows the element of the alert policy table with index equal to *set-selector*. Indexes are 1-based. The following information is shown about each alert policy:

- the policy number
- the policy type (with respect to the alert sent to the previous destination)
- destination channel number
- destination selector
- alert string key.

If the set selector is omitted, all active alert policy table entries are shown, with their numbers.

Example:

```
# clia getpefconfig alert_policy 2
Pigeon Point Shelf Manager Command Line Interpreter

    Policy:
       0x02: Policy# 5, Policy Type: 0, Channel: 1, DST: 1, Alert
String Sel: 1
#
```

system_guid

Syntax:

```
getpefconfig system_guid
getpefconfig 10
```

Purpose:

This command shows the current value of the PEF parameter `system_guid`. This parameter represents the GUID that is sent in a PET Trap PDU to an alert destination. This GUID may be defined as a separate GUID or as being equal to the System GUID (which can be obtained via the `Get System GUID IPMI` command).

Example:

```
# clia getpefconfig system_guid
Pigeon Point Shelf Manager Command Line Interpreter

    PEF GUID: 23662f7f-ba1b-4b65-8808-94ca09c9bbb0
#
```

alert_string_count

Syntax:

```
getpefconfig alert_string_count
getpefconfig 11
```

Purpose:

This command shows the current value of the PEF parameter `alert_string_count`. This read-only value is the size of the alert string table, which is the maximum number of alert strings in simultaneous use. This value is the configuration parameter for the Shelf Manager and can be changed only through the `shelfman` configuration file.

Example:

```
# clia getpefconfig alert_string_count
Pigeon Point Shelf Manager Command Line Interpreter

    Alert Strings Count: 64
#
```

alert_string_key

Syntax:

getpefconfig alert_string_key [*set-selector*]
getpefconfig 12 [*set-selector*]

Purpose:

This command shows the element of the alert string key table with index equal to *set-selector*. Indexes are 1-based. Index 0 can be used to designate the volatile alert string. Each key associates an event filter with an alert string for alert generation purposes. The following information is shown about each alert string key:

- the alert string key number
- the associated event filter number
- the associated alert string number

If the set selector is omitted, all active alert string key table entries are shown with their numbers.

Example:

```
# clia getpefconfig alert_string_key 2
Pigeon Point Shelf Manager Command Line Interpreter

   Alert string key: set selector 2, event_filter 0x10, string_set
0x11
#
```

alert_string

Syntax:

getpefconfig alert_string [*set-selector*]
getpefconfig 13 [*set-selector*]

Purpose:

This command shows the element of the alert string table with index equal to *set-selector*. Indexes are 1-based. Index 0 can be used to designate the volatile alert string. This command shows the whole string at once.

If the set selector is omitted, all defined alert strings are shown with their numbers.

Example:

```
# clia getpefconfig alert_string 2
Pigeon Point Shelf Manager Command Line Interpreter

    Alert Strings:
        0x02: "This is the alert string"
#
```

oem_filter_count

Syntax:

```
getpefconfig oem_filter_count
getpefconfig 96
```

Purpose:

This command shows the current value of the PEF parameter `oem_filter_count`. This read-only value is the size of the OEM filter table. This value is a configuration parameter for the Shelf Manager and can be changed only through the `shelfman` configuration file.

The OEM filter table is a Pigeon Point Systems-defined OEM extension of the IPMI specification. It allows PEF to be applied, in addition to platform events, to OEM-timestamped and non-timestamped SEL entries (record type range C0h–FFh).

Example:

```
# clia getpefconfig oem_filter_count
Pigeon Point Shelf Manager Command Line Interpreter

    PEF Number of OEM Filters: 16
#
```

oem_filter

Syntax:

getpefconfig oem_filter [*set-selector*]
getpefconfig 97 [*set-selector*]

Purpose:

The OEM filter table is a Pigeon Point Systems-defined OEM extension of the IPMI specification. It allows PEF to be applied, in addition to platform events, to OEM-timestamped and non-timestamped SEL entries (record type range C0h-FFh).

Each entry of the OEM filter table defines the range of record types (in the range of OEM record types), to which this OEM filter applies, and the alert policy number that is to be invoked when a record with the matching record type is placed in the SEL.

This command shows the element of the OEM filter table with index equal to *set-selector*. Indexes are 1-based. The following information is shown about each OEM filter:

- Byte 1: SEL Record Type Range Low boundary
- Byte 2: SEL Record type Range high boundary
- Byte 3: Alert policy number that will be invoked for SEL entries that have record types matching the range specified in Bytes 1 and 2.

If the set selector is omitted, all active OEM filter table entries are shown, with their numbers.

Example:

```
# clia getpefconfig oem_filter
Pigeon Point Shelf Manager Command Line Interpreter

    Active OEM Filters:
    0x01: OEM range boundary 0xff:0xff, alert policy # 1
#
```

getsensoreventenable

Syntax:

`getsensoreventenable [`*IPMB-address* `[`*sensor-name* `|[`*lun*`:]`*sensor-number*`]`
`getsensoreventenable board` *n* `[`*sensor-name* `| [`*lun*`:]`*sensor-number*`]]`
`getsensoreventenable shm` *n* `[`*sensor-name* `| [`*lun*`:]`*sensor-number*`]]`

This command shows the current event enable mask values of the specified sensors.

This command allows the user to qualify the sensor number with the Logical Unit Number (LUN) if the target controller supports sensors on multiple LUNs. If the LUN is omitted, information about sensors with the specified sensor number on all LUNs is shown. *lun* can take the value 0, 1, or 3. (LUN 2 is reserved.)

Sensor names are not qualified with LUN numbers, since it is assumed that sensor names will normally be unique within the controller. However, if there are several sensors with the same name within the controller, information is shown about all of them.

This command shows the current sensor event mask values for the supported events of the specified sensors. The following attributes for each sensor are also shown:

- 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

Examples:

Get event enable values for a temperature sensor `Local Temp` on IPM controller FE.

```
# clia getsensoreventenable -v fe "Local Temp"
Pigeon Point Shelf Manager Command Line Interpreter

fe: LUN: 0, Sensor # 3 ("Local Temp")
    Type: Threshold (0x01), "Temperature" (0x01)
    Assertion event mask:   0x0a80
        Assertion event for "Upper Non-Recoverable Going High"
enabled
        Assertion event for "Upper Critical Going High" enabled
       Assertion event for "Upper Non-Critical Going High" enabled
    Deassertion event mask: 0x0a80
        Deassertion event for "Upper Non-Recoverable Going High"
enabled
        Deassertion event for "Upper Critical Going High" enabled
        Deassertion event for "Upper Non-Critical Going High"
enabled
#
```

Get event enable information for the same sensor but specify sensor LUN and number.

```
# clia getsensoreventenable -v fe 0:3
Pigeon Point Shelf Manager Command Line Interpreter

fe: LUN: 0, Sensor # 3 ("Local Temp")
    Type: Threshold (0x01), "Temperature" (0x01)
    Assertion event mask:   0x0a80
        Assertion event for "Upper Non-Recoverable Going High"
enabled
        Assertion event for "Upper Critical Going High" enabled
       Assertion event for "Upper Non-Critical Going High" enabled
    Deassertion event mask: 0x0a80
        Deassertion event for "Upper Non-Recoverable Going High"
enabled
```

```
        Deassertion event for "Upper Critical Going High" enabled
        Deassertion event for "Upper Non-Critical Going High"
enabled
#
```

getthreshold | threshold

Syntax:

getthreshold [*IPMB-address* [*sensor-name* | [*lun*:]*sensor-number*]]
getthreshold board *n* [*sensor-name* | [*lun*:]*sensor-number*]]
getthreshold shm *n* [*sensor-name* | [*lun*:]*sensor-number*]]

The verb threshold can also be used instead of getthreshold.

Purpose:

This command shows the current threshold values for the supported thresholds of the specified sensors. The sensor must be a threshold-based sensor. Both raw and processed values are shown. The following attributes for each sensor are also shown:

- 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

This command allows the user to qualify the sensor number with the Logical Unit Number (LUN) if the target controller supports sensors on multiple LUNs. If the LUN is omitted, information about sensors with the specified sensor number on all LUNs is shown. *lun* can take the value 0, 1, or 3. (LUN 2 is reserved.)

Sensor names are not qualified with LUN numbers, since it is assumed that sensor names will normally be unique within the controller. However, if there are several sensors with the same name within the controller, information is shown about all of them.

Examples:

Get threshold values for a temperature sensor `Local Temp` on IPM controller FE.

```
# clia getthreshold -v fe "Local Temp"
Pigeon Point Shelf Manager Command Line Interpreter

fe: LUN: 0, Sensor # 3 ("Local Temp")
    Type: Threshold (0x01), "Temperature" (0x01)
        Lower Critical Threshold, Raw Data: 0x80, Processed Data:
-128.000000 degrees C
        Upper Non-Critical Threshold, Raw Data: 0x50, Processed
Data: 80.000000 degrees C
        Upper Critical Threshold, Raw Data: 0x50, Processed Data:
80.000000 degrees C
       Upper Non-Recoverable Threshold, Raw Data: 0x50, Processed
Data: 80.000000 degrees C
#
```

Get threshold information for the same sensor but specify sensor LUN and number.

```
# clia getthreshold -v fe 0:3
Pigeon Point Shelf Manager Command Line Interpreter

fe: LUN: 0, Sensor # 3 ("Local Temp")
    Type: Threshold (0x01), "Temperature" (0x01)
        Lower Critical Threshold, Raw Data: 0x80, Processed Data:
-128.000000 degrees C
        Upper Non-Critical Threshold, Raw Data: 0x50, Processed
Data: 80.000000 degrees C
        Upper Critical Threshold, Raw Data: 0x50, Processed Data:
80.000000 degrees C
       Upper Non-Recoverable Threshold, Raw Data: 0x50, Processed
Data: 80.000000 degrees C
#
```

help

Syntax:

help [*command* [*subcommand*]]

Purpose:

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

Examples:

Get a list of commands and their syntax.

```
# clia help
Pigeon Point Shelf Manager Command Line Interpreter

Command Line Interface command set:
Parameters are case insensitive
In general:
        IPMB address is hexadecimal ALWAYS.
        All other numbers may be either decimal and hexadecimal (0x notation
        required for hexadecimal numbers)
        -v turns on verbose output

  activate <addr> <fru_id>
  alarm <alarm status/action>
  board [slot_number]
  boardreset <slot number>
  busres force <res>
  busres info [<res>]
  busres lock <res>
  busres query [-v] <res> [<target> [noupdate]]
  busres release <res>
  busres sendbusfree <res> <target>
  busres setowner <res> <target>
  busres unlock <res>
  deactivate <addr> <fru_id>
  debuglevel [<mask> [<console mask>]]
  exit
  fans <addr> <fru id>
  fru [<addr> [id=<fru_id> | type=<site_type>]] | [type=<site_type>
           [/<site_number>]]
  frucontrol <addr> <fru_id> <command>
  frudata [<addr>] [<fru id>] [<block number>]
```

```
frudata shm <N> [<block number>]
frudata <addr> <fru id> <byte offset> <byte_1> [byte2 .. [byte_16]]
frudatar <addr> <fru id> <file name>
frudataw <addr> <fru id> <file name>
fruinfo <addr> <fru_id>
getbootdev <addr> [<fru-id> | <amc-addr>]
getfanlevel <addr> <fru_id>
getfanpolicy [<addr> [<fru_id>]] [-s <addr>|site_type
                   [<fru_id>|site_number]]
getfruledstate [-v] [<addr> [<fru_id> [<LedId>|ALL]]]
gethysteresis [ <addr> [ [ lun: ]<sensor id> | <sensor name> ] ]
getipmbstate <addr> [<link>]
getlanconfig <channel number> <parameter number> | <parameter name>
getmgmtportroute [slot]
getpefconfig <parameter name> | <parameter number> [<set selector>]
getsensoreventenable [ <addr> [ [ lun: ]<sensor_id> | <sensor name>  ] ]
getthreshold [ <addr> [ [ lun: ]<sensor id> | <sensor name> ] ]
help [<command>]
ipmc [<addr>]
localaddress
mgmtportstate <slot>
minfanlevel [<min fan level>]
networkelementid [<id>]
poll
quit
sel [clear] [ <addr> [ <number of items> [<number of first item>] ] ]
sel info [<addr>]
sendamc <addr> <amc> <netfn> <command> [<parameters ...>]
sendcmd <addr> <netfn> <command> [<parameters ...>]
sensor [ <addr> [ [ lun: ]<sensor id> | <sensor name> ] ]
sensordata [ <addr> [ [ lun: ]<sensor id> | <sensor name> ] ]
sensorread <addr> [ lun: ]<sensor id>
session
setbootdev <addr> <fru-id | amc-addr> <boot-device>
setextracted <addr> <fru_id>
setfanlevel <addr> <fru_id> <state>
setfanpolicy <addr> <fru_id> <ENABLE|DISABLE [timeout]>
                   [-s <addr>|site_type <fru_id>|site_number]
setipmbstate <addr> A|B [<link>] 0|1
setlanconfig <channel number> <parameter number> | parameter name
                  <parameters ...>
setlocked <addr> <fru_id> <value>
setmgmtportroute <slot> < 1 / 0 >
setpefconfig <parameter name> | <parameter number> [<set selector>]
                  <parameters ...>
setsensoreventenable <addr> [ lun: ]<sensor_id> | <sensor name> global
                  [assertion_events [deassertion_events]]
setthreshold <addr> [ lun: ]<sensor_id> | <sensor name> unc | uc | unr
                  | lnc | lc | lnr [-r] value
```

```
  setfruledstate <addr> <fru_id> <LedId>|ALL <LedOp|tail> [LedColor]
  setpowerlevel <addr> <fru_id> [<pwr_lvl>|OFF] [Copy]
  shelf <parameters>
  shelfaddress ["<shelf address>"]
  shmstatus
  showunhealthy
  switchover
  terminate [-reboot]
  threshold [ <addr> [ [ lun: ]<sensor id> | <sensor name> ] ]
  user [<user id>]
  user add <user id> <user name> <flags> <privilege level> <password>
  user channel <user id> <channel number> <flags> <privilege level>
  user delete <user id>
  user delete <user id>
  user enable <user id> 1|0
  user name <user id> <user name
  user passwd <user id> <user password>
  version
  console [slot_number]
  userlabel [ shelf | slot ] <slot number>
  setuserlabel [ shelf | slot ] [ <shelf name> | <slot number> ] <slot name>
  flashupdate <slot number> -s <server IP address> -f <path name>
  showhost <slot number> [version]
  amcportstate [-v] <ipmc> <fru_id>
```

Get help for a specific command.

```
# clia help shelf pwrreorder
Pigeon Point Shelf Manager Command Line Interpreter

        Set the Power Order
        PwrReorder <addr1> <fru_id1> before/after <addr2> <fru_id2>
#
```

ipmc

Syntax:

```
ipmc [-v] [IPMB-address]
ipmc board n
ipmc fan_tray n
```

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

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

- Information returned by the `Get Device ID` IPMI command, including manufacturer ID, product ID, device ID, device firmware revision, and supported IPMI version
- Device ID string from the controller SDR
- Power state notification attribute from the controller SDR, as a hexadecimal number

- Global initialization attribute from the controller SDR, as a hexadecimal number
- Device capabilities attribute from the controller SDR, as a hexadecimal number
- Whether the controller provides Device SDRs
- Supported features mask, with a textual explanation of each bit
- The list of ports subject to E-Keying, with their states (Enabled/Disabled)

Examples:

Get information about the IPM controller at address 9C.

```
# clia ipmc 9c
Pigeon Point Shelf Manager Command Line Interpreter

9c: Entity: (0xd0, 0x0) Maximum FRU device ID: 0x08
    PICMG Version 2.0
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
#
```

Get verbose information about the IPM controller at address 9C.

```
# clia ipmc -v 9c
Pigeon Point Shelf Manager Command Line Interpreter

9c: Entity: (0xd0, 0x0) Maximum FRU device ID: 0x08
    PICMG Version 2.0
    Hot Swap State: M4 (Active), Previous: M3 (Activation In
Process), Last State Change Cause: Normal State Change (0x0)
    Device ID: 0x00, Revision: 0, Firmware: 1.01, IPMI ver 1.5
    Manufacturer ID: 00315a (PICMG), Product ID: 0000, Auxiliary
Rev: 01ac10ac
    Device ID String: "PPS Sentry 6"
    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"
#
```

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). These addresses will be different between redundant Shelf Managers (while the BMC address is shared between them).

Example:

```
# clia localaddress
Pigeon Point Shelf Manager Command Line Interpreter

Local IPMB Address = 0xFC
#
```

mgmtportstate

Syntax:

`mgmtportstate` [*slot-number*]

Purpose:

This command is useful for determining the current port configuration and determining if there is a need to change it. This command employs the IPMC OEM command `get ethernet port access` to query the IPMC for management port state or port routing configuration.

- For setting management route configuration, see "`setmgmtportroute`" on page 284 command.
- For configuring management port (front or rear access), see "Set Management Port" on page 363.

If you are using a third-party chassis instead of a Sun Netra CT900 server chassis, modify your system manager or Shelf manager software to send the `set ethernet port access` OEM command to IPMC before starting power budget negotiations and applying power. The command has to be sent after M2 but before M4 state.

Example:

Display the management port state for the slot.

```
# clia mgmtportstate 3
Pigeon Point Shelf Manager Command Line Interpreter

Current slot MGMT port Routing Information
------------------------------------------
Slot   MGMT
------------------------------------------
 3     Rear (ARTM)
```

minfanlevel

Syntax:

`minfanlevel [`*level*`]`

Purpose:

This command shows or sets the minimum fan level. Under normal conditions, the cooling management algorithm gradually decreases the level for the fans in the system while thermal conditions stay normal. However the cooling management algorithm would not try to decrease the fan level below the minimum level specified by the configuration parameter `MIN_FAN_LEVEL`, or by this command.

The default value for the minimum fan level is 1. Setting the minimum fan level to a higher value does not prevent the fan level from being set below that value via the command `clia setfanlevel` or via the ATCA command `SetFanLevel` submitted over RMCP. The minimum fan level affects only the automatic management of the fan level by the cooling management facility.

This command without parameters shows the current minimum fan level.

This command with an integer parameter sets the minimum fan level to the value of the parameter.

Example:

```
# clia minfanlevel 3
Pigeon Point Shelf Manager Command Line Interpreter

Minimal Fan Level is set to 3

# clia minfanlevel
Pigeon Point Shelf Manager Command Line Interpreter

Minimal Fan Level is 3
#
```

muxstate

Syntax:

`muxstate` *slot-number*

Purpose:

This command queries the IPMC for the current multiplexer (MUX) state and port routing information for the blade in the specified slot. This command is only valid on a host that is runing the Oracle Solaris OS.

The Shelf Manager provides the slot's MUX configuration information stored in the shelf's midplane FRU to the IPMC so that the IPMC can program the host MUX configuration prior to blade activation.

Note – The ShMM MUX configuration must match the Oracle Solaris host's NIU driver configuration or the configuration can fail.

The `setmuxconfig` command is used to store the MUX configuration on the ShMM and the `getmuxconfig` command is used to show the stored Mux configuration. See `getmuxconfig` and `setmuxconfig` commands for more information.

Example:

Display the port routing information for the MUX on the node board in physical slot 4.

```
# clia muxstate 4
Pigeon Point Shelf Manager Command Line Interpreter

Current slot 10GbE NIU-XAUI Routing Information
------------------------------------------------
Slot   XAUI-1           XAUI-2
--------------------------------------------
 4     Zone3 (ARTM)    Zone2 (Back Plane)
```

networkelementid

Syntax:

`networkelementid ["`*id*`"]`

Purpose:

This command is carrier-specific and is not necessarily supported on all ShMM carriers.

This command shows or sets the Network Element Identifier if this parameter is supported by the current carrier. The superuser (UID 0) privilege is required for setting the Network Element Identifier.

The Network Element Identifier specified as the command line parameter id must be in the format defined by the specific carrier.

If no parameter is specified in the command line, the current Network Element Identifier is displayed.

Examples:

```
# clia networkelementid
Pigeon Point Shelf Manager Command Line Interpreter

Network Element ID: "0123456789A"
#
# clia networkelementid "01234567890"
Pigeon Point Shelf Manager Command Line Interpreter

Network Element ID is set successfully to "01234567890"
#
```

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.

This command is mostly useful in PICMG 2.x shelves, where Hot Swap state machine support for IPM controllers is optional and a new IPM controller on IPMB may not be immediately recognized by the Shelf Manager. The command `poll` causes the Shelf Manager to recognize new IPM controllers.

In AdvancedTCA shelves, this command is not necessary, because a new IPM controller is recognized by the Shelf Manager automatically when it sends its first Hot-Swap event. Nevertheless, this command can be used in AdvancedTCA shelves if an IPMB-0 population rediscovery cycle is needed.

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*]

IPMB-address can be replaced by the `board` *n* or `shm` *n* abbreviations

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* can be specified that 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. Both *record-count* and *starting-entry* must be within the range from 1 to the total number of records in the SEL. The default value of the optional parameter *starting-entry* is 1. The *starting-entry* is independent of the `RecordID` field of the SEL record

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/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.

Examples:

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: 43
    Free space: 15680 bytes
    Last addition timestamp: Nov 19 17:12:47 2003
    Last erase timestamp: Oct 31 23:59:59 2003
    Supported operations: 0x0f

# clia sel 20 5
Pigeon Point Shelf Manager Command Line Interpreter

0x0027: Event: at Nov 19 17:12:42 2003; from:(0x9c,0,0);
sensor:(0xf0,0); event:0x6f(asserted): HotSwap: FRU 0 M4->M6,
Cause=0x1
0x0028: Event: at Nov 19 17:12:42 2003; from:(0x9c,0,0);
sensor:(0xf0,0); event:0x6f(asserted): HotSwap: FRU 0 M6->M1,
Cause=0x0
0x0029: Event: at Nov 19 17:12:46 2003; from:(0x9c,0,0);
sensor:(0xf0,0); event:0x6f(asserted): HotSwap: FRU 0 M1->M2,
Cause=0x2
0x006F or 2A (Sun legacy): Event: at Nov 19 17:12:46 2003;
from:(0x9c,0,0); sensor:(0xf0,0); event:0x6f(asserted): HotSwap:
FRU 0 M2->M3, Cause=0x1
0x002B: Event: at Nov 19 17:12:47 2003; from:(0x9c,0,0);
sensor:(0xf0,0); event:0x6f(asserted): HotSwap: FRU 0 M3->M4,
Cause=0x0
```

Get five SEL entries from entry # 15 (0x0f).

```
# clia sel 20 5 15
Pigeon Point Shelf Manager Command Line Interpreter

0x000F: Event: at Nov 19 16:49:21 2003; from:(0x20,0,0);
sensor:(0xf0,3); event:0x6f(asserted): HotSwap: FRU 2 M2->M3,
Cause=0x1
0x0010: Event: at Nov 19 16:49:22 2003; from:(0x20,0,0);
sensor:(0xf0,2); event:0x6f(asserted): HotSwap: FRU 1 M2->M3,
Cause=0x1
```

```
0x0011: Event: at Nov 19 16:49:22 2003; from:(0x20,0,0);
sensor:(0xf0,2); event:0x6f(asserted): HotSwap: FRU 1 M3->M4,
Cause=0x0
0x0012: Event: at Nov 19 16:49:22 2003; from:(0xfc,0,0);
sensor:(0xf0,0); event:0x6f(asserted): HotSwap: FRU 0 M3->M4,
Cause=0x0
0x0013: Event: at Nov 19 16:49:22 2003; from:(0x20,0,0);
sensor:(0xf0,3); event:0x6f(asserted): HotSwap: FRU 2 M3->M4,
Cause=0x0
#
```

Clear the SEL.

```
# clia sel clear
Pigeon Point Shelf Manager Command Line Interpreter

SEL clear: issued successfully
    SEL clearing completed
# clia sel
Pigeon Point Shelf Manager Command Line Interpreter

SEL is empty
#
```

sendamc

Syntax:

`sendamc` *addr amc-address netfn command-code* `[`*parameter*`1 ...` *parameterN*`]`

Purpose:

This command allows the user to send an IPMI command to an Advanced Management Controller (AMC) that resides behind its correspondent IPM controller in a transparent way. All the parameters of this command are hexadecimal numbers in the range 00h - FFh. The prefix "0x" is not required. The target controller is specified by the *AMC-address* parameter. If it is greater then 70h, this is the actual AMC address on IPMB-L. If it is less then 70h, it is the FRU device ID that represents the corresponding AMC. The NetFn code of the command is specified by the *netfn* parameter. The code of the command is specified by the *command-code* parameter. The request data bytes of the command are represented by *parameter1*, *parameter2*, and so on.

The command reports the completion code of the IPMI command and the response data are displayed as hexadecimal bytes.

Examples:

Send the `Get Device ID` command to the AMC (IPMB address 84h, FRU ID 1). The NetFn of the command is 06h, the code of the command is 01h. Since this command does not require request data, no *parameter1*, *parameter2*, ... are specified.

```
# clia sendamc 84 1 6 1
Pigeon Point Shelf Manager Command Line Interpreter

Completion code: 0x0 (0)
Response data: 34 80 01 20 51 29 0A 40 00 EF BE
#
```

Send the "`Get Device ID` command to the AMC (IPMB address 84h, AMC address 72h). The NetFn of the command is 06h, the code of the command is 01h. Since this command does not require request data, no *parameter1*, *parameter2*, ... are specified.

```
# clia sendamc 84 72 6 1
Pigeon Point Shelf Manager Command Line Interpreter

Completion code: 0x0 (0)
Response data: 34 80 01 20 51 29 0A 40 00 EF BE
#
```

sendcmd

Syntax:

`sendcmd` *IPMB-address netfn command-code* [*parameter*1 ... *parameterN*]

Purpose:

This command allows the user to send an IPMI command to an IPM controller in a transparent way. All the parameters of this command are hexadecimal numbers in the range 00h - FFh. The prefix "0x" is not required. The target controller is specified by the *IPMB-address* parameter. The NetFn code of the command is specified by the *netfn* parameter. The code of the command is specified by the *command-code* parameter. The request data bytes of the command are represented by *parameter1*, *parameter2*, and so on.

The command reports the completion code of the IPMI command and the response data are displayed as hexadecimal bytes.

Examples:

Send the `Get Device ID` command to the Shelf Manager (IPMB address 20h). The NetFn of the command is 06h, the code of the command is 01h. Since this command does not require request data, no *parameter1*, *parameter2*, ... are specified.

```
# clia sendamc 84 1 6 1
Pigeon Point Shelf Manager Command Line Interpreter

Completion code: 0x0 (0)
Response data: 00 80 02 30 51 BF 0A 40 00 00 00
#
```

sensor

Syntax:

`sensor [-v] [`*IPMB-address* `[`*sensor-name* `|` `[`*lun*`:]`*sensor-number*`]]`
`sensor [-v] board` *n* `[`*sensor-name* `|` `[`*lun*`:]`*sensor-number*`]]`
`sensor [-v] shm` *n* `[`*sensor-name* `|` `[`*lun*`:]`*sensor-number*`]]`

Purpose:

This command shows information about specific sensors. The target sensor is selected by its IPM controller's IPMB address and by sensor number or by sensor name (device ID string from the sensor SDR, enclosed in double quotes). If neither sensor name nor sensor number is specified, information about all sensors on the specified IPM controller is shown. If no parameters are specified, information about all known sensors is shown.

This command allows the user to qualify the sensor number with the Logical Unit Number (LUN) if the target controller supports sensors on multiple LUNs. If the LUN is omitted, information about sensors with the specified sensor number on all LUNs is shown. *lun* can take the value 0, 1, or 3 (LUN 2 is reserved.)

Sensor names are not qualified with LUN numbers, since it is assumed that sensor names will normally be unique within the controller. However, if there are several sensors with the same name within the controller, information is shown about all of them.

The following information is shown for each sensor in standard mode:

- 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 Entity ID, Entity Instance of the related entity (the FRU device ID if the sensor is associated with a FRU)

The following information is shown for the sensor in verbose mode only (refer to the IPMI specification for information about these attributes):

- Assertion mask
- Deassertion mask
- Settable/readable mask for sensor states (in the case of a discrete sensor) or thresholds (in the case of a threshold-based sensor)

The following information is shown in verbose mode for threshold-based sensors only:

- Sensor units: base and modified
- Unit percentage, modifier, and rate
- Analog format and flags
- Linearization parameters, M, B, K1, K2 coefficients
- Tolerance and accuracy coefficients
- Nominal, normal maximum, normal minimum, maximum, and minimum values
- Upper thresholds: non-critical, critical, and non-recoverable
- Lower thresholds: non-critical, critical, and non-recoverable
- Hysteresis values: positive and negative

Examples:

Get standard information about sensor `FAN TRAY 0` on IPM controller 20.

```
# clia sensor 20 "Fan Tray 0"
Pigeon Point Shelf Manager Command Line Interpreter

20: LUN: 0, Sensor # 12 ("FAN Tray 0")
    Type: Discrete (0x6f), "Entity Presence" (0x25)
    Belongs to entity: (0x1e, 96) [FRU # 3]
#
```

Get verbose information about sensor 5 on IPM controller 8a.

```
# clia sensor -v 8a 5
Pigeon Point Shelf Manager Command Line Interpreter

8a: LUN: 0, Sensor # 5 ("Board Temp")
    Type: Threshold (0x01), "Temperature" (0x01)
    Belongs to entity: (0x3, 96) [FRU # 0]
    Assertion Mask: 0x0a80
        Upper Non-Critical Going High
        Upper Critical Going High
        Upper Non-Recoverable Going High
    Deassertion Mask: 0x7a80
        Upper Non-Critical Going High
        Upper Critical Going High
        Upper Non-Recoverable Going High
    Settable / Readable Mask: 0x3838
        Upper Non-Critical Threshold is Comparison Returned
        Upper Critical Threshold is Comparison Returned
        Upper Non-Recoverable Threshold Comparison is Returned
        Upper Non-Critical Threshold is Readable
        Upper Critical Threshold is Readable
        Upper Non-Recoverable Threshold is Readable
        Upper Non-Critical Threshold is Settable
        Upper Critical Threshold is Settable
        Upper Non-Recoverable Threshold is Settable
    Unit Percentage: OFF (0), Unit Modifier: none (0), Unit Rate:
none (0)
    Analog Format: 2's complement (signed) (2)
    Base Unit: degrees C (1), Modifier Unit: unspecified (0)
    Linearization: linear (0), M = 100, B = 0, K1 = 0, K2 = -2
    Tolerance = 0, Accuracy = 0, Accuracy EXP = 0
    Analog Flags: 0x0
    Nominal: 60 (0x3c), Normal max: 100 (0x64), Normal min: -40
(0xd8)
    Sensor max: 120 (0x78), Sensor min: -40 (0xd8)
    Upper Thresholds:
      Non-Critical: 48 (0x30) Critical: 60 (0x3c) Non-Recoverable:
68 (0x44)
    Lower Thresholds:
        N/A
    Hysteresis:
        Positive: 0 (0x00), Negative 0 (0x00)
#
```

Same as above, but explicitly specifying the LUN for the sensor.

```
# clia sensor -v 8a 0:5
Pigeon Point Shelf Manager Command Line Interpreter

8a: LUN: 0, Sensor # 5 ("Board Temp")
    Type: Threshold (0x01), "Temperature" (0x01)
    Belongs to entity: (0x3, 96) [FRU # 0]
    Assertion Mask: 0x0a80
        Upper Non-Critical Going High
        Upper Critical Going High
        Upper Non-Recoverable Going High
    Deassertion Mask: 0x7a80
        Upper Non-Critical Going High
        Upper Critical Going High
        Upper Non-Recoverable Going High
    Settable / Readable Mask: 0x3838
        Upper Non-Critical Threshold is Comparison Returned
        Upper Critical Threshold is Comparison Returned
        Upper Non-Recoverable Threshold Comparison is Returned
        Upper Non-Critical Threshold is Readable
        Upper Critical Threshold is Readable
        Upper Non-Recoverable Threshold is Readable
        Upper Non-Critical Threshold is Settable
        Upper Critical Threshold is Settable
        Upper Non-Recoverable Threshold is Settable
    Unit Percentage: OFF (0), Unit Modifier: none (0), Unit Rate:
none (0)
    Analog Format: 2's complement (signed) (2)
    Base Unit: degrees C (1), Modifier Unit: unspecified (0)
    Linearization: linear (0), M = 100, B = 0, K1 = 0, K2 = -2
    Tolerance = 0, Accuracy = 0, Accuracy EXP = 0
    Analog Flags: 0x0
    Nominal: 60 (0x3c), Normal max: 100 (0x64), Normal min: -40
(0xd8)
    Sensor max: 120 (0x78), Sensor min: -40 (0xd8)
    Upper Thresholds:
      Non-Critical: 48 (0x30) Critical: 60 (0x3c) Non-Recoverable:
68 (0x44)
    Lower Thresholds:
        N/A
    Hysteresis:
        Positive: 0 (0x00), Negative 0 (0x00)
#
```

sensordata

Syntax:

`sensordata [`*IPMB-address* `[`*sensor-name* `|` `[`*lun*`:]`*sensor-number*`]]`
`sensordata [-v] board` *n* `[`*sensor-name* `|` `[`*lun*`:]`*sensor-number*`]]`
`sensordata [-v] shm` *n* `[`*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 target sensor is selected by its IPM controller's IPMB address and by sensor number or by sensor name (device ID string from the sensor SDR, enclosed in double quotes). If neither sensor name nor sensor number is specified, values of all sensors on the specified IPM controller are shown. If no parameters are specified, values of all known sensors are shown.

This command allows the user to qualify the sensor number with the Logical Unit Number (LUN) if the target controller supports sensors on multiple LUNs. If the LUN is omitted, information about sensors with the specified sensor number on all LUNs is shown. *lun* can take the value 0, 1, or 3 (LUN 2 is reserved.)

Sensor names are not qualified with LUN numbers, since it is assumed that sensor names will normally be unique within the controller. However, if there are several sensors with the same name within the controller, information is shown about all of them.

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

The value/asserted states are shown both in raw and processed form. In processed form, the analog value are converted according to M, B, and R and shown together with the unit name (for example, 27 degrees). The discrete value is annotated according to the event/reading code type (for example, for the event/reading code 2, the asserted state 0 is shown as `Transition to Idle`).

Examples:

Get sensor data values for a temperature sensor `Local Temp` on IPM controller FE.

```
# clia sensordata FE "Local Temp"
Pigeon Point Shelf Manager Command Line Interpreter

fe: LUN: 0, Sensor # 3 ("Local Temp")
    Type: Threshold (0x01), "Temperature" (0x01)
    Status: 0xc0
        All event messages enabled from this sensor
        Sensor scanning enabled
        Initial update completed
    Raw data: 22 (0x16)
    Processed data: 22.000000 degrees C
    Status: 0x00
```

Get sensor data values for a discrete (Hot Swap) sensor (#0) on IPM controller 9C.

```
# clia sensordata 9c 0
Pigeon Point Shelf Manager Command Line Interpreter

9c: LUN: 0, Sensor # 0 ("FRU 0 HOT_SWAP")
    Type: Discrete (0x6f), "Hot Swap" (0xf0)
    Status: 0xc0
        All event messages enabled from this sensor
        Sensor scanning enabled
        Initial update completed
    Sensor reading: 0x00
    Current State Mask 0x0010
```

Get sensor data values for the same sensor, but qualifying it explicitly with the LUN.

```
# clia sensordata 9c 0:0
Pigeon Point Shelf Manager Command Line Interpreter

9c: LUN: 0, Sensor # 0 ("FRU 0 HOT_SWAP")
    Type: Discrete (0x6f), "Hot Swap" (0xf0)
    Status: 0xc0
        All event messages enabled from this sensor
        Sensor scanning enabled
        Initial update completed
    Sensor reading: 0x00
    Current State Mask 0x0010
```

sensorread

Syntax:

`sensorread` *IPMB-address* `[`*lun*`:]`*sensor-number*

Purpose:

This command shows the raw value of the specified sensor. The only difference between the commands `sensorread` and `sensordata` is that the command `sensorread` does not check the presence of the target IPM controller or the validity of the sensor number, but just sends the `Get Sensor Reading` request directly via IPMB. This command does not retrieve the SDR of the sensor and thus it can not process the obtained data.

This command allows the user to qualify the sensor number with the Logical Unit Number (LUN) if the target controller supports sensors on multiple LUNs. If the LUN is omitted, LUN 0 is used. *lun* can take values 0, 1, or 3. (LUN 2 is reserved.)

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

Examples:

Get sensor data values for sensor 4 on IPM controller FC. Notice that the `sensorread` command provides only unprocessed sensor values. Also notice the command example with an explicit LUN.

```
# clia sensordata fc 4
Pigeon Point Shelf Manager Command Line Interpreter

fc: LUN: 0, Sensor # 4 ("3.3STBY voltage")
    Type: Threshold (0x01), "Voltage" (0x02)
    Status: 0xc0
        All event messages enabled from this sensor
        Sensor scanning enabled
        Initial update completed
    Raw data: 193 (0xc1)
```

```
    Processed data: 3.396800 Volts
    Status: 0x00

# clia sensorread fc 4
Pigeon Point Shelf Manager Command Line Interpreter

fc: LUN: 0, Sensor # 4
    Raw data: 193 (0xc1)
    Status: 0xc0
        All event messages enabled from this sensor
        Sensor scanning enabled
        Initial update completed
    Threshold Sensor Status: 0x00
    Discrete Sensor Current State Mask 0x0000

# clia sensorread fc 0:4
Pigeon Point Shelf Manager Command Line Interpreter

fc: LUN: 0, Sensor # 4
    Raw data: 193 (0xc1)
    Status: 0xc0
        All event messages enabled from this sensor
        Sensor scanning enabled
        Initial update completed
    Threshold Sensor Status: 0x00
    Discrete Sensor Current State Mask 0x0000
#
```

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
#
```

setacousticlevel

Syntax:

`setacousticlevel` *ETSI NEBS-A NEBS-U*

Purpose:

This command sets the system acoustic level. This command is available for use with the server chassis and is only applicable only if you are using the SUNCT900 cooling algorithm. This command allows users to set the default fan speed to various acoustic levels based on custom configuration requirements. When you set an acoustic level, the variable will be changed and the default cooling algorithm will set the fan speeds accordingly. This change will take effect dynamically, is persistant, and no ShMM reboot is required. See also `getacousticlevel`.

Example:

Set the system acoustic level and fan speed.

```
# clia setacousticlevel NEBS-A
Pigeon Point Shelf Manager Command Line Interpreter

Changed the system acoustic level to 'NEBS-A'
Fan speed changed to '7'.
```

setbootdev

Syntax:

`getbootdev` *IPMB-0-address FRU-device-ID* | *IPMB-L-address boot-device*

Purpose:

This command sets the system boot parameter for a designated IPM controller. The second parameter of the command should be set to 0 if the AdvancedMC access is not targeted. If the second parameter exceeds 70h it is treated as an IPMB-L address for an AMC address. Otherwise, the second parameter is treated as a FRU ID and converted to an IPMB-L address via AMC address mapping.

The *boot-device* parameter can be:

- `1` or `pxe` (Pre-Boot Execution Environment)
- `2` or `disk` (Default Hard Drive)
- `3` or `safe` (Default Hard Drive, Safe Mode)
- `4` or `diag` (Default Diagnostic Partition)
- `5` or `cd` (Default CD/DVD)
- `14` or `bios` (BIOS)
- `15` or `floppy` (Floppy/Primary Removable Media).

Example:

Get the system boot options for IPM controller at IPMB address 82h as `pxe` (Pre-Boot Execution Environment).

```
# clia setbootdev 82 0  1
Pigeon Point Shelf Manager Command Line Interpreter

Set boot device option: status = 0x0 (0)
Boot device set to 1 (Force PXE)
Response data (raw): A2
#
```

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. If the specified FRU is in state M7, the Shelf Manager places it in state M0 (FRU physically absent).

Example:

```
# clia setextracted 9c 0
Pigeon Point Shelf Manager Command Line Interpreter

Set FRU extracted state successfully
#
```

setfanlevel

Syntax:

`setfanlevel` *IPMB-address fru-id level*
`setfanlevel fan_tray` *n level*
`setfanlevel all` *level*

Purpose:

This command sets the new level for the fan controlled by the FRU specified in the command parameters. The minimum is 1 and maximum is 15.

The version of this command with an `all` qualifier attempts to set the same level for all known fans in the shelf.

Examples:

Set fan level for the fan residing at FRU #2 at IPMB address 0x20 to 5.

```
# clia setfanlevel 20 2 5
Pigeon Point Shelf Manager Command Line Interpreter

20: FRU # 2 Set Fan Level to: 5
#
```

Set fan level to 4 for all known fans in the shelf:

```
# clia setfanlevel all 4
Pigeon Point Shelf Manager Command Line Interpreter

72: FRU # 0 Set Fan Level to: 4
76: FRU # 0 Set Fan Level to: 4
#
```

setfanpolicy

Syntax:

`setfanpolicy` *fan-tray-addr* *fan-tray-fru_id* `ENABLE | DISABLE [timeout]` `[-s` *addr*|*site_type* `[`*fru_id* | *site_number*`]`

Purpose:

This command enables or disables fan trays for cooling management in addition to the Fan Geography record if this one is presented in the ShelfFRU.

The parameters *fan-tray-addr* and *fan-tray-fru_id* specify a fan tray. If the `DISABLE` policy for the fan tray is specified, the additional parameter *timeout* may be used to specify the duration of the policy. The *timeout* parameter is treated in seconds and later is rounded by 5 seconds units according to the PICMG 3.0 specification. The value of the *timeout* parameter may not be greater then 21 minutes (1260 seconds), the minimum value of *timeout* is 5 seconds. If the *timeout* variable is not specified then the `DISABLE` policy is assumed to be infinite.

The flag `-s` precedes the parameters that define a site covered by the fan tray.

The *site_type* parameter can accept one of the following values: `Board`, `PEM`, `ShelfFRU`, `ShelfManager`, `FanTray`, `FanFilterTray`, `Alarm`, `Mezzanine`, `PMC`, `RTM`.

If a numeric argument is expected to be treated as a hexadecimal, the "0x" prefix should be used, otherwise the error will be returned.

Examples:

Disable fan policy for 60 seconds. The fan tray is at IPMB address 20h, FRU ID 3. The site covered by the fan tray is at IPMB address 12h, FRU ID 0.

```
# clia setfanpolicy 0x20 3 DISABLE 60  -s 0x12 0
Pigeon Point Shelf Manager Command Line Interpreter

 Fan policy updated successfully
#
```

Disable fan policy for infinite period of time. The fan tray is at IPMB address 20h, FRU ID 3. The site covered by the fan tray is defined by Site Type "PICMG Board" and Site Number 7.

```
# clia setfanpolicy 0x20 3 DISABLE  -s board 7
Pigeon Point Shelf Manager Command Line Interpreter

 Fan policy updated successfully
#
```

Enable fan policy for the fan tray at IPMB address 20h, FRU ID 3, and for all sites covered by this fan.

```
# clia setfanpolicy 0x20 3 ENABLE
Pigeon Point Shelf Manager Command Line Interpreter

 Fan policy updated 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 allows the user 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 – turn on the LED
- OFF – turn off the LED
- LOCAL – revert to the local control of the LED
- BLINK – cause the LED to blink, repeatedly turning it on for *onTime* milliseconds and then turning it off for *offTime* milliseconds
- TEST – run 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 10–2500 ms range.

The optional parameter *LedColor* designates a color, either via a symbolic name or a decimal value. Symbolic names of colors correspond to decimal values in accordance with the PICMG 3.0 specification, as listed below. (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)

Examples:

Turn off LED #1 of FRU #0 of IPM controller at IPMB address 20h.

```
# clia setfruledstate  20 0 1 OFF
Pigeon Point Shelf Manager Command Line Interpreter

    Setting FRU’s led state completed successfully, status = 0x0
#
```

Enable local control for LED #1 of FRU #0 of IPM controller at IPMB address 20h.

```
# clia setfruledstate  20 0 1 LOCAL
Pigeon Point Shelf Manager Command Line Interpreter

    Setting FRU’s led state completed successfully, status = 0x0
#
```

Enable blinking on LED #1 of FRU #0 of IPM controller at IPMB address 20h. The blinking is in the default color. The on duration is 100 ms and the off duration is 200 ms.

```
# clia setfruledstate 20 0 0 BLINK 100 200
Pigeon Point Shelf Manager Command Line Interpreter

    Setting FRU’s led state completed successfully, status = 0x0
#
```

sethysteresis

Syntax:

`sethysteresis` *IPMB-address* `[`*lun*`:]` *sensor_id* | *sensor_name* `pos` | `neg` `[-r]` *value*

Purpose:

This command sets the value for the specified hysteresis for the specified sensor. The sensor must be a threshold-based sensor. It must support the designated threshold hysteresis and the hysteresis must be settable.

The command allows the user to qualify the sensor number with the Logical Unit Number (LUN) if the target controller supports sensors on multiple LUNs. The command sets the positive hysteresis if the `pos` argument is present and sets the negative hysteresis if the `neg` argument is present.

Examples:

Set positive hysteresis for sensor #2 of the IPM controller at IPMB address 0xFC.

```
# clia sethysteresis FC 2 pos 10
Pigeon Point Shelf Manager Command Line Interpreter

     Positive hysteresis set successfully to 0xA, previous: 0x0
#
```

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.

The command works differently in bused and radial environments. In a bused environment, and in 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.

Examples:

Disable IPMB-A link on the IPM controller at IPMB address 92h.

```
# clia setipmbstate 92 A 0
Pigeon Point Shelf Manager Command Line Interpreter

     Command executed successfully
```

Enable radial IPMB link 3, bus B on the Shelf Manager (which is the IPMB hub).

```
# clia setipmbstate 20 B 3 1
Pigeon Point Shelf Manager Command Line Interpreter

     Command executed successfully
```

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. The channel number, the configuration parameter name or number, and the parameter value should be explicitly specified.

TABLE A-3 lists names and numbers of LAN configuration parameters supported by the setlanconfig command:

TABLE A-3 LAN Configuration Parameters for setlanconfig

Parameter Name	Number	Description
auth_enables	2	Five 8-bit values that contain authentication types 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.

TABLE A-3 LAN Configuration Parameters for `setlanconfig` *(Continued)*

`backup_gw_ip`	14	A string value that contains the IP address of the backup gateway in dotted decimal notation.
`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)

auth_enables

Syntax:

`setlanconfig` *channel* `auth_enables` *value1 value2 value3 value4 value5*
`setlanconfig` *channel* `2` *value1 value2 value3 value4 value5*

Purpose:

This command sets the current value of the LAN parameter `auth_enables`. This parameter specifies which authentication types are currently enabled by the Shelf Manager for each of five supported privilege levels (Callback, User, Administrator, Operator, and OEM) and is represented by a sequence of five bytes, each corresponding to the respective privilege level, treated as a bit mask with the bits defined as follows:

- 0x01 None
- 0x02 MD2

- 0x04 MD
- 0x10 Straight password/key
- 0x20 OEM proprietary

Parameters *value1* to *value5* should represent the values of these bytes, in hexadecimal. The Shelf Manager does not currently support callback and OEM privilege levels. Therefore, the *value1* and *value5* parameters corresponding to these privilege levels should be specified as 0.

Example:

```
# clia setlanconfig 1 auth_enables 0 1 1 1 0
Pigeon Point Shelf Manager Command Line Interpreter

Authentication Type Enables set successfully
#
```

ip

Syntax:

```
setlanconfig channel ip value
setlanconfig channel 3 value
```

Purpose:

This command sets the current IP address used by the channel. The value should represent an IP address in dotted decimal notation.

Example:

```
# clia setlanconfig 1 ip 172.16.2.203
Pigeon Point Shelf Manager Command Line Interpreter

IP Address set successfully
#
```

subnet_mask

Syntax:

`setlanconfig` *channel* `subnet_mask` *value*
`setlanconfig` *channel* `6` *value*

Purpose:

This command sets the current IP subnet mask used by the channel. The value should represent a subnet mask in dotted decimal notation.

Example:

```
# clia setlanconfig 1 subnet_mask 255.255.255.0
Pigeon Point Shelf Manager Command Line Interpreter

Subnet Mask set successfully
#
```

ipv4_hdr_param

Syntax:

setlanconfig *channel* ipv4_hdr_param *value1 value2 value3*
setlanconfig *channel* 7 *value1 value2 value3*

Purpose:

This command sets the IP 4 header parameters for the Shelf Manager. They are represented as three single-byte values in hexadecimal notation: *value1*, *value2,* and *value3*. The content of the bytes conforms to section 19.2 of the IPMI 1.5 specification and contains the following attributes:

- Time-to-live in byte 1
- IP header flags (bits [7:5]) in byte 2
- Precedence (bits [7:5]) and type of service (bits [4:1]) in byte 3

Example:

```
# clia setlanconfig 1 ipv4_hdr_param 37 E0 11
Pigeon Point Shelf Manager Command Line Interpreter

IPv4 Header Parameters set successfully
#
```

arp_control

Syntax:

setlanconfig *channel* arp_control *value*
setlanconfig *channel* 10 *value*

Purpose:

This command sets the current value of the LAN parameter `arp_control`. This parameter specifies additional ARP support provided by the Shelf Manager, and is represented by a single byte, treated as a bit mask with the bits defined as follows:

- 0x01 Enable Shelf Manager-generated Gratuitous ARPs
- 0x02 Enable Shelf Manager-generated ARP responses

Other bits are reserved and should be set to 0.

Example:

```
# clia setlanconfig 1 arp_control 3
Pigeon Point Shelf Manager Command Line Interpreter

BMC-generated ARP control set successfully
#
```

arp_interval

Syntax:

setlanconfig *channel* arp_interval *value*
setlanconfig *channel* 11 *value*

Purpose:

This command sets the current ARP interval used by the channel. The value should represent the number of seconds/milliseconds in fixed-point numeric format (with a possible fractional part). Due to the definition of this parameter in IPMI, it is truncated to the largest time interval that is divisible by 500 milliseconds.

Example:

```
# clia setlanconfig 1 arp_interval 3.5
Pigeon Point Shelf Manager Command Line Interpreter

Gratuitous ARP interval set successfully
#
```

dft_gw_ip

Syntax:

setlanconfig *channel* dft_gw_ip *value*
setlanconfig *channel* 12 *value*

Purpose:

This command sets the IP address of the default gateway used by the channel. The value should represent an IP address in dotted decimal notation.

Example:

```
# clia setlanconfig 1 dft_gw_ip 172.16.2.100
Pigeon Point Shelf Manager Command Line Interpreter

Default Gateway Address set successfully
#
```

backup_gw_ip

Syntax:

`setlanconfig` *channel* `backup_gw_ip` *value*
`setlanconfig` *channel* `14` *value*

Purpose:

This command sets the IP address of the backup gateway used by the channel. The value should represent an IP address in dotted decimal notation.

Example:

```
# clia setlanconfig 1 backup_gw_ip 172.16.2.100
Pigeon Point Shelf Manager Command Line Interpreter

Backup Gateway Address set successfully
#
```

community

Syntax:

`setlanconfig` *channel* `community` *value*
`setlanconfig` *channel* `16` *value*

Purpose:

This command sets the community string parameter used in PET traps. The value should be a string enclosed in double quotes.

Example:

```
# clia setlanconfig 1 community "Community"
Pigeon Point Shelf Manager Command Line Interpreter

Community string set successfully
#
```

destination_type

Syntax:

`setlanconfig` *channel* `destination_type` *set-selector value1 value2 value3*
`setlanconfig` *channel* `18` *set-selector value1 value2 value3*

Purpose:

This command sets the element of the destination table with the index equal to *set-selector*. Indexes are 0-based. Selector 0 is used to address the volatile destination. Values *value1*, *value2*, and *value3* supply information about the new destination according to section 19.2 of the IPMI specification. The following information is supplied:

- The alert destination type (PET Trap or OEM destination; whether the alert should be acknowledged)
- Alert acknowledge timeout
- Retry count

Example:

```
# clia setlanconfig 1 destination_type 2 80 3 5
Pigeon Point Shelf Manager Command Line Interpreter

Destination Type set successfully
#
```

destination_address

Syntax:

setlanconfig *channel* destination_address *set-selector gateway-sel IP-address MAC-address*
setlanconfig *channel* 19 *set-selector gateway-sel IP-address MAC-address*

Purpose:

This command sets the element of the destination address table with the index equal to *set-selector*. Indexes are 0-based. Selector 0 is used to address the volatile destination. The command parameters supply the necessary information:

- *gateway-sel* – gateway to use: 0 for default gateway, 1 for backup gateway
- *IP-address* – the destination IP address in dotted-decimal notation
- *MAC-address* – the destination MAC address, six hexadecimal bytes separated by colons

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
#
```

setlocked

Syntax:

setlocked *IPMB-address fru-id* 0 | 1
setlocked board *n* 0 | 1
setlocked shm *n* 0 | 1
setlocked fan_tray *n* 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 FRU is specified using the IPMB address of the owning IPM controller and the FRU device ID. FRU device ID 0 designates the IPM controller proper in PICMG 3.0 contexts.

The Locked bit controls, according to the PICMG 3.0 specification, 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. When the Shelf Manager sends the Deactivate command to the FRU, the FRU transitions to state M1 and sets the Locked bit, preventing subsequent state transitions.

This command can be used to re-activate a previously manually deactivated FRU by clearing the Locked bit for it.

This command can also be issued on the backup Shelf Manager; in that case, the Locked bit can only be set to the specified state for FRUs that are local to the backup Shelf Manager.

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
#
```

setmgmtportroute

Syntax:

`setmgmtportroute` [*slot-number*] [*1/0*]

Purpose:

This command sets the management port route. This information is stored on the midplane as an OEM record.

The configuration is persistent and designated for each slot, regardless of blade presence.

- For current management route configuration, see "`getmgmtportroute`" on page 222 command.
- For querying the IPMC to determine management port state or routing configuration, see "`mgmtportstate`" on page 247 command.
- For configuring management port (front or rear access), see "Set Management Port" on page 363.

Example:

Display the management port routing state for the slot.

```
# clia setmgmtportroute 4 0
Pigeon Point Shelf Manager Command Line Interpreter

MGMT port Routing Information from Midplane FRU
------------------------------------------
Slot   MGMT
------------------------------------------
 4     Rear (ARTM)
```

setmuxconfig

Syntax:

`setmuxconfig` *slot-number* [*port* : *zone*] [*port* : *zone*]

Purpose:

This command updates the multiplexer (MUX) configuration (or port routing) information that is stored in the shelf's midplane OEM multi record for a specified slot. This command is used to ensure that the MUX configuration is persistent after board swap.

The `port` parameter specifies the 10 GbE NIU port and zone parameter specifies where it is routed. The valid *port* values are: `XAUI-1` and `XAUI-2` (or `x1` and `x2` for short). The valid *zone* values are `zone2` and `zone3` (or `z2` and `z3` for short); `zone2` is the back plane and `zone3` is the advance rear transition module (ARTM).

The Shelf Manager provides the slot's MUX configuration information stored in the shelf's midplane FRU to the IPMC so that the IPMC can program the host MUX configuration prior to blade activation. See `getmuxconfig` and `muxstate` commands for more information.

Example:

Set the port routing information for the MUX on the node board in physical slot 5.

```
# clia setmuxconfig 5 x1:z3 xaui2:Zone2
Pigeon Point Shelf Manager Command Line Interpreter

Slot 5 : Successfully  modified the mux routing.
```

setpefconfig

Syntax:

`setpefconfig` *parameter-name additional-parameters*
`setpefconfig` *parameter-number additional-parameters*

Purpose:

This command sets a new value of the specified PEF configuration parameter. TABLE A-4 lists names and numbers of PEF configuration parameters that can be set via this command.

TABLE A-4 PEF configuration parameters for `setpefconf`

Parameter Name	Number	Description
`control`	1	An 8-bit value that represents control flags for PEF (enable PEF, enable PEF startup delay, etc.)
`action_control`	2	An 8-bit value that represents action global control flags for PEF (enable reset, enable power down, etc.)
`startup_delay`	3	Time to delay PEF after system power-ups and resets, in seconds
`alert_startup_delay`	4	Time to delay alerts after system power-ups and resets, in seconds

TABLE A-4 PEF configuration parameters for `setpefconf` *(Continued)*

`event_filter`	6	An event filter table entry identified by the specified set selector. Consists of the following 19 numeric values, in hexadecimal, encoded according to the definition in table 15-2 of the IPMI specification version 1.5: • Filter configuration • Event filter action • Alert policy number • event severity • Generator ID byte 1 • Generator ID byte 2 • Sensor type • Sensor number • Event trigger (event/reading type) • Event data 1 event offset mask • Event data 1 AND mask • Event data 1 compare 1 • Event data 1 compare 2 • Event data 2 AND mask • Event data 2 compare 1 • Event data 2 compare 2 • Event data 3 AND mask • Event data 3 compare 1 • Event data 3 compare 2
`event_filter_data1`	7	The first byte of the event filter table entry identified by the specified set selector
`alert_policy`	9	An alert policy table entry identified by the specified set selector. Consists of the following five numeric values, in hexadecimal, encoded according to the definition in table 15-4 of IPMI 1.5: • Policy number (4-bit value) • Policy (4-bit value); includes the enable/disable bit • Channel number (4-bit value) • Destination selector (4-bit value) • Alert string set/selector
`system-guid`	10	A GUID used to fill in the GUID field in the PET trap
`alert_string_key`	12	An alert string key identified by the specified set selector. Consists of two 8-bit values: event filter number and alert string set.

TABLE A-4 PEF configuration parameters for `setpefconf` *(Continued)*

`alert_string`	13	An alert string identified by the specified set selector.
`oem_filter`	97	An OEM filter table entry identified by the specified set selector. Consists of the following three numeric values: • Byte 1: SEL Record Type Range Low boundary • Byte 2: SEL Record type Range high boundary • Byte 3: Alert policy number that will be invoked for SEL entries that have record types matching the range above.

`control`

Syntax:

```
setpefconfig control value
setpefconfig 1 value
```

Purpose:

This command sets a new value of the PEF parameter `control`. This parameter is a single byte, treated as a bit mask with the bits defined as follows:

- 0x01 – Enable PEF
- 0x02 – Enable generation of event messages for PEF actions
- 0x04 – Enable PEF startup delays on system power-ups and resets
- 0x08 – Enable PEF Alert Startup delays

Other bits are reserved and should be set to 0. The value should be entered in hexadecimal.

Example:

```
# clia setpefconfig control 7
Pigeon Point Shelf Manager Command Line Interpreter

PEF control set successfully
#
```

action_control

Syntax:

`setpefconfig action_control` *value*
`setpefconfig 2` *value*

Purpose:

This command sets a new value of the PEF parameter `action_control`. This parameter is a single byte, treated as a bit mask with the bits defined as follows:

- 0x01 – Enable alert action
- 0x02 – Enable power down action
- 0x04 – Enable reset action
- 0x08 – Enable power cycle action
- 0x10 – Enable OEM action
- 0x20 – Enable diagnostic interrupt

Other bits are reserved and should be set to 0. The value should be entered in hexadecimal

Example:

```
# clia setpefconfig action_control 3f
Pigeon Point Shelf Manager Command Line Interpreter

PEF action control set successfully
#
```

startup_delay

Syntax:

setpefconfig startup_delay *value*
setpefconfig 3 *value*

Purpose:

This command sets the new value of the PEF parameter `startup_delay`. This parameter is a single byte, representing the number of seconds that the PEF facility delays at startup. The value is specified as a decimal number of seconds.

Example:

```
# clia setpefconfig startup_delay 45
Pigeon Point Shelf Manager Command Line Interpreter

PEF startup delay set successfully
#
```

alert_startup_delay

Syntax:

setpefconfig startup_delay *value*
setpefconfig 4 *value*

Purpose:

This command sets the current value of the PEF parameter `alert_startup_delay`. This parameter is a single byte, representing the number of seconds that the alerting facility delays at startup. The value is specified as a decimal number of seconds.

Example:

```
# clia setpefconfig alert_startup_delay 45
Pigeon Point Shelf Manager Command Line Interpreter

Alert startup delay set successfully
#
```

event_filter

Syntax:

`setpefconfig event_filter` *set-selector value1 … value19*
`setpefconfig 6` *set-selector value1 … value19*

Purpose:

This command sets the element of the event filter table with the index equal to *set-selector*. Indexes are 1-based. The contents of the new element are specified by 19 numeric values *value1* to *value19*, in hexadecimal, encoded according to the definition in table 15-2 of the IPMI specification version 1.5:

- Filter configuration
- Event filter action
- Alert policy number
- Event severity
- Generator ID byte 1
- Generator ID byte 2
- Sensor type
- Sensor number
- Event trigger (event/reading type)
- Event data 1 event offset mask
- Event data 1 AND mask
- Event data 1 compare 1
- Event data 1 compare 2
- Event data 2 AND mask
- Event data 2 compare 1
- Event data 2 compare 2
- Event data 3 AND mask

- Event data 3 compare 1
- Event data 3 compare 2

Example:

Set event filter 2 to trigger an alert action when an IPM controller at address 9C, FRU 0, reaches state M0 (the alert will be sent according with the Alert Policy #1):

```
# clia setpefconfig event_filter 2 80 1 1 10 9C FF F0 FF FF FF FF
0F FF 0 0 0 0 FF FF 0
Pigeon Point Shelf Manager Command Line Interpreter

Event filter set successfully
#
```

event_filter_data1

Syntax:

`setpefconfig event_filter_data1` *set-selector value*
`setpefconfig 7` *set-selector value*

Purpose:

This command sets the first byte of the element of the event filter table with the index equal to *set-selector*. Indexes are 1-based. This byte should be specified in hexadecimal. Bits in this byte have the following meaning:

- 0x80 – This filter is enabled
- 0x40 – This filter is preconfigured by the manufacturer and should not be altered by software

Other bits are reserved and should be 0.

This command can be used to quickly toggle the enabled/disabled state of an event filter; that is, turn it on and off without rewriting the whole table entry.

Examples:

Turn on event filter 2.

```
# clia setpefconfig event_filter_data1 2 80
Pigeon Point Shelf Manager Command Line Interpreter
```

```
Event filter data1 set successfully
#
```

Turn off event filter 2.

```
# clia setpefconfig event_filter_data1 2 0
Pigeon Point Shelf Manager Command Line Interpreter

Event filter data1 set successfully
#
```

alert_policy

Syntax:

`setpefconfig alert_policy` *set-selector value1 value2 value3 value4 value5*
`setpefconfig 9` *set-selector value1 value2 value3 value4 value5*

Purpose:

This command sets an alert policy table entry identified by the specified set selector. The contents of the new element are specified by the following five numeric values *value1* to *value5*, in hexadecimal, encoded according to the definition in Table 15-4 of IPMI 1.5:

- Policy number (4-bit value)
- Policy (4-bit value); includes the enable/disable bit
- Channel number (4-bit value)
- Destination selector (4-bit value)
- Alert string set/selector

Example:

The following example sets up the alert policy table entry 2 with the following attributes:

- Policy number = 5
- Enabled Policy = always send alert to this destination
- Destination channel = 1
- Destination selector = 1

- Alert String selector = use string 1 for all events

```
# clia setpefconfig alert_policy 2 5 8 1 1 1
Pigeon Point Shelf Manager Command Line Interpreter

Policy set successfully
#
```

system_guid

Syntax:

setpefconfig system_guid *guid-value*
setpefconfig 10 *guid-value*
setpefconfig system_guid none
setpefconfig 10 none

Purpose:

This command sets the current value of the PEF parameter `system_guid`. This parameter represents the GUID that is sent in a PET Trap PDU to an alert destination. This GUID may be defined as a separate GUID or as being equal to the System GUID.

The *guid-value* can be specified as an actual GUID, conforming to the standard GUID format `xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx`, or as a symbolic value `none`. In the first case, the PEF facility uses the specified GUID in PET Traps. In the second case, the PEF facility defaults to the System GUID (the result of the IPMI `Get System GUID` command) for PET Traps.

Examples:

```
# clia setpefconfig system_guid 23662F7F-BA1B-4b65-8808-
94CA09C9BBB0
Pigeon Point Shelf Manager Command Line Interpreter

GUID set successfully
#
# clia setpefconfig system_guid none
Pigeon Point Shelf Manager Command Line Interpreter

Using the system GUID
#
```

alert_string_key

Syntax:

`setpefconfig alert_string_key` *set-selector value1 value2*
`setpefconfig 12` *set-selector value1 value2*

Purpose:

This command sets the element of the alert string key table with the index equal to *set-selector*. Indexes are 1-based. Set-selector 0 can be used to designate the volatile alert string. Each key associates an event filter with an alert string for the alert generation purposes, and consists of the event filter number and the alert string number. Both values are 8-bit values and are specified by the parameters *value1* and *value2* respectively, in hexadecimal.

Example:

```
# clia setpefconfig alert_string_key 2 10 11
Pigeon Point Shelf Manager Command Line Interpreter

Alert string keys set successfully
#
```

alert_string

Syntax:

setpefconfig alert_string *set-selector* <string-value>
setpefconfig 13 *set-selector* <string-value>

Purpose:

This command sets the element of the alert string table with the index equal to *set-selector*. Indexes are 1-based. Index 0 can be used to designate the volatile alert string. The string value should be enclosed in double quotes (“) and may contain special characters and line feeds inside the quotes.

Example:

```
# clia setpefconfig alert_string 2 "This string has a line feed
inside."
Pigeon Point Shelf Manager Command Line Interpreter

Alert string set successfully
#
```

oem_filter

Syntax:

setpefconfig oem_filter *set-selector value1 value2 value3*
setpefconfig 97 *set-selector value1 value2 value3*

Purpose:

The OEM filter table is a Pigeon Point Systems-defined OEM extension of the IPMI specification. It allows PEF to be applied, in addition to platform events, to OEM-timestamped and non-timestamped SEL entries (record type range C0h-FFh).

Each entry of the OEM filter table defines the range of record types (in the range of OEM record types), to which this OEM filter applies, and the alert policy number that is to be invoked when a record with the matching record type is placed in the SEL.

This command sets an OEM filter table entry, the number of which is identified by the specified set selector. The entry consists of the following three numeric values:

- Byte 1: SEL Record Type Range Low boundary
- Byte 2: SEL Record type Range high boundary
- Byte 3: Alert policy number that will be invoked for SEL entries that have record types matching the range above

Example:

```
# clia getpefconfig oem_filter
Pigeon Point Shelf Manager Command Line Interpreter

Active OEM Filters:
    0x01: OEM range boundary 0xff:0xff, alert policy # 1
#
# clia setpefconfig oem_filter 4 0xdc 0xf3 5
Pigeon Point Shelf Manager Command Line Interpreter

OEM filter set successfully
#
# clia getpefconfig oem_filter
Pigeon Point Shelf Manager Command Line Interpreter

Active OEM Filters:
    0x01: OEM range boundary 0xff:0xff, alert policy # 1
    0x04: OEM range boundary 0xdc:0xf3, alert policy # 5
#
```

setpowerlevel

Syntax:

setpowerlevel *IPMB-address* *fruid* [*pwr_lvl* |OFF] [COPY]

Instead of the *IPMB-address*, the user may use:

board *n*
shm *n*

Purpose:

This command allows controlling the power level of a FRU and results in the Shelf Manager issuing a `Set Power Level` command on IPMB-0 to the designated IPM controller. Since the Shelf Manager core is responsible for managing power levels and tracking the corresponding power budgets, this command must be used with extreme care, especially when specifying a non-zero power level. Users of this command must be thoroughly familiar with the AdvancedTCA power management architecture as defined in the AdvancedTCA specification.

The target FRU is specified by the IPMB address of its IPM controller, plus the FRU device ID (*fruid)*. Alternatively, the board number or a dedicated Shelf Manager number can be used to designate the target FRU.

The third argument *pwr_lvl* is a power level. The power levels allowed are 0h to 14h, if available. A zero power level is equivalent to the keyword OFF; in that case, the command performs a power off of the designated FRU, if possible.

If no power level is specified, the command does not change the current power level of the FRU; this is equivalent to specifying 0xFF as the power level value. If specified, the power level is an index that selects one of the previously arranged power draw values for the designated FRU. Each power draw value corresponds to a maximum power draw (in Watts) that the FRU is authorized to use.

At any given time, an AdvancedTCA FRU that has been powered on has a set of up to 20 (14h) power draws that have been established between the FRU (actually, the IPM controller that represents the FRU) and the Shelf Manager. The *pwr_lvl* argument selects one of this set of power draws as the maximum power that the FRU is authorized to use.

Thereafter until another change is made, that FRU must not draw more than that authorized amount of power. The current and maximum power levels, plus the associated authorized power draw (in Watts) associated with the current power level, are available for any FRU via the `clia fru -v` command.

The optional parameter `COPY` specifies whether to copy the desired power levels to the present power levels (see the AdvancedTCA specification for background). If this parameter is not specified, the command does not copy the desired power levels to the present power levels.

Example:

```
# clia setpowerlevel 84 0 OFF
Pigeon Point Shelf Manager Command Line Interpreter

Operation completed with status = 0x0
#
```

setsensordata

Syntax:

`setsensordata` <*IPMB-address*> <*sensor-name*> [reading [-r] <value>] [assertion_<*mask*>] [deassertion_<*mask*>] [event_data <*b1*> <*b2*> <*b3*> | event_data_no_offset <*b1*> <*b2*> <*b3*>]

`setsensordata` <*IPMB-address*> [<*lun*>:]<*sensor-number*> [reading [-r] <*value*>] [assertion_<*mask*>] [deassertion_<*mask*>] [event_data <*b1*> <*b2*> <*b3*> | event_data_no_offset <*b1*> <*b2*> <*b3*>]

Using the `-d` option shows only discrete sensors having a specific state set in the state mask.

Purpose:

This command changes the reading, assertion/deassertion mask or event data bytes for the specified sensor. The sensor must be defined as settable, as defined in Addendum E372 to the IPMI specification v 2.0 (that is, bit [7] must be set to 1 in the Sensor Initialization byte of the corresponding SDR).

The sensor is specified by the IPMB address of the owning IPM controller and the sensor name or number. Alternatively, the board number or the dedicated Shelf Manager number can be used to designate the target IPM controller.

This command allows the user to qualify the sensor number with the Logical Unit Number (LUN) if the target controller supports sensors on multiple LUNs. <*lun*> can take the value 0, 1, or 3. (LUN 2 is reserved.) If the LUN is omitted, the command is applied to the sensor with the specified sensor number on the lowest LUN. (For example, if the command specifies sensor 3 without explicit LUN qualification, and the target controller exposes sensor 3 on LUN 1 and another sensor 3 on LUN 3, the command is applied to the sensor 3 on LUN 1.)

The parameters of this command follow the conventions of the IPMI command Set Sensor Reading and Event Status, defined in the Addendum E372 to the IPMI specification v 2.0.

The clause reading [-r] <*value*> specifies the new reading for the sensor. If the option -r is supplied, the <*value*> is interpreted as a raw value. Otherwise it is interpreted as a processed value and translated to the raw value using the SDR data (linearization, M, B, etc).

The clauses assertion <*mask*> and deassertion <*mask*> specify the current assertion/deassertion condition masks for the sensor, as 16-bit values. Their meaning is defined as follows.

For threshold-based sensors:

- [15:12] - reserved, must be 0000
- [11] - 1b = assertion/deassertion condition for upper non-recoverable going high
- [10] - 1b = assertion/deassertion condition for upper non-recoverable going low
- [9] - 1b = assertion/deassertion condition for upper critical going high
- [8] - 1b = assertion/deassertion condition for upper critical going low
- [7] - 1b = assertion/deassertion condition for upper non-critical going high
- [6] - 1b = assertion/deassertion condition for upper non-critical going low
- [5] - 1b = assertion/deassertion condition for lower non-recoverable going high
- [4] - 1b = assertion/deassertion condition for lower non-recoverable going low
- [3] - 1b = assertion/deassertion condition for lower critical going high
- [2] - 1b = assertion/deassertion condition for lower critical going low
- [1] - 1b = assertion/deassertion condition for lower non-critical going high
- [0] - 1b = assertion/deassertion condition for lower non-critical going low

For discrete sensors:

- [15] - reserved, must be 0
- [14] - 1b = state 14 assertion/deassertion event occurred
- [13] - 1b = state 13 assertion/deassertion event occurred
- [12] - 1b = state 12 assertion/deassertion event occurred
- [11] - 1b = state 11 assertion/deassertion event occurred
- [10] - 1b = state 10 assertion/deassertion event occurred
- [9] - 1b = state 9 assertion/deassertion event occurred
- [8] - 1b = state 8 assertion/deassertion event occurred
- [7] - 1b = state 7 assertion/deassertion event occurred
- [6] - 1b = state 6 assertion/deassertion event occurred
- [5] - 1b = state 5 assertion/deassertion event occurred

- [4] - 1b = state 4 assertion/deassertion event occurred
- [3] - 1b = state 3 assertion/deassertion event occurred
- [2] - 1b = state 2 assertion/deassertion event occurred
- [1] - 1b = state 1 assertion/deassertion event occurred
- [0] - 1b = state 0 assertion/deassertion event occurred

The clauses event_data *<b1>* *<b2>* *<b3>* and event_data_no_offset *<b1>* *<b2>* *<b3>* are used to specify the event data bytes that are sent in the IPMI Platform Event request when the sensor generates an event. If the variant event_data is used, the event offset (the lower nibble of the event data byte 1) is taken from *<b1>*; if the variant event_data_no_offset is used, the even offset is generated automatically when the event happens, and the lower nibble of the byte *<b1>* is ignored.

This command can also be issued on the backup Shelf Manager; in that case, the command can apply only to sensors that are local to the backup Shelf Manager.

In the following examples, the settable sensor Eth0 Front has the type Entity Presence and both monitors and controls the state of the Ethernet connection on the front panel of the carrier board. The command setsensordata is issued to change the sensor state from Entity Present to Entity Absent, which effectively turns off the Ethernet connection on the front panel.

Examples:

Show Eth0 Front Entity Present

```
# clia sensordata 10 10
Pigeon Point Shelf Manager Command Line Interpreter
10: LUN: 0, Sensor # 10 ("Eth0 Front")
Type: Discrete (0x6f), "Entity Presence" (0x25)
Belongs to entity (0xf0, 0x60): FRU # 0
Status: 0xc0
All event messages enabled from this sensor
Sensor scanning enabled
Initial update completed
Sensor reading: 0x00
Current State Mask 0x0001
Entity Present
```

Change Event Present to Absent (Assertion Event)

```
# clia setsensordata 10 10 assertion 2
Pigeon Point Shelf Manager Command Line Interpreter
Sensor data set successfully
```

Show Eth0 Front Entity Absent

```
# clia sensordata 10 10
Pigeon Point Shelf Manager Command Line Interpreter
10: LUN: 0, Sensor # 10 ("Eth0 Front")
Type: Discrete (0x6f), "Entity Presence" (0x25)
Belongs to entity (0xf0, 0x60): FRU # 0
Status: 0xc0
All event messages enabled from this sensor
Sensor scanning enabled
Initial update completed
Sensor reading: 0x00
Current State Mask 0x0002
Entity Absent
```

setsensoreventenable

Syntax:

`setsensoreventenable` *IPMB-address* *sensor-name* `global [assertion_events [deassertion_events]]`

`setsensoreventenable` *IPMB-address* `[`*lun*`:]`*sensor-number* `global [assertion_events [deassertion_events]]`

Instead of the *IPMB-address*, the user may use:

`board` *n*
`shm` *n*

Purpose:

This command changes the event enable mask for the specified sensor. The sensor is specified by the IPMB address of the owning IPM controller and the sensor name or number. Alternatively, the board number or dedicated Shelf Manager number can be used to designate the target IPM controller.

This command allows the user to qualify the sensor number with the Logical Unit Number (LUN) if the target controller supports sensors on multiple LUNs. *lun* can take value 0, 1, or 3. (LUN 2 is reserved.) If the LUN is omitted, the command is applied to the sensor with the specified sensor number on the lowest LUN. (For example, if the command specifies sensor 3 without explicit LUN qualification, and the target controller exposes sensor 3 on LUN 1 and another sensor 3 on LUN 3, the command is applied to the sensor 3 on LUN 1.)

Examples:

Enable the `Lower Non-Critical Going Low` event on the temperature sensor `Local Temp` on the IPM controller FE

```
# clia setsensoreventenable fe "Local Temp" 0x90 0x01 0x00
Pigeon Point Shelf Manager Command Line Interpreter

Event enable mask set successfully
#
# clia getsensoreventenable -v fe "Local Temp"
Pigeon Point Shelf Manager Command Line Interpreter

fe: LUN: 0, Sensor # 3 ("Local Temp")
    Type: Threshold (0x01), "Temperature" (0x01)
    Sensor scanning disabled
    Assertion event mask:   0x0001
       Assertion event for "Lower Non-Critical Going Low" enabled
    Deassertion event mask: 0x0000
#
```

Perform the same operation on the same sensor, but specify the sensor using LUN and sensor number:

```
# clia setsensoreventenable fe 0:3 0x90 0x01 0x00
Pigeon Point Shelf Manager Command Line Interpreter

Event enable mask set successfully
#
```

setthreshold

Syntax:

`setthreshold` *IPMB-address sensor-name threshold-type* `[-r]` *value*
`setthreshold` *IPMB-address* `[`*lun*`:]`*sensor-number threshold-type* `[-r]` *value*

Instead of *IPMB-address*, the user may use:

`board` *n*
`shm` *n*

Purpose:

This command changes the current threshold value for the specified threshold of the specified sensor. The sensor is specified by the IPMB address of the owning IPM controller and the sensor name or number. The target sensor must be a threshold-based sensor. The parameter *threshold-type* can be specified as one of the following symbolic values:

- `upper_non_recoverable` (can be abbreviated to `unr`)
- `upper_critical` (can be abbreviated to `uc`)
- `upper_non_critical` (can be abbreviated to `unc`)
- `lower_non_recoverable` (can be abbreviated to `lnr`)
- `lower_critical` (can be abbreviated to `lc`)
- `lower_non_critical` (can be abbreviated to `lnc`)

By default, the target value is specified in processed form (that is, in Volts for voltage sensors or in Celsius for temperature sensors). Option `-r` means that a raw value is used instead (usually a byte-size quantity, converted according to sensor-specific rules).

This command allows the user to qualify the sensor number with the Logical Unit Number (LUN) if the target controller supports sensors on multiple LUNs. *lun* can take the value 0, 1, or 3. (LUN 2 is reserved.) If the LUN is omitted, the command is applied to the sensor with the specified sensor number on the lowest LUN. (For example, if the command specifies sensor 3 without explicit LUN qualification, and the target controller exposes sensor 3 on LUN 1 and another sensor 3 on LUN 3, the command is applied to the sensor 3 on LUN 1.)

Example:

Set the upper non-critical threshold value for the temperature sensor `emulated temp` on IPM controller 9C to 99 degrees Celsius.

```
# clia threshold 9c 2
Pigeon Point Shelf Manager Command Line Interpreter

9c: LUN: 0, Sensor # 2 ("emulated temp")
    Type: Threshold (0x01), "Temperature" (0x01)
        Lower Non-Critical Threshold, Raw Data: 0x03, Processed
Data: 3.000000 degrees C
        Lower Critical Threshold, Raw Data: 0x14, Processed Data:
20.000000 degrees C
       Lower Non-Recoverable Threshold, Raw Data: 0xfb, Processed
Data: -5.000000 degrees C
        Upper Non-Critical Threshold, Raw Data: 0x46, Processed
Data: 70.000000 degrees C
```

```
        Upper Critical Threshold, Raw Data: 0x50, Processed Data:
80.000000 degrees C
       Upper Non-Recoverable Threshold, Raw Data: 0x5a, Processed
Data: 90.000000 degrees C
#
# clia setthreshold 9c 0:2 unc 99
Pigeon Point Shelf Manager Command Line Interpreter

Threshold set successfully
#
# clia threshold 9c 0:2
Pigeon Point Shelf Manager Command Line Interpreter

9c: LUN: 0, Sensor # 2 ("emulated temp")
    Type: Threshold (0x01), "Temperature" (0x01)
        Lower Non-Critical Threshold, Raw Data: 0x03, Processed
Data: 3.000000 degrees C
        Lower Critical Threshold, Raw Data: 0x14, Processed Data:
20.000000 degrees C
       Lower Non-Recoverable Threshold, Raw Data: 0xfb, Processed
Data: -5.000000 degrees C
        Upper Non-Critical Threshold, Raw Data: 0x63, Processed
Data: 99.000000 degrees C
        Upper Critical Threshold, Raw Data: 0x50, Processed Data:
80.000000 degrees C
       Upper Non-Recoverable Threshold, Raw Data: 0x5a, Processed
Data: 90.000000 degrees C
#
```

setuserlabel

Syntax:

`setuserlabel shelf` *shelf-name*
`setuserlabel slot` *slot-number slot-name*

Purpose:

This command configures the user assigned name for the Sun Netra CT900 server shelf and slots. Valid slot numbers are from 1-14. User assigned labels are displayed using the `userlabel` command.

User assigned shelf and slot names are text strings that can be used to uniquely identify the shelf and board during a console session. See "Displaying Board's User Label During Console Session" on page 151 for more information.

Examples:

```
# clia setuserlabel shelf ATCA02

Pigeon Point Shelf Manager Command Line Interpreter

 Shelf label set successfully
#
# clia setuserlabel slot 6 CP3020-06

Pigeon Point Shelf Manager Command Line Interpreter

    Slot#6 label set successfully
#
```

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 command `shelf` shows key Shelf FRU information, plus selected current operating data for the shelf, and allows modifying some fields in the Shelf FRU information. The type of the information this command shows or modifies is specified in the command parameter.

The following subsections describe the syntax of the `shelf` command for different applications of this command.

Display Shelf FRU Information

Syntax:

```
shelf [cooling_state | fans_state | address_table
       | power_distribution | power_management
       | pci_connectivity | ha_connectivity
       | h110_connectivity | point-to-point_connectivity]
```

Purpose:

This syntax of the command `shelf` shows key Shelf FRU information, plus selected current operating data for the shelf. The type of the information this command shows is specified in the command parameter. TABLE A-5 lists the parameters supported by the `shelf` command:

TABLE A-5 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 of them

TABLE A-5 Parameter for `shelf` Command *(Continued)*

`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. It should be entered before the command parameter: `clia shelf -v cooling_state`. If used, the command will display 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).

Examples:

Get shelf cooling status.

```
# clia shelf cooling_state
Pigeon Point Shelf Manager Command Line Interpreter

Cooling state: "Normal"
#
```

Get shelf fan tachometer status (verbose).

```
# clia shelf -v fans_state
Pigeon Point Shelf Manager Command Line Interpreter

Fans state: "Major Alert"
 Sensor(s) at this state: (0x7e,10) (0x7e,11) (0x7e,12) (0x7e,13)
                          (0x7e,14) (0x7e,15) (0x7e,16) (0x7e,17)
#
```

Get address table.

```
# clia shelf address_table
Pigeon Point Shelf Manager Command Line Interpreter

PICMG Address Table Record (ID=0x10)
        Version = 1
    Shelf Address            =
    Address Table Entries#   = 16
    Hw Addr: 41, Site # 1, Type: "AdvancedTCA Board" 00
    Hw Addr: 42, Site # 2, Type: "AdvancedTCA Board" 00
    Hw Addr: 43, Site # 3, Type: "AdvancedTCA Board" 00
    Hw Addr: 44, Site # 4, Type: "AdvancedTCA Board" 00
    Hw Addr: 45, Site # 5, Type: "AdvancedTCA Board" 00
    Hw Addr: 46, Site # 6, Type: "AdvancedTCA Board" 00
    Hw Addr: 47, Site # 7, Type: "AdvancedTCA Board" 00
    Hw Addr: 48, Site # 8, Type: "AdvancedTCA Board" 00
    Hw Addr: 49, Site # 9, Type: "AdvancedTCA Board" 00
    Hw Addr: 4a, Site # 10, Type: "AdvancedTCA Board" 00
    Hw Addr: 4b, Site # 11, Type: "AdvancedTCA Board" 00
    Hw Addr: 4c, Site # 12, Type: "AdvancedTCA Board" 00
    Hw Addr: 4d, Site # 13, Type: "AdvancedTCA Board" 00
    Hw Addr: 4e, Site # 14, Type: "AdvancedTCA Board" 00
#
```

Get power distribution information.

```
# clia shelf power_distribution
Pigeon Point Shelf Manager Command Line Interpreter

Power Distribution:
    Feed count: 1
    Feed 00:
        Maximum External Available Current: 50.0 Amps
        Maximum Internal Current: Not specified
        Minimum Expected Operating Voltage: -40.5 Volts
        Actual Power Available: 2025.000 Watts
        Currently Used Power: 160.000 Watts
        Feed-to-FRU Mapping entries count: 16
           FRU Addr: 41, FRU ID: fe
           FRU Addr: 42, FRU ID: fe
           FRU Addr: 43, FRU ID: fe
           FRU Addr: 44, FRU ID: fe
           FRU Addr: 45, FRU ID: fe
           FRU Addr: 46, FRU ID: fe
           FRU Addr: 47, FRU ID: fe
           FRU Addr: 48, FRU ID: fe
           FRU Addr: 49, FRU ID: fe
           FRU Addr: 4a, FRU ID: fe
           FRU Addr: 4b, FRU ID: fe
           FRU Addr: 4c, FRU ID: fe
           FRU Addr: 4d, FRU ID: fe
           FRU Addr: 4e, FRU ID: fe
           FRU Addr: 4f, FRU ID: fe
           FRU Addr: 50, FRU ID: fe
#
```

Get power management information

```
# clia shelf pm
Pigeon Point Shelf Manager Command Line Interpreter

PICMG Shelf Activation And Power Management Record (ID=0x12)
        Version = 0
    Allowance for FRU Activation Readiness: 10 seconds
    FRU Activation and Power Description Count: 16
    Hw Address: 41, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds
```

```
    Hw Address: 42, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
         Shelf Manager Controlled Activation: Enabled
         Delay Before Next Power On: 0.0 seconds
    Hw Address: 43, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
         Shelf Manager Controlled Activation: Enabled
         Delay Before Next Power On: 0.0 seconds
    Hw Address: 44, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
         Shelf Manager Controlled Activation: Enabled
         Delay Before Next Power On: 0.0 seconds
    Hw Address: 45, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
         Shelf Manager Controlled Activation: Enabled
         Delay Before Next Power On: 0.0 seconds
    Hw Address: 46, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
         Shelf Manager Controlled Activation: Enabled
         Delay Before Next Power On: 0.0 seconds
    Hw Address: 47, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
         Shelf Manager Controlled Activation: Enabled
         Delay Before Next Power On: 0.0 seconds
    Hw Address: 48, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
         Shelf Manager Controlled Activation: Enabled
         Delay Before Next Power On: 0.0 seconds
    Hw Address: 49, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
         Shelf Manager Controlled Activation: Enabled
         Delay Before Next Power On: 0.0 seconds
    Hw Address: 4a, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
         Shelf Manager Controlled Activation: Enabled
         Delay Before Next Power On: 0.0 seconds
    Hw Address: 4b, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
         Shelf Manager Controlled Activation: Enabled
         Delay Before Next Power On: 0.0 seconds
    Hw Address: 4c, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
         Shelf Manager Controlled Activation: Enabled
         Delay Before Next Power On: 0.0 seconds
    Hw Address: 4d, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
```

```
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds
    Hw Address: 4e, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds
    Hw Address: 4f, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds
    Hw Address: 50, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds
#
```

Modify Maximum External Available Current

Syntax:

`shelf maxcurrent` [*feed*] *current*

Purpose:

This command sets the Maximum External Available Current for the specified feed number and updates all known instances of shelf FRU information in the shelf. If the *feed* parameter is omitted, the value is set for the first feed (feed 0) in the shelf FRU information.

The parameter *feed* is a 0-based feed number in the shelf FRU information based on the sequential order of the description of that feed.

The parameter *current* is the desired current value in Amps.

Example:

Change the Maximum Available External Current for Feed 0 from 50 Amps to 99 Amps.

```
# clia shelf pd
Pigeon Point Shelf Manager Command Line Interpreter

PICMG Shelf Power Distribution Record (ID=0x11)
        Version = 0
    Feed count: 1
```

```
    Feed 00:
        Maximum External Available Current: 50.0 Amps
        Maximum Internal Current: Not specified
        Minimum Expected Operating Voltage: -40.5 Volts
        Actual Power Available: 2025.000 Watts
        Currently Used Power: 200.000 Watts
        Feed-to-FRU Mapping entries count: 16
           FRU Addr: 41, FRU ID: 0xfe
           FRU Addr: 42, FRU ID: 0xfe
           FRU Addr: 43, FRU ID: 0xfe
           FRU Addr: 44, FRU ID: 0xfe
           FRU Addr: 45, FRU ID: 0xfe
           FRU Addr: 46, FRU ID: 0xfe
           FRU Addr: 47, FRU ID: 0xfe
           FRU Addr: 48, FRU ID: 0xfe
           FRU Addr: 49, FRU ID: 0xfe
           FRU Addr: 4a, FRU ID: 0xfe
           FRU Addr: 4b, FRU ID: 0xfe
           FRU Addr: 4c, FRU ID: 0xfe
           FRU Addr: 4d, FRU ID: 0xfe
           FRU Addr: 4e, FRU ID: 0xfe
           FRU Addr: 4f, FRU ID: 0xfe
           FRU Addr: 50, FRU ID: 0xfe

# clia shelf maxcurrent 0 99
Pigeon Point Shelf Manager Command Line Interpreter

Updating Shelf FRU Info
Cached information updated

# clia shelf pd
Pigeon Point Shelf Manager Command Line Interpreter

PICMG Shelf Power Distribution Record (ID=0x11)
        Version = 0
    Feed count: 1
    Feed 00:
        Maximum External Available Current: 99.0 Amps
        Maximum Internal Current: Not specified
        Minimum Expected Operating Voltage: -40.5 Volts
        Actual Power Available: 2025.000 Watts
        Currently Used Power: 200.000 Watts
        Feed-to-FRU Mapping entries count: 16
           FRU Addr: 41, FRU ID: 0xfe
           FRU Addr: 42, FRU ID: 0xfe
```

```
            FRU Addr: 43, FRU ID: 0xfe
            FRU Addr: 44, FRU ID: 0xfe
            FRU Addr: 45, FRU ID: 0xfe
            FRU Addr: 46, FRU ID: 0xfe
            FRU Addr: 47, FRU ID: 0xfe
            FRU Addr: 48, FRU ID: 0xfe
            FRU Addr: 49, FRU ID: 0xfe
            FRU Addr: 4a, FRU ID: 0xfe
            FRU Addr: 4b, FRU ID: 0xfe
            FRU Addr: 4c, FRU ID: 0xfe
            FRU Addr: 4d, FRU ID: 0xfe
            FRU Addr: 4e, FRU ID: 0xfe
            FRU Addr: 4f, FRU ID: 0xfe
            FRU Addr: 50, FRU ID: 0xfe
#
```

Modify Minimum Expected Operating Voltage

Syntax:

`shelf minvoltage [`*feed*`]` *voltage*

Purpose:

This command sets the Minimum Expected Operating Voltage for the specified feed number and updates all known shelf FRU information instances in the shelf. If the *feed* parameter is omitted, the value is set for the first feed (feed 0) in the shelf FRU information.

The parameter *feed* is a 0-based feed number in the shelf FRU Information based on the sequential order of the description of that feed.

The parameter *voltage* is the desired value.

Example:

Change the Minimum Expected Operating Voltage for the Feed 0

```
# clia shelf pd
Pigeon Point Shelf Manager Command Line Interpreter

PICMG Shelf Power Distribution Record (ID=0x11)
        Version = 0
    Feed count: 1
    Feed 00:
```

```
        Maximum External Available Current: 99.0 Amps
        Maximum Internal Current: Not specified
        Minimum Expected Operating Voltage: -40.5 Volts
        Actual Power Available: 2025.000 Watts
        Currently Used Power: 200.000 Watts
        Feed-to-FRU Mapping entries count: 16
           FRU Addr: 41, FRU ID: 0xfe
           FRU Addr: 42, FRU ID: 0xfe
           FRU Addr: 43, FRU ID: 0xfe
           FRU Addr: 44, FRU ID: 0xfe
           FRU Addr: 45, FRU ID: 0xfe
           FRU Addr: 46, FRU ID: 0xfe
           FRU Addr: 47, FRU ID: 0xfe
           FRU Addr: 48, FRU ID: 0xfe
           FRU Addr: 49, FRU ID: 0xfe
           FRU Addr: 4a, FRU ID: 0xfe
           FRU Addr: 4b, FRU ID: 0xfe
           FRU Addr: 4c, FRU ID: 0xfe
           FRU Addr: 4d, FRU ID: 0xfe
           FRU Addr: 4e, FRU ID: 0xfe
           FRU Addr: 4f, FRU ID: 0xfe
           FRU Addr: 50, FRU ID: 0xfe

# clia shelf minvoltage 0 -59
Pigeon Point Shelf Manager Command Line Interpreter

    Updating Shelf FRU Info
    Cached information updated

# clia shelf pd
Pigeon Point Shelf Manager Command Line Interpreter

    PICMG Shelf Power Distribution Record (ID=0x11)
        Version = 0
    Feed count: 1
    Feed 00:
        Maximum External Available Current: 99.0 Amps
        Maximum Internal Current: Not specified
        Minimum Expected Operating Voltage: -59.0 Volts
        Actual Power Available: 2025.000 Watts
        Currently Used Power: 200.000 Watts
        Feed-to-FRU Mapping entries count: 16
           FRU Addr: 41, FRU ID: 0xfe
           FRU Addr: 42, FRU ID: 0xfe
           FRU Addr: 43, FRU ID: 0xfe
```

```
            FRU Addr: 44, FRU ID: 0xfe
            FRU Addr: 45, FRU ID: 0xfe
            FRU Addr: 46, FRU ID: 0xfe
            FRU Addr: 47, FRU ID: 0xfe
            FRU Addr: 48, FRU ID: 0xfe
            FRU Addr: 49, FRU ID: 0xfe
            FRU Addr: 4a, FRU ID: 0xfe
            FRU Addr: 4b, FRU ID: 0xfe
            FRU Addr: 4c, FRU ID: 0xfe
            FRU Addr: 4d, FRU ID: 0xfe
            FRU Addr: 4e, FRU ID: 0xfe
            FRU Addr: 4f, FRU ID: 0xfe
            FRU Addr: 50, FRU ID: 0xfe
#
```

Modify Shelf Manager Controlled Activation Flag

Syntax:

`shelf activation` *hardware-addr fru-id* `[1/0]`
`shelf activation board` *n* `[1/0]`
`shelf activation board all [1/0]`
`shelf activation fan_tray` *n* `[1/0]`

Purpose:

This command displays or changes the Shelf Manager Controlled Activation field for the specified FRU of the specified IPM controller. The command modifies the Shelf Manager Controlled Activation flag only for already existing entries in the Shelf Activation and Power Management record. This command also updates the cached version of the Shelf FRU Information used by the Shelf Manager. Thus the new value of the Shelf Manager Controlled Activation field becomes effective immediately without the need to restart the Shelf Manager.

The parameter *hardware-addr* is the 7-bit hardware address in hexadecimal format.

The parameter *fru-id* is a FRU ID in hexadecimal format; 0xFE means ALL FRUs at that hardware address.

Example:

Enable Shelf Manager Controlled Activation on an IPM controller with hardware address 0x42 (IPMB address 0x84).

```
# clia shelf pm
Pigeon Point Shelf Manager Command Line Interpreter

PICMG Shelf Activation And Power Management Record (ID=0x12)
        Version = 0
    Allowance for FRU Activation Readiness: 10 seconds
    FRU Activation and Power Description Count: 16
    Hw Address: 41, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 42, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 43, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 44, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 45, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 46, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 47, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds
```

```
    Hw Address: 48, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
         Shelf Manager Controlled Activation: Enabled
         Delay Before Next Power On: 0.0 seconds

    Hw Address: 49, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
         Shelf Manager Controlled Activation: Enabled
         Delay Before Next Power On: 0.0 seconds

    Hw Address: 4a, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
         Shelf Manager Controlled Activation: Enabled
         Delay Before Next Power On: 0.0 seconds

    Hw Address: 4b, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
         Shelf Manager Controlled Activation: Enabled
         Delay Before Next Power On: 0.0 seconds

    Hw Address: 4c, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
         Shelf Manager Controlled Activation: Enabled
         Delay Before Next Power On: 0.0 seconds

    Hw Address: 4d, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
         Shelf Manager Controlled Activation: Enabled
         Delay Before Next Power On: 0.0 seconds

    Hw Address: 4e, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
         Shelf Manager Controlled Activation: Enabled
         Delay Before Next Power On: 0.0 seconds

    Hw Address: 4f, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
         Shelf Manager Controlled Activation: Enabled
         Delay Before Next Power On: 0.0 seconds

    Hw Address: 50, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
         Shelf Manager Controlled Activation: Enabled
         Delay Before Next Power On: 0.0 seconds
```

```
# clia shelf activation 42 0xfe 0
Pigeon Point Shelf Manager Command Line Interpreter

    Updating Shelf FRU Info, address: 0x42, FRU ID # 254
    Cached information updated
    Wrote Information to the Shelf FRU

# clia shelf pm
Pigeon Point Shelf Manager Command Line Interpreter

PICMG Shelf Activation And Power Management Record (ID=0x12)
        Version = 0
    Allowance for FRU Activation Readiness: 10 seconds
    FRU Activation and Power Description Count: 16
    Hw Address: 41, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 42, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 43, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 44, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 45, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 46, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds
```

```
    Hw Address: 47, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 48, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 49, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 4a, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 4b, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 4c, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 4d, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 4e, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 4f, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds
```

```
    Hw Address: 50, FRU ID: 0xfe, Maximum FRU Power Capabilities:
150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds
#
```

Modify Shelf Manager Controlled Deactivation Flag

Syntax:

shelf deactivation *hardware-addr* *fru-id* [1|0]
shelf deactivation board *n* [1|0]
shelf deactivation board all [1|0]
shelf deactivation fan_tray *n* [1|0]

Purpose:

This command displays or changes the Shelf Manager Controlled Deactivation field for the specified FRU of the specified IPM controller. The command modifies the Shelf Manager Controlled Deactivation flag only for already existing entries in the Shelf Activation and Power Management record. This command also updates the cached version of the Shelf FRU Information used by the Shelf Manager. Thus the new value of the Shelf Manager Controlled Deactivation field becomes effective immediately without the need to restart the Shelf Manager.

The parameter *hardware-addr* is the 7-bit hardware address in hexadecimal format.

The parameter *fru-id* is a FRU ID in hexadecimal format; 0xFE means ALL FRUs at that hardware address.

A value of `0` *enables* the Shelf Manager Controlled Deactivation which gives the Shelf Manager the responsible for deactivating the board. A value of `1` *disables* the Shelf Manager Controlled Deactivation which prevents the Shelf Manager from automatically deactivating the board. When the Shelf Manager Controlled Deactivation is disabled, the system administrator is responsible for board deactivation.

Example:

Enabling Shelf Manager Controlled Deactivation on an IPM controller with hardware address 0x42 (IPMB address 0x84).

```
# clia shelf pm
Pigeon Point Shelf Manager Command Line Interpreter

 PICMIG Shelf Activation And Power Management Record (ID=0x12)
        Version = 1
    Allowance for FRU Activation Readiness: 10 seconds
    FRU Activation and Power Description Count: 16
    Hw Address: 41, FRU ID: 0xfe, Maximum FRU Power Capabilities:
200 Watts
        Shelf Manager Controlled Auto-Activation: Disabled
        Shelf Manager Controlled Auto-Deactivation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 42, FRU ID: 0xfe, Maximum FRU Power Capabilities:
200 Watts
        Shelf Manager Controlled Auto-Activation: Disabled
        Shelf Manager Controlled Auto-Deactivation: Disabled
        Delay Before Next Power On: 0.0 seconds

# clia shelf deactivation 42 0xfe 0
Pigeon Point Shelf Manager Command Line Interpreter

    Updating Shelf FRU Info, address: 0x42, FRU ID # 254
    Cached information updated
    Wrote Information to the Shelf FRU

# clia shelf pm
Pigeon Point Shelf Manager Command Line Interpreter

PICMG Shelf Activation And Power Management Record (ID=0x12)
        Version = 1
    Allowance for FRU Activation Readiness: 10 seconds
    FRU Activation and Power Description Count: 16
    Hw Address: 41, FRU ID: 0xfe, Maximum FRU Power Capabilities:
200 Watts
        Shelf Manager Controlled Auto-Activation: Disabled
        Shelf Manager Controlled Auto-Deactivation: Enabled
        Delay Before Next Power On: 0.0 seconds
```

```
    Hw Address: 42, FRU ID: 0xfe, Maximum FRU Power Capabilities:
200 Watts
        Shelf Manager Controlled Auto-Activation: Disabled
        Shelf Manager Controlled Auto-Deactivation: Enabled
        Delay Before Next Power On: 0.0 seconds

#
```

Modify Maximum FRU Power Capability

Syntax:

`shelf pwrcapability` *hardware-addr fru-id value*
`shelf pwrcapability board` *n value*
`shelf pwrcapability fan_tray` *n value*

Purpose:

This command changes the Maximum FRU Power Capability field for the specified FRU of the specified IPM controller.

Note – Never set the Maximum FRU Power Capability field to a larger value than is safe for your shelf environment.

The command modifies this field only for already existing entries in the Shelf Activation and Power Management record. This command also updates the cached version of the Shelf FRU Information used by the Shelf Manager. Thus the new value of the Maximum FRU Power Capability field becomes effective immediately without the need to restart the Shelf Manager.

- The parameter *hardware-addr* is the 7-bit hardware address in hexadecimal format.
- The parameter *fru-id* is a FRU ID in hexadecimal format; 0xFE means ALL FRUs at that hardware address.
- The parameter *value* is the new value for the field in Watts. The range of the possible values is 0..65535.

Example:

Set Maximum FRU Power Capability on an IPM controller with hardware address 0x42 (IPMB address 0x84) to 150 Watts.

```
# clia shelf pm
Pigeon Point Shelf Manager Command Line Interpreter

    Power Management:
    Allowance for FRU Activation Readiness: 10 seconds
    FRU Activation and Power Description Count: 2
    Hw Address: 41, FRU ID: fe, Maximum FRU Power Capabilities: 
200 Watts
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 2.2 seconds

    Hw Address: 42, FRU ID: fe, Maximum FRU Power Capabilities: 
200 Watts
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 2.2 seconds
#
# clia shelf pwrcapability 42 0xfe 150
Pigeon Point Shelf Manager Command Line Interpreter

Updating Shelf FRU Info
    Cached information updated
#
# clia shelf pm
Pigeon Point Shelf Manager Command Line Interpreter

Power Management:
    Allowance for FRU Activation Readiness: 10 seconds
    FRU Activation and Power Description Count: 2
    Hw Address: 41, FRU ID: fe, Maximum FRU Power Capabilities: 
200 Watts
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 2.2 seconds

    Hw Address: 42, FRU ID: fe, Maximum FRU Power Capabilities: 
150 Watts
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 2.2 seconds
#
```

Modify Delay Before Next Power On

Syntax:

shelf pwrdelay *hardware-addr fru-id value*
shelf pwrdelay board *n value*
shelf pwrdelay fan_tray *n value*

Purpose:

This command changes the Delay Before Next Power On field for the specified FRU of the specified IPM controller. The command modifies this field only for already existing entries in the Shelf Activation and Power Management record. This command also updates the cached version of the Shelf FRU Information used by the Shelf Manager. Thus the new value of the Delay Before Next Power On field becomes effective immediately without the need to restart the Shelf Manager.

- The parameter *hardware-addr* is the 7-bit hardware address in hexadecimal format.
- The parameter *fru-id* is a FRU ID in hexadecimal format; 0xFE means ALL FRUs at that hardware address.
- The parameter *value* is the new value for the field in tenths of a second. The range of the possible values is 0..63.

Example:

Set Delay Before Next Power On for an IPM controller with hardware address 0x42 (IPMB address 0x84) to 5 seconds.

```
# clia shelf pm
Pigeon Point Shelf Manager Command Line Interpreter

Power Management:
    Allowance for FRU Activation Readiness: 10 seconds
    FRU Activation and Power Description Count: 2
    Hw Address: 41, FRU ID: fe, Maximum FRU Power Capabilities:
200 Watts
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 2.2 seconds
    Hw Address: 42, FRU ID: fe, Maximum FRU Power Capabilities:
200 Watts
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 2.2 seconds
#
# clia shelf pwrdelay 42 0xfe 50
Pigeon Point Shelf Manager Command Line Interpreter
```

```
Updating Shelf FRU Info
    Cached information updated
#
# clia shelf pm
Pigeon Point Shelf Manager Command Line Interpreter

Power Management:
    Allowance for FRU Activation Readiness: 10 seconds
    FRU Activation and Power Description Count: 2
    Hw Address: 41, FRU ID: fe, Maximum FRU Power Capabilities:
200 Watts
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 2.2 seconds
    Hw Address: 42, FRU ID: fe, Maximum FRU Power Capabilities:
200 Watts
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 5.0 seconds
#
```

Modify Allowance for FRU Activation Readiness

Syntax:

`shelf allowance` *value*

Purpose:

This command changes the Allowance for FRU Activation Readiness parameter.

The parameter *value* is the new value for the parameter in seconds. The range of the possible values is 0..255.

Example:

Set Allowance for FRU Activation Readiness to 5 seconds.

```
# clia shelf pm
Pigeon Point Shelf Manager Command Line Interpreter
Power Management:

    Allowance for FRU Activation Readiness: 10 seconds
    FRU Activation and Power Description Count: 2
    Hw Address: 41, FRU ID: fe, Maximum FRU Power Capabilities:
200 Watts
```

```
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 2.2 seconds

    Hw Address: 42, FRU ID: fe, Maximum FRU Power Capabilities:
200 Watts
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 2.2 seconds
#
# clia shelf allowance 5
Pigeon Point Shelf Manager Command Line Interpreter

Updating Shelf FRU Info
#
# clia shelf pm
Pigeon Point Shelf Manager Command Line Interpreter
Power Management:
    Allowance for FRU Activation Readiness: 5 seconds
    FRU Activation and Power Description Count: 2
    Hw Address: 41, FRU ID: fe, Maximum FRU Power Capabilities:
200 Watts
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 2.2 seconds

    Hw Address: 42, FRU ID: fe, Maximum FRU Power Capabilities:
200 Watts
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 2.2 seconds
#
```

Reorder the FRU Activation and Power Descriptors

Syntax:

`shelf pwrreorder` *hardware-addr-1 fru-id-1* `before|after` *hardware-addr-2 fru-id-2*

hardware-address fru-id can be replaced by the following:

board *n*
fan_tray *n*

Purpose:

This command changes the order of the FRU Activation and Power Descriptors in the Shelf FRU Information. The command can reorder only the already existing descriptors. The current implementation is also limited to reordering the descriptors only inside a single Shelf Activation and Power Management record. This command also updates the cached version of the Shelf FRU Information used by the Shelf Manager. Thus the new order of the descriptors becomes effective immediately without the need to restart the Shelf Manager.

- The parameter *hardware-addr-1* is the 7-bit hardware address in hexadecimal format of the descriptor that needs to be moved.
- The parameter `fru`*-id-1* is a FRU ID in hexadecimal format of the descriptor that needs to be moved; 0xFE means ALL FRUs at that hardware address.
- The parameter *hardware-addr-2* is the 7-bit hardware address in hexadecimal format of the descriptor, before or after which the *hardware-addr-1*/`fru`*-id-1* descriptor should be placed.
- The parameter `fru`*-id-2*is a FRU ID in hexadecimal format of the descriptor, before/after which the *hardware-addr-1*/`fru`*-id-1* descriptor should be placed.

Example:

Place the descriptor for an IPM controller with hardware address 0x42 (IPMB address 0x84) before the descriptor for an IPM controller with hardware address 0x41 (IPMB address 0x82).

```
# clia shelf pm
Pigeon Point Shelf Manager Command Line Interpreter

Power Management:
    Allowance for FRU Activation Readiness: 10 seconds
    FRU Activation and Power Description Count: 2
    Hw Address: 41, FRU ID: fe, Maximum FRU Power Capabilities:
200 Watts
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 2.2 seconds

    Hw Address: 42, FRU ID: fe, Maximum FRU Power Capabilities:
200 Watts
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 2.2 seconds
#
# clia shelf pwrreorder 42 0xfe before 41 0xfe
Pigeon Point Shelf Manager Command Line Interpreter

Updating Shelf FRU Info
```

```
    Cached information updated
#
# clia shelf pm
Pigeon Point Shelf Manager Command Line Interpreter

Power Management:
    Allowance for FRU Activation Readiness: 10 seconds
    FRU Activation and Power Description Count: 2
    Hw Address: 42, FRU ID: fe, Maximum FRU Power Capabilities:
200 Watts
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 2.2 seconds

    Hw Address: 41, FRU ID: fe, Maximum FRU Power Capabilities:
200 Watts
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 2.2 seconds
#
```

Refresh the Shelf FRU Information

Syntax:

`shelf info_refresh`

Purpose:

This command causes the Shelf Manager to re-read the previously found sources of shelf FRU information in the shelf and reassess which of the sources contain valid shelf FRU information. Assuming that valid shelf FRU information is confirmed, all of the shelf FRU information storage devices and the cached master copy of the shelf FRU information are updated with the contents of the new shelf FRU information.

As specified by PICMG 3.0 (section 3.6.4), the Shelf Manager tries to find possible shelf FRU information storage devices during initialization. If the Shelf Manager finds at least two FRU Information devices that contain valid shelf FRU information, the Shelf Manager performs an `election` to determine which shelf FRU information sources to use. This election is based on validating the data the storage devices contain and comparing the contents. After a successful election, the Shelf Manager creates a cached master copy of the shelf FRU information (in volatile memory) which is used for any updating of shelf FRU information sources and is treated as the sole source of the shelf FRU information. Thus, all shelf FRU information-related operations work with the master copy and changes of the master copy are automatically propagated to all shelf FRU information source devices as incremental updates.

However, dynamic reconfiguration is not supported. If the new shelf FRU information is different from the previous shelf FRU information, the changes will become fully effective only after a reboot of the Shelf Manager.

Example:

Successful refresh: two matching sources of the Shelf FRU Info.

```
# clia shelf info_refresh
Pigeon Point Shelf Manager Command Line Interpreter

Read 0x20 # 2, size = 1024
Read 0x20 # 1, size = 1024
Found 2 Matching Shelf FRU Info

0x20 # 2, size = 1024 (data size = 775), "Valid" Shelf FRU,
"Matching"
0x20 # 1, size = 1024 (data size = 775), "Valid" Shelf FRU,
"Matching"
Shelf FRU Info was not changed

#
```

Example:

Unsuccessful refresh: both data sources contain non-matching or invalid data.

```
# clia shelf info_refresh
Pigeon Point Shelf Manager Command Line Interpreter

Read 0x20 # 2, size = 1024
Read 0x20 # 1, size = 1024
No Matching Shelf FRU Info found

0x20 # 2, size = 1024 (data size = 293), "Invalid" Shelf FRU, "Non-
Matching"
0x20 # 1, size = 1024 (data size = 529), "Valid" Shelf FRU, "Non-
Matching"
Refresh was not done because system found only 1 (of 2) Matching
Shelf FRU info

#
```

Update the Shelf FRU Info Storage Devices

Syntax:

`shelf info_force_update`

Purpose:

This command causes a check of the shelf FRU information source devices and copying the contents of the shelf FRU information master copy to all of them. This command is useful in the case of a conflict between the shelf FRU information master copy and the non-volatile source devices, where the conflict is not resolved automatically (for example both EEPROMs and the master copy are different from each other). In that case, the operator can forcibly synchronize the EEPROMs with the contents of the master copy, using this command. Also, this command clears the error condition that has occurred due to the original conflict; that is, after this command has been issued, subsequent updates to the shelf FRU information will resume being propagated to the SEEPROMs.

This command initiates an update of the shelf FRU information source devices in an asynchronous fashion.

Example:

```
# clia shelf info_force_update
Pigeon Point Shelf Manager Command Line Interpreter

Starting the Shelf FRU Info source device update
#
```

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).

The shelf address is stored with a type indicator that differentiates between text and binary data. If the shelf address is specified as text, it is stored as text. If the shelf address is specified in hexadecimal, it is stored as binary data.

When the shelf address is displayed using this command, it is printed as text or as a sequence of hexadecimal bytes, depending on the type indicator.

Examples:

```
# clia shelfaddress
Pigeon Point Shelf Manager Command Line Interpreter

 Shelf Address Info: "1234"
#
# 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"
#
# clia shelfaddress -x 01 02 03 04 05
Pigeon Point Shelf Manager Command Line Interprete

 Shelf Address Info set successfully
#
# clia shelfaddress
Pigeon Point Shelf Manager Command Line Interpreter

 Shelf Address Info: " 0x01 0x02 0x03 0x04 0x05 "
#
```

shmstatus

Syntax:

`shmstatus`

Purpose:

This command returns the Shelf Manager status in redundant configurations: Active or Backup. In verbose mode it reports a more detailed picture: status of the shelf FRU information, status of the RMCP interface, and state of the backup Shelf Manager (if the Shelf Manager being queried is the active one). The ready for operation flag is a parameter that shows as `Yes`:

- On the active Shelf Manager if it finds valid shelf FRU information and successfully initializes its RMCP interface.
- On the backup Shelf Manager if it successfully received the redundancy state information from the active Shelf Manager.

Example:

```
# clia shmstatus -v
Pigeon Point Shelf Manager Command Line Interpreter

Shelf Manager status: "Active"
Ready For Operation: Yes
Detailed State Flags: "Shelf FRU Found" "RMCP Up" "Backup Healthy"
#
```

showhost

Syntax:

`showhost` *slot-number* `[version]`

Purpose:

This command is used to display the system firmware versions installed on Sun Netra node boards. This command is only valid for Sun Netra CP3x60 node boards installed in the Sun Netra CT900 server.

The parameter *slot-number* specifies the slot number of the Sun Netra CP3x60 board and the *version* option is used to display full version information.

Example:

Show the current firmware versions for the Sun Netra CP3060 board in slot 2.

```
# clia showhost 2
Pigeon Point Shelf Manager Command Line Interpreter

System Frimware 6.2.5 Netra CP3060 2006/09/15 15:30

Host flash versions:
 Hypervisor 1.2.3 2006/08/18 12:25
 OBP 4.23.4 2006/08/04 20:46
```

```
 Netra[TM] CP3060 POST 4.23.4 2006/08/04 21:17

#
```

showunhealthy

Syntax:

`showunhealthy`

Purpose:

This command shows the list of FRUs that appear to have a problem. In the PICMG 3.0 context, this list includes FRUs for which the cause of last hot swap state change is `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.
#
```

switchover

Syntax:

`switchover`

`switchover -force`

Purpose:

This command initiates switchover of the redundant Shelf Manager instances. This command can be executed on either the active or the backup instance of the Shelf Manager.

Forced switchovers from the backup Shelf Manager to handle the case of a frozen active ShMM (CLI command switchover -force) are supported. No negotiations between the Shelf Managers happen in this case; the backup Shelf Manager immediately takes control of hardware.

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 initiates terminates the Shelf Manager without rebooting the shelf management card. If the current ShMM is active, a switchover takes place.

Example:

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

```
# clia terminate

#
```

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 and provides a simple way to add, delete, and modify the user accounts.

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

Display User Information

Syntax:

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

Purpose:

This command shows information about users. When it is launched with a `-v` option, it also shows information about disabled users. (By default, only enabled users are listed.) If the optional User ID is specified, 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.

Examples:

```
# clia user
Pigeon Point Shelf Manager Command Line Interpreter

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

1: ""
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
7: "TEST1" Disabled
        Channels 0-15 Privilege level: "NO ACCESS"
#
```

Add a New User

Syntax:

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

Purpose:

This command adds a new user to the system. It sets the same maximum privilege level and channel access flags for all channels, as specified in the command. The command will return an error if the specified user does not exist. Command parameters have the following meaning:

- *userid* – a valid user id;
- *user-name* – the user name (it will be truncated to the 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 will be truncated to the 16 characters without any notice)

Example:

Add user 9 with the name `root`, administrator privilege level, and password `PICMG guru`.

```
# clia user
Pigeon Point Shelf Manager Command Line Interpreter

1: ""
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
#
# clia user add 9 "root" 0x40 4 "PICMG guru"
Pigeon Point Shelf Manager Command Line Interpreter

User 9 added successfully
#
# clia user
Pigeon Point Shelf Manager Command Line Interpreter

1: ""
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
9: "root"
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
#
```

Delete a User

Syntax:

`user delete` *userid*

Purpose:

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

Example:

Delete the user with user ID = 10.

```
# clia user delete 10
Pigeon Point Shelf Manager Command Line Interpreter

User 10 deleted successful
#
# clia user
Pigeon Point Shelf Manager Command Line Interpreter

1: ""
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
9: "root"
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
#
```

Enable and Disable a User

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 – disable the specified user;
- non-zero – enable the specified user

Examples:

Disable and enable user with *userid* 9.

```
# clia user
Pigeon Point Shelf Manager Command Line Interpreter

1: ""
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
9: "root"
```

```
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
#
# clia user enable 9 0
Pigeon Point Shelf Manager Command Line Interpreter

    User 9 disabled successfully
#
# clia user -v
Pigeon Point Shelf Manager Command Line Interpreter

1: ""
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
9: "root" Disabled
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
#
# clia user enable 9 1
Pigeon Point Shelf Manager Command Line Interpreter

User 9 enabled successfuly
#
# clia user
Pigeon Point Shelf Manager Command Line Interpreter

1: ""
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
9: "root"
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
#
```

Modify a User Name

Syntax:

`user name` *userid* *user-name*

Purpose:

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

- *userid* – the valid user id
- *user-name* – the user name (which will be truncated to 16 characters without any notice)

Example:

Change the name of user 9 to `newby`.

```
# clia user
Pigeon Point Shelf Manager Command Line Interpreter

1: ""
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
9: "root"
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
#
# clia user name 9 newby
Pigeon Point Shelf Manager Command Line Interpreter

User 9, name changed successfully
#
# clia user
Pigeon Point Shelf Manager Command Line Interpreter

1: ""
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
9: "newby"
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
#
```

Modify a User's Password

Syntax:

`user passwd` *userid password*

Purpose:

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

- *userid* – the valid userid;
- *password* – the user password (which will be truncated to 16 characters without any notice)

Example:

Change the password of userid 9 to `RIP`

```
# clia user
Pigeon Point Shelf Manager Command Line Interpreter

1: ""
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
9: "newby"
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
#
# clia user passwd 9 RIP
Pigeon Point Shelf Manager Command Line Interpreter

User 9, password changed successfully
#
# clia user
Pigeon Point Shelf Manager Command Line Interpreter

1: ""
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
9: "newby"
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
#
```

Modify Channel Access Settings for a Specified User and a Specified 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 userid
- *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 9 on channel 5 to `User`:

```
# clia user 9
Pigeon Point Shelf Manager Command Line Interpreter

9: "newby"
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"
#
# clia user channel 9 5 0x60 2
Pigeon Point Shelf Manager Command Line Interpreter

User 9, channel 5 access updated successfully
#
# clia user 9
Pigeon Point Shelf Manager Command Line Interpreter

9: "newby"
        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"
#
```

userlabel

Syntax:

`userlabel shelf | slot [`*slot-number*`]`

Purpose:

This command displays the user assigned name for the shelf and slots in the server. Valid slot numbers are from 1-14. User assigned labels are set using the `setuserlabel` command.

User assigned shelf and slot names are text strings that can be used to uniquely identify the shelf and board during a console session. See "Displaying Board's User Label During Console Session" on page 151 for more information.

Examples:

```
# clia userlabel shelf

Pigeon Point Shelf Manager Command Line Interpreter

 Shelf: "ATCA02"
#
# clia userlabel slot

Pigeon Point Shelf Manager Command Line Interpreter

    Slot#  1:   "CP3260-01"
    Slot#  2:   "CP3220-02"
    Slot#  3:   ""
    Slot#  4:   ""
    Slot#  5:   "CP3260-05"
    Slot#  6:   "CP3020-06"
    Slot#  7:   "Switch-07"
    Slot#  8:   "Switch-08"
    Slot#  9:   ""
```

```
    Slot# 10:   "CP3260-10"
    Slot# 11:   ""
    Slot# 12:   "CP3220-12"
    Slot# 13:   ""
    Slot# 14:   ""
#
# clia userlabel slot 6

Pigeon Point Shelf Manager Command Line Interpreter

    Slot#  6:   "CP3020-06"
#
```

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. 2.4.2
Pigeon Point is a trademark of Pigeon Point Systems.
Copyright (c) 2002-2007 Pigeon Point Systems
All rights reserved
Build date/time: Jan 30 2007 16:39:37
Carrier: PPS; Subtype: 0; Subversion: 0
#
```

APPENDIX **B**

Shelf Manager IPMI Extension Commands and Sensors

The Shelf Manager implements several OEM-defined IPMI commands and sensors for the convenience of the System Manager.

The first set of mechanisms provides an alternative for Telco alarm control to the PEF-based mechanism. The mechanisms consist of one sensor and two OEM extension commands. The commands can be issued by the System Manager over the RMCP interface and implement the following functionality:

- Set/clear specified Telco alarms (both the set and clear operations are performed atomically within the same command)
- Get the number of the Telco alarm sensor. This sensor can then be used to read the current state of Telco alarms in an IPMI-compliant way.

Some ShMM carriers support general purpose digital output pins that are intended to be accessible by the System Manager and other external applications. There is no explicit support for such entities in the IPMI specification, so a second set of extension commands is provided to allow the System Manager to work with such digital outputs.

The commands can be issued by the System Manager over the RMCP interface and provide the following functionality:

- query the properties and number of available digital outputs
- get the current state of digital outputs
- set/clear digital outputs

Currently, these commands are implemented for selected carriers on some FRUs of the Shelf Manager (IPMB address 20h). However, in future they may be implemented on other IPM controllers/FRUs.

A final set of extensions provide additional functionality, currently including the following:

- Reading the contents of a Shelf FRU Information multirecord

- Notifying the Shelf Manager about an extracted FRU
- Initiating a Shelf Manager switchover
- Subscribing for event notifications

The commands in this set are implemented as OEM Group commands (Network Function Code 2Eh) and require that the first three bytes of the request and response be the Pigeon Point Systems IANA: (0Ah, 40h, 00h).

IPMI Extension Commands

The Pigeon Point Systems (PSS) extension IPMI commands are listed in TABLE B-1 and described in the following sections

TABLE B-1 Shell Manager IPMI Extension Commands

Command	NetFn	Opcode
Notify Shelf Manager About an Extracted FRU	2Eh	2h
Initiate Shelf Manager Switchover	2Eh	3h
Reset IPMC Configuration Variables	0x2E	0x9A
Set AMC Time Out Parameter	0x2E	0xF1
Get AMC Time Out Parameter	0x2E	0xF0
Set Boot Page	0x2E	0x81
Get Boot Page	0x2E	0x82
Set Front Panel Reset Button State	0x2E	0x83
Get Front Panel Reset Button State	0x2E	0x84
Set IPMC Control Bits	0x2E	0xE9
Get IPMC Control Bits	0x2E	0xE8
Set Management Port	0x2E	0x9B
Get Management Port	0x2E	0x9C
Get Shelf FRU Record Data	2Eh	1h
Set Shelf FRU Record Data	2Eh	05h
Set SOL Fail Over Link Change Timeouts	0x2E	0xE7
Get SOL Fail Over Link Change Timeouts	0x2E	0xE6
Get Version	0x2E	0x80

TABLE B-1 Shell Manager IPMI Extension Commands *(Continued)*

Command	NetFn	Opcode
Set Thermal Trip	0x2E	E5
Get Thermal Trip	0x2E	0xE4
Subscribe for Event Notifications	2Eh	4h

Notify Shelf Manager About an Extracted FRU

This extension command is sent to the logical Shelf Manager (20h) and informs the Shelf Manager that the specified FRU (which is typically in the communication lost (M7) state) is not in the shelf any more and can be moved to state M0, and have all its resources released. If the FRU Device ID is 0, the command applies to all FRUs represented by the specified IPM controller. This command is similar in effects to the CLI command `setextracted`.

Unless the `Forced Mode` is specified, the Shelf Manager verifies that the target FRU is indeed in the `Communication Lost` (M7) state; if not, the completion code D5 (Command Not Supported in Present State) is returned.

Command	NetFn	Opcode
Notify Shelf Manager About an Extracted FRU	2Eh	2h

Data Bytes

Type	Byte	Data Field
Request Data	1	PPS IANA Low Byte. A value 0Ah shall be used.
	2	PPS IANA Middle Byte. A value 40h shall be used.
	3	PPS IANA High Byte. A value 00h shall be used.
	4	IPMB Address. Indicates IPMB address of the target IPM Controller
	5	Target FRU Device ID. Indicates the FRU Device ID that is targeted by this command.
	6	Flags. An optional bit field: [7:1] Reserved; shall be set to 0 [0] Forced Mode. This bit is set to 1b if "forced mode" is to be used; in this mode the Shelf Manager does not check that the target FRU is indeed in state M7.

Type	Byte	Data Field
Response Data	1	Completion Code: [00] OK [C1] Command not supported [CC] Invalid data in request (Refer to IPMI specification for more completion codes)
	2	PPS IANA Low Byte. A value 0Ah shall be used.
	3	PPS IANA Middle Byte. A value 40h shall be used.
	4	PPS IANA High Byte. A value 00h shall be used.

Initiate Shelf Manager Switchover

This Pigeon Point extension command can be targeted to the logical Shelf Manager address (20h) or alternatively to the physical address for either the active or backup Shelf Managers. It initiates a switchover from the active to the backup Shelf Manager. If a switchover cannot be performed (for instance, if there is no backup Shelf Manager available), the completion code D5 (Command Not Supported in Present State) is returned.

Command	NetFn	Opcode
Initiate Shelf Manager Switchover	2Eh	3h

Data Bytes

Type	Byte	Data Field
Request Data	1	PPS IANA Low Byte. A value 0Ah shall be used.
	2	PPS IANA Middle Byte. A value 40h shall be used.
	3	PPS IANA High Byte. A value 00h shall be used.
	4	Flags. A bit field that specifies the action [0]: 1b = subscribe for event notifications on the current session; 0b = unsubscribe [7:1] Reserved. Shall be set to 0.

Type	Byte	Data Field
Response Data	1	Completion Code: [00] OK [C1] Command not supported [CC] Invalid data in request (Refer to IPMI specification for more completion codes)
	2	SPPS IANA Low Byte. A value 0Ah shall be used.
	3	PPS IANA Middle Byte. A value 40h shall be used.
	4	PPS IANA High Byte. A value 00h shall be used.

Reset IPMC Configuration Variables

IPMC configuration variables are retained across firmware upgrades and downgrades, as long as the version being downgraded to supports this feature.

IPMC has a group of parameters that are stored in SEEPROM (NV Storage). All variables are part of one data structure. Each of these variables performs a function and can be set or read via commands. These variables are stored across IPMC resets. However, previously, these variables were reset to default values when new or upgraded firmware was installed.

Set AMC Time Out Parameter

Use the following command from ShMM, Payload, or Debug interface to set the timeout values of AMC.

Command	NetFn	Opcode
Set AMC Time Out Parameter	0x2E	0xF1

Data Bytes

Type	Byte	Data Field
Request Data	1	00
	2	00
	3	6F or 2A (Sun legacy)
	4	Delay LSB

Type	Byte	Data Field
	5	Delay MSB
Response Data	1	Completion Code: [00] OK [C1] Command not supported [CC] Invalid data in request (Refer to IPMI specification for more completion codes)
	2	00
	3	00
	4	6F or 2A (Sun legacy)

Get AMC Time Out Parameter

Use the following command from ShMM, Payload, or Debug interface to read the default timeout values of AMC.

Command	NetFn	Opcode
Get AMC Time Out Parameter	0x2E	0xF0

Data Bytes

Type	Byte	Data Field
Request Data	1	00
	2	00
	3	6F or 2A (Sun legacy)
Response Data	1	Completion Code: [00] OK [C1] Command not supported [CC] Invalid data in request [CB] If parameter is not set. (Refer to IPMI specification for more completion codes)
	2	00
	3	00

Type	Byte	Data Field
	4	6F or 2A (Sun legacy)
	5	Delay LSB
	6	Delay MSB

Set Boot Page

Use the following command from ShMM, Payload, or Debug interface to set the BIOS boot page. The default value for the boot page is 0. Bits 7 to 1 should be set to zeroes. The value set by the user is stored in SEEPROM. Upon next boot, the same value of the boot page will be used.

Command	NetFn	Opcode
Set Boot Page	0x2E	0x82

Data Bytes

Type	Byte	Data Field
Request Data	1	00
	2	00
	3	6F or 2A (Sun legacy)
Response Data	1	Completion Code: [00] OK [C1] Command not supported [CC] Invalid data in request [CB] If parameter is not set. (Refer to IPMI specification for more completion codes)
	2	00
	3	00
	4	6F or 2A (Sun legacy)
	5	Boot page value. 0 = page 0, 1 = page 1.

Get Boot Page

Use the following command from ShMM, Payload, or Debug interface to read the BIOS boot page.

Command	NetFn	Opcode
Get Boot Page	0x2E	0x81

Data Bytes

Type	Byte	Data Field
Request Data	1	00
	2	00
	3	6F or 2A (Sun legacy)
	4	Boot page: 0 or 1
Response Data	1	Completion Code: [00] OK [C1] Command not supported [CC] Invalid data in request (Refer to IPMI specification for more completion codes)
	2	00
	3	00
	4	6F or 2A (Sun legacy)

Set Front Panel Reset Button State

Use the following command to change the way the front panel reset is handled by CPLD when this button is pressed. Default on CPLD power up is 10.

Command	NetFn	Opcode
Set Front Panel Reset Button State	0x2E	0x83

Data Bytes

Type	Byte	Data Field
Request Data	1	00
	2	00
	3	6F or 2A (Sun legacy)
	4	Front panel settings • Bits 0 and 1 = front panel reset state: –00 = Reset IPMC and hard reset to system –01 = NMI to system –10 = Hard reset to system –11 = Front panel reset button disabled • Bits 2 to 7 = 0
Response Data	1	Completion Code: [00] OK [C1] Command not supported [CC] Invalid data in request (Refer to IPMI specification for more completion codes)
	2	00
	3	00
	4	6F or 2A (Sun legacy)

Get Front Panel Reset Button State

Use the following command to read settings of the front panel reset button handling. By default on CPLD, power on it comes up as 10, i.e., pressing this button causes Power on Reset to CPU.

Command	NetFn	Opcode
Get Front Panel Reset Button State	0x2E	0x84

Data Bytes

Type	Byte	Data Field
Request Data	1	00
	2	00
	3	6F or 2A (Sun legacy)
Response Data	1	Completion Code: [00] OK [C1] Command not supported [CC] Invalid data in request (Refer to IPMI specification for more completion codes)
	2	00
	3	00
	4	6F or 2A (Sun legacy)
	5	Front panel settings • Bits 0 and 1 = front panel reset state: –00 = Reset IPMC and hard reset to system –01 = XIR to CPU –10 = Hard reset to system (POR to CPU) –11 = Front panel reset button disabled • Bits 2 to 7 = 0

Set IPMC Control Bits

Use the following command to set the configuration of the blade server's LED and the AMC shutdown behavior.

Command	NetFn	Opcode
Set IPMC Control Bits	0xE9	0xE9

Data Bytes

Type	Byte	Data Field
Request Data	1	00
	2	00
	3	6F or 2A (Sun legacy)
	4	Control byte. • Bit 0 = LED 2 (Green LED) control bit: –1 = IPMC controls green LED. –0 = IPMC does not control green LED. • Bit 1 = LED1 (Amber/Red LED) control bit. –1 = IPMC controls LED1 for default behavior. –0 = IPMC does not control LED1. • Bit 2= AMC latch control bit. –1 = IPMC intiates shutdown of AMC upon latch opening. –0 = IPMC does not initiate shutdown of AMC upon latch opening. • Bits 3 to 7 = Reserved for future use. Write as 1s. (See Note.)
Response Data	1	Completion Code: [00] OK [C1] Command not supported [CC] Invalid data in request (Refer to IPMI specification for more completion codes)
	2	00
	3	00
	4	6F or 2A (Sun legacy)

Note – If an attempt is made to write 0 to any reserved bit (3 to 7), IPMC will reject the command with completion code 0xCC.

Get IPMC Control Bits

Use the following command to display the current configuration of the blade server's OOS LED behavior.

Command	NetFn	Opcode
Get IPMC Control Bits	0x2E	0xE8

Data Bytes

Type	Byte	Data Field
Request Data	1	00
	2	00
	3	6F or 2A (Sun legacy)
Response Data	1	Completion Code: [00] OK [C1] Command not supported [CC] Invalid data in request (Refer to IPMI specification for more completion codes)
	2	00
	3	00
	4	6F or 2A (Sun legacy)
	5	IPMC control bit: • Bit 0: Green LED control bit. • Bit 1: LED1 control bit. • Bit 2: AMC latch control bit. • Bits 3 to 7: Reserved for future use.

Set Management Port

Use the following command to configure management port access to font or rear panel.

Command	NetFn	Opcode
Set Management Port	0x2E	0x9B

Data Bytes

Type	Byte	Data Field
Request Data	1	00
	2	00
	3	6F
	4	Control byte • Bit 0 = port access setting: 0 = Route to rear panel 1 = Route to front panel (default) • Bits 1 to 7 = Reserved. Write zeros.
Response Data	1	Completion Code: [00] OK [C1] Command not supported [CC] Invalid data in request (Refer to IPMI specification for more completion codes)
	2	00
	3	00
	4	6F

Get Management Port

Use the following command to display the current settings of the management port configuration (rear or front access).

Command	NetFn	Opcode
Get Management Port	0x2E	0x9C

Data Bytes

Type	Byte	Data Field
Request Data	1	00
	2	00
	3	6F
Response Data	1	Completion Code: [00] OK [C1] Command not supported [CC] Invalid data in request (Refer to IPMI specification for more completion codes)
	2	00
	3	00
	4	6F
	5	IPMC control bit: • Bit 0: 0 = Port routed to rear. 1 = Port routed to front (default). • Bits 1 to 7: Reserved for future use.

Set Shelf FRU Record Data

Using this Pigeon Point extension command, a specified range of bytes can be written into any multirecord in the Shelf FRU Information. This command can be applied to any PICMG-defined or OEM-defined record in the Shelf FRU Information, including the Address Table Record (PICMG Record ID = 10h), Shelf Power Distribution Record (PICMG Record ID = 11h), etc.

The type of the record to be written is specified by the Manufacturer IANA and manufacturer-specific record type. The 0-based record number can be used to distinguish different instances of the same type of record. (There may be several records of the same type in the Shelf FRU Information.)

The implementation of the command takes care of updating the checksum of the target record so that the checksum stays correct.

The number of bytes to be written by this command is limited by the size of IPMB packet and must be 15 bytes or fewer. This number is specified by the parameter "Byte Count" and must be equal to the actual number of data bytes in the command. If the actual number of data bytes in a command does not correspond to the value of the parameter "Byte Count", the command is rejected.

Command	NetFn	Opcode	IANA:
Set Shelf FRU Record Data	2Eh	05h	0400Ah (Assigned to PPS

Data Bytes

Type	Byte	Data Field
Request Data	1	PPS IANA Low Byte.
	2	PPS IANA Middle Byte.
	3	PPS IANA High Byte.
	4	Record Manufacturer IANA Low Byte.
	5	Record Manufacturer IANA Middle Byte.
	6	Record Manufacturer IANA High Byte.
	7	Record Type.
	8	Record Number. This field specifies the number of the record to be accessed. The record numbers are 0-based.
	9	Offset. This field specifies the offset from the beginning of the record in bytes.
	10	Byte Count.This field specifies the number of bytes to be written.
	11:N+10	
Response Data	1	Completion Code: [00] OK [C1] Command not supported [CC] Invalid data in request (Refer to IPMI specification for more completion codes)
	2	PPS IANA Low Byte.
	3	PPS IANA Middle Byte.
	4	PPS IANA High Byte.
	5	Count written. Indicates the number of bytes written to the record.
	6:N+5	Data. This variable length field contains data retrieved from the record. N is specified in the Count read byte.

For example, to set the first 9 bytes of the first Address Table record in the Shelf FRU Information, use the following parameters:

```
Record Manufacturer IANA Low Byte = 5Ah (PICMG)
Record Manufacturer IANA Middle Byte = 31h (PICMG)
Record Manufacturer IANA High Byte = 00h (PICMG)
Record type = 10h (Address Table)
Record Number = 00h (first record)
Offset = 00h
Byte Count = 09
Data = 41h 08h 00h 12h 09h 00 43h 0Ah 00(9 bytes)
```

This feature retains any custom values you configured for the following variables.

TABLE B-2 IPMC Variables for Which You Can Retain Configurations

Variable	Description
`log_level`	Controls the log level of the debug messages.
`Payload timeout`	Controls the timeout for payload response. IPMC, after sending payload alert, expects OS to respond within this time. Used in graceful reboot and graceful shutdown features.
`Payload shutdown timeout`	Controls the payload shutdown timeout as described in the graceful reboot/shutdown specification. This is the time that an OS would take to shutdown its applications before IPMC shuts it down or resets the payload.
`Verbosity`	Controls the debug mode verbosity of the messages. Selectively, messages from/to a particular interface can be enabled/disabled. Used for debugging only.
`AMC power up timeout`	Controls the time it would take for the AMCs/ARTM to come up before the payload is brought to M4 state.
`IPMC Control`	Reflects IPMC's control bits. Currently two bits, Bit 0 and Bit 1 control LED2 and LED1 on the boards.
`Set SOL Fail Over Link Change Timeouts`	Controls the time for which IPMC waits to switch to second serial over LAN (SOL) link when primary link fails, and the time it waits to switch back to primary channel if the primary channel link comes back up.
`Get SOL Fail Over Link Change Timeouts`	Gets the time for which IPMC waits to switch to second serial over LAN (SOL) link when primary link fails.

If you want to reset these variables to their default, use the `Reset IPMC Configuration Variables` command.

Command	NetFn	Opcode
Reset IPMC Configuration Variables	0x2E	0x9A

Data Bytes

Type	Byte	Data Field
Request Data	1	00
	2	00
	3	6F or 2A (Sun legacy)
	4	Parameter to be set. 00 = Reset all parameters to default value. NOTE: Any custom configuration parameters will be erased.
Response Data	1	Completion Code: [00] OK [C1] Command not supported [CC] Invalid data in request [C8] Request data field length limit exceeded. (Refer to IPMI specification for more completion codes)
	2	00
	3	00
	4	6F or 2A (Sun legacy)

Get Shelf FRU Record Data

Using this Pigeon Point extension command, a specified range of bytes can be retrieved from any multirecord in the Shelf FRU Information. This command can be applied to any PICMG-defined or OEM-defined record from the Shelf FRU Information, including the Address Table Record (PICMG Record ID = 10h), Shelf Power Distribution Record (PICMG Record ID = 11h), etc.

The type of the record to be retrieved is specified by the Manufacturer IANA and manufacturer-specific record type. The 0-based record number can be used to distinguish different instances of the same type of record. (There may be several records of the same type in the Shelf FRU Information.).

Command	**NetFn**	**Opcode**
Get Shelf FRU Record Data	2Eh	1h

Data Bytes

Type	**Byte**	**Data Field**
Request Data	1	PPS IANA Low Byte. A value 0Ah shall be used.
	2	PPS IANA Middle Byte. A value 40h shall be used.
	3	PPS IANA High Byte. A value 00h shall be used.
	4	Record Manufacturer IANA Low Byte.
	5	Record Manufacturer IANA Middle Byte.
	6	Record Manufacturer IANA High Byte.
	7	Record Type.
	8	Record Number. This field specifies the number of the record to be accessed. The record numbers are 0-based.
	9	Offset. This field specifies the offset from the beginning of the record in bytes.
	10	Byte Count.This field specifies the number of bytes to be read.

Type	Byte	Data Field
Response Data	1	Completion Code: [00] OK [C1] Command not supported [CC] Invalid data in request (Refer to IPMI specification for more completion codes)
	2	PPS IANA Low Byte. A value 0Ah shall be used.
	3	PPS IANA Middle Byte. A value 40h shall be used.
	4	PPS IANA High Byte. A value 00h shall be used.
	5	Count read. Indicates the number of bytes in the Data field.
	6:N+5	Data. This variable length field contains data retrieved from the record. N is specified in the Count read byte.

For example, to retrieve the first 10 bytes of the first Address Table record in the Shelf FRU Information, use the following parameters:

```
Record Manufacturer IANA Low Byte = 5Ah (PICMG)
Record Manufacturer IANA Middle Byte = 31h (PICMG)
Record Manufacturer IANA High Byte = 00h (PICMG)
Record type = 10h (Address Table)
Record Number = 00h (first record)
Offset = 00h
Byte Count = Ah (10 bytes)
```

Set SOL Fail Over Link Change Timeouts

Use the following command to set the time for which IPMC waits to switch to second serial over LAN (SOL) link when primary link fails, and the time it waits to switch back to primary channel if the primary channel link comes back up. Wait times are useful to filter out the link up/down bounces.

Note – The default settings are recommended over user-customized settings.

Wait times are in seconds. For example, a number 10 (0xA) in Byte 4 means IPMC will wait 10 seconds before switching the link to secondary channel. And a number 15(0xf) means IPMC will wait for 15 seconds before switching back to primary channel once it comes back up.

Command	NetFn	Opcode
Set SOL Fail Over Link Change Timeouts	0x2E	0xE7

Data Bytes

Type	Byte	Data Field
Request Data	1	00
	2	00
	3	6F or 2A (Sun legacy)
	4	Primary Link down, fail-over wait time.
	5	Primary Link up, wait time to switch to primary.
Response Data	1	Completion Code: [00] OK [C1] Command not supported [CC] Invalid data in request (Refer to IPMI specification for more completion codes)
	2	00
	3	00
	4	6F or 2A (Sun legacy)

Get SOL Fail Over Link Change Timeouts

Use the following command to read the time for which IPMC waits to switch to second serial over LAN (SOL) link when primary link fails, and the time it waits to switch back to primary channel if the primary channel link comes back up.

Command	NetFn	Opcode
Get SOL Fail Over Link Change Timeouts	0x2E	0xE6

Data Bytes

Type	Byte	Data Field
Request Data	1	00
	2	00
	3	6F or 2A (Sun legacy)
Response Data	1	Completion Code: [00] OK [C1] Command not supported [CC] Invalid data in request (Refer to IPMI specification for more completion codes)
	2	00
	3	00
	4	6F or 2A (Sun legacy)
	5	Primary Link down, fail-over wait time.
	6	Primary Link up, wait time to switch to primary.

Get Version

Use the following command to read IPMC firmware version and standby CPLD version. Although this command returns IPMC firmware version with CPLD version, the primary reason for this command is to provide CPLD version for IPMC version. To obtain IPMC version, use the IPMI get device ID command.

Command	NetFn	Opcode
Get Version	0x2E	0x80

Data Bytes

Type	Byte	Data Field
Request Data	1	00
	2	00
	3	6F or 2A (Sun legacy)
Response Data	1	Completion Code: [00] OK [C1] Command not supported [CC] Invalid data in request (Refer to IPMI specification for more completion codes)
	2	00
	3	00
	4	6F or 2A (Sun legacy)
	5	CPLD version
	6	REV1 byte of IPMC firmware
	7	REV2 Byte of IPMC Firmware
	8	Bit 7 to Bit 1: Reserved Bit 8 to Bit 1: Reserved 1 => Test release. 0 => Regular release.
	9	Reserved for future use (ignore).
	A	Reserved for future use (ignore).

Note – IPMC version is read as low nibble of REV1, high nibble of REV2, and low nibble of REV2.

Set Thermal Trip

Use the following command to enable or disable the thermal trip that determines when a board shuts down because maximum temperature is reached.

Caution – Damage to boards and systems can occur if temperature thresholds are reached and shut down does not occur. Unless the operating situation warrants overriding the default, use the default value.

In extreme situations such as operating in a war zone, there may be a requirement by the user to override the maximum temperature thresholds. Referred to as "war zone mode," users can override thermal trip to keep boards and systems running even if they reach maximum temperature thresholds.

Command	NetFn	Opcode
Set Thermal Trip	0x2E	E5

Data Bytes

Type	Byte	Data Field
Request Data	1	00
	2	00
	3	6F or 2A (Sun legacy)
	4	Bit 7 to Bit 1: Reserved. Shall be set as 0. Bit 0: Thermal trip bit value: 1 => Enable thermal trip (Default) 0 => Disable thermal trip.
Response Data	1	Completion Code: [00] OK [C1] Command not supported [CC] Invalid data in request (Refer to IPMI specification for more completion codes)
	2	00
	3	00
	4	6F or 2A (Sun legacy)

Get Thermal Trip

Use the following command to read the current thermal trip value.

Command	NetFn	Opcode
Get Thermal Trip	0x2E	0xE4

Data Bytes

Type	Byte	Data Field
Request Data	1	00
	2	00
	3	6F or 2A (Sun legacy)
Response Data	1	Completion Code: [00] OK [C1] Command not supported [CC] Invalid data in request (Refer to IPMI specification for more completion codes)
	2	00
	3	00
	4	6F or 2A (Sun legacy)
	5	Thermal trip value: 1 => Thermal trip enabled. Board will shut down if maximum temperature occurs. 0 => Thermal trip disabled. Board will not shut down if maximum temperature occurs.

Subscribe for Event Notifications

This Pigeon Point extension command can be used by an RMCP client to subscribe or unsubscribe for event notification on the current RMCP session. This command should be targeted to the logical Shelf Manager address (20h). It either subscribes or unsubscribes for event notification on the current session, depending on the value of the parameter Flags. If a session is subscribed for notifications, each time a new entry is placed in the SEL, a notification in the form of an Add SEL Entry request is sent to the RMCP client over this session. The client should confirm the notification by sending an Add SEL Entry response, according to the normal IPMI rules. The subscription is automatically terminated when the corresponding session is closed.

Command	NetFn	Opcode
Subscribe for Event Notifications	2Eh	4h

Data Bytes

Type	Byte	Data Field
Request Data	1	PPS IANA Low Byte. A value 0Ah shall be used.
	2	PPS IANA Middle Byte. A value 40h shall be used.
	3	PPS IANA High Byte. A value 00h shall be used.
	4	Flags. A bit field that specifies the action [0]: 1b = subscribe for event notifications on the current session; 0b = unsubscribe [7:1] Reserved. Shall be set to 0.
Response Data	1	Completion Code: [00] OK [C1] Command not supported [CC] Invalid data in request (Refer to IPMI specification for more completion codes)
	2	SPPS IANA Low Byte. A value 0Ah shall be used.
	3	PPS IANA Middle Byte. A value 40h shall be used.
	4	PPS IANA High Byte. A value 00h shall be used.

Glossary

Knowledge of the following terms and acronyms is useful in the administration of Oracle's Sun Netra CT900 server.

A

ATCA (Advanced Telecom Computing Architecture) Also referred to as AdvancedTCA. A series of industry standard specifications for the next generation of carrier grade communications equipment. AdvancedTCA incorporates the latest trends in high speed interconnect technologies, next generation processors, and improved reliability, manageability and serviceability, resulting in a new blade (board) and chassis (shelf) form factor optimized for communications at the lowest cost due to standardization.

B

backup Shelf Management card Any Shelf Management card capable of assuming support for the shelf manager function.

Base channel A physical connection within the Base interface composed of up to four differential signal pairs. Each Base channel is the endpoint of a slot-to-slot connection within the base interface.

Base switch A switch that supports the Base interface. A Base switch provides 10/100/1000BASE-T packet switching services to all node boards installed in the shelf. In the Sun Netra CT900 server, the Base switches reside in physical slots 7 and 8 (logical slots 1 and 2) in the shelf and support connections to all node slots and boards. Boards that support the Fabric interface and Base interface are also referred to as "switches."

Base interface An interface that is used to support 10/100 or 1000BASE-T connections between node boards and switches in a shelf. Midplanes are required to support the Base interface by routing four different signal pairs between all node board slots and each switch slot (in the Sun Netra CT900 server, the Base switch slots are physical slots 7 and 8, logical slots 1 and 2).

D

data transport interface A collection of point-to-point interfaces and bused signals intended to provide interconnect among the payloads on switches and node boards.

Dual Star topology An interconnect fabric topology in which two switch resources provide redundant connections to all end points within the network. A pair of switches provide redundant interconnects between node boards.

E

Electronic Keying or E-Keying A protocol used to describe the compatibility between the Base interface, Fabric interface, update channel interface, and synchronization clocks connections of front boards.

ETSI European Telecommunications Standards Institute.

F

Fabric channel A Fabric channel is comprised of two rows of signal pairs for a total of eight signal pairs per channel. Thus, each connector supports up to five channels available for board to board connectivity. A channel may also be viewed as being comprised of four 2-pair ports.

Fabric interface A Zone 2 interface that provides 15 connections per board or slot, each comprising up to 8 differential signal pairs (channels) supporting connections with up to 15 other slots or boards. Midplanes can support the Fabric interface in a variety of configurations including Full Mesh and Dual Star topologies. Boards that support the Fabric interface can be configured as fabric node boards, fabric switches, or mesh-enabled boards. Board implementations of the Fabric interface are defined by the PICMG 3.x subsidiary specifications.

field-replaceable unit (FRU) From a service point of view, the smallest irreducible elements of a server. Examples of FRUs are disk drives, I/O cards, and power entry modules. Note that a server, with all of its cards and other components, is not a FRU. However, an empty server is a FRU.

frame A physical or logical entity that can contain one or more shelves. Also called a rack, or, if enclosed, a cabinet.

front board A board that conforms to PICMG 3.0 mechanicals (8U x280mm), including a PCB and a panel. A front board connects with the Zone 1 and Zone 2 midplane connectors. It can optionally connect with a Zone 3 midplane connector or directly to a rear transition module connector and is installed into the front position in the shelf.

Full channel A Fabric channel connection that uses all eight differential signal pairs between end-points.

Full Mesh topology A Full Mesh configuration that can be supported within the Fabric interface to provide one dedicated channel of connectivity between each pair of slots within a shelf. Full Mesh-configured midplanes are capable of supporting mesh-enabled boards or switches and node boards installed in a Dual Star arrangement.

H

hot-swap The connection and disconnection of peripherals or other components without interrupting system operation. This facility may have design implications for both hardware and software.

I

I^2C Inter-integrated circuit bus. A multi-master, 2-wire serial bus used as the basis for current IPMBs.

IPMB (Intelligent Platform Management Bus) The lowest level hardware management bus as described in the Intelligent Platform Management Bus Communications Protocol specification.

IPMB-0 hub A hub device that provides multiple radial IPMB-0 links to various FRUs in the system. For example, an IPMB-0 hub is present in an ShMM that has radial IPMB-0 links.

IPMB-0 link With radial topology, the physical IPMB-0 segment between an IPMB-0 segment between an IPMB-0 hub and a single FRU. Each IPMB-0 link on an IPMB-0 hub is usually associated with a separate IPMB-0 sensor. An IPMB-0 link can also connect in a bused topology to multiple FRUs.

IPM controller (IPMC) The portion of a FRU that interfaces to the ATCA IPMB-0 and represents that FRU and any device subsidiary to it.

IPMI (Intelligent Platform Management Interface) A specification and mechanism for providing inventory management, monitoring, logging, and control for elements of a computer system. As defined in Intelligent Platform Management Interface specification.

L

logic ground A shelf-wide electrical net used on boards and midplanes as a reference and return path for logic-level signals that are carried between boards.

M

Mesh Enabled board A board that provides connectivity to all other boards within the midplane. Mesh Enabled boards support the Fabric interface and can also support the Base interface. Mesh Enabled boards can use 2 to 15 Fabric interface channels (typically all 15 channels) to support direct connections to all other boards in the shelf. The number of channels supported dictate the maximum number of boards that can be connected to within a shelf. Mesh Enabled boards that do not use the Base interface can be installed in the lowest available logical slot. Mesh Enabled boards supporting the Base interface can be Base switches, in which case they can support Base channels 1 and 2 and can be installed into logical slots 3 to 16. Boards supporting the Base interface use Base channels 1 and 2 only to support 10/100/1000BASE-T Ethernet.

midplane The functional equivalent of a backplane. The midplane is secured to the back of the server. The CPU card, I/O cards, and storage devices connect to the midplane from the front, and the rear transition modules connect to the midplane from the rear.

N

NEBS (Network Equipment/Building System) A set of requirements for equipment installed in telecommunications control offices in the United States. These requirements cover personnel safety, protection of property, and operational continuity. NEBS testing involves subjecting equipment to various vibration stresses, fire, and other environmental and quality metrics. There are three levels of NEBS compliance, each a superset of the preceding. NEBS level 3, the highest level, certifies that a piece of equipment can be safely deployed in an "extreme environment." A telecommunications central office is considered an extreme environment.

The NEBS standards are maintained by Telcordia Technologies, Inc., formerly Bellcore.

node board A board intended for use in a star topology midplane that has connectivity to a switch within the midplane. Node boards can support either or both the Base interface and Fabric interface. Boards supporting the Fabric interface use Fabric channels 1 and 2. Boards supporting the Base interface use Base channels 1 and 2 only to support 10/100/1000BASE-T Ethernet.

node slot A slot in the midplane that supports only node boards. A node slot is not capable of supporting a switch, thus a node board can never occupy logical slots 1 and 2. Node slots apply only to midplanes designed to support star topologies. Node slots support both the Base interface and Fabric interface. Typically, a node slot supports two or four Fabric channels and Base channels 1 and 2. Each two channel node slots establish connections to logical slots 1 and 2, respectively. Four channel node slots establish connections to logical slots 1, 2, 3, and 4, respectively.

P

PCI (Peripheral Component Interconnect) A standard for connecting peripherals to a computer. It runs at 20 - 33 MHz and carries 32 bits at a time over a 124-pin connector or 64 bits over a 188-pin connector. An address is sent in one cycle followed by one word of data (or several in burst mode).

Technically, PCI is not a bus but a bridge or mezzanine. It includes buffers to decouple the CPU from relatively slow peripherals and allow them to operate asynchronously.

physical address An address that defines the physical slot location of a FRU. A physical address consists of a site type and site number.

PICMG (PCI Industrial Computer Manufacturers Group) A consortium of companies who develop open specifications for telecommunications and industrial computing applications, including the CompactPCI standard.

R

rear-access A configuration option for the Sun Netra CT900 server in which all of the cables come out from the back of the shelf.

rear transition module A card used only on the rear-access models of the Sun Netra CT900 server to extend the connectors to the back of the shelf.

Reliability, Availability, Serviceability (RAS) A hardware and software feature that implements or improves the reliability, availability and serviceability of a server.

S

shelf A collection of components that consists of the midplane, front boards, cooling devices, rear transition modules, and power entry modules. The shelf was historically known as a chassis.

shelf address A variable length, variable format descriptor of up to 20 bytes in length that provides a unique identifier for each shelf within a management domain.

shelf ground A safety ground and earth return that is connected to the frame and is available to all boards.

shelf manager The entity in the system that is responsible for managing the power, cooling, and interconnects (with Electronic Keying) in an AdvancedTCA shelf. The shelf manager (ShMM) also routes messages between the System Manager Interface and IPMB-0, provides interfaces to system repositories, and responds to event messages. Note that while Oracle refers to ShMM for the entire management system, other ATCA vendors might refer to as the ShMC.

ShMC (Shelf Management Controller) An IPMC that is also capable of supporting the functions required of the shelf manager.

SNMP Simple Network Management Protocol.

star topology A midplane topology having one or more hub slots providing connectivity among the supported node slots.

switch A board intended for use in a star topology midplane that provides connectivity to a number of node boards within the midplane. Switches can support either or both the Base interface and Fabric interface. Boards utilizing the Fabric interface typically provide switching resources to all 15 available Fabric channels. Switches supporting the Base interface are installed into logical slots 1 and 2 and use all 16 Base channels to provide 10/100/1000BASE-T Ethernet switching resources to up to 14 node boards and the other switch. One Base channel is assigned to support a connection to the Shelf Management card.

switch slot In a star topology midplane, switch slots must reside in logical slots 1 and 2. Switch slots support both the Base interface and Fabric interface. Switch slots located in logical sots 1 and 2 are capable of supporting both Base interface and Fabric interface switches. Logical slots 1 and 2 are always switch slots regardless of the fabric topology. These slots support up to 16 Base channels and up to 15 Fabric channels each.

system A managed entity that can include one or more of the following components: node and switches, shelves, and frames.

U

U A unit of measure equal to 1.75 in. (44.45 mm).

update channel interface Also referred to as the update channel. A Zone 2 interface that provides connections comprising of ten differential signal pairs between two boards. This direct connection between two boards can be used to synchronize state information. The transport implemented for the update channel on a board is not defined. Update channels can be used only by two like-function boards created by a single vendor. Electronic Keying is used to ensure that update channel end points have matching transport protocols mapped prior to enabling the drivers. Midplanes must support the update channel. Boards can support the update channel.

Z

Zone 1 The linear space along the height dimension of an ATCA slot that is allocated for power, management, and other ancillary functions.

Zone 2 The linear space along the height dimension of an ATCA slot that is allocated to the data transport interface.

Zone 3 The linear space along the height dimension of an ATCA slot that is reserved for user-defined connections and/or interconnections to the rear transition modules for rear access systems.

READ AND DELETE

1. To display the “Beta Draft” footer , show the BetaDraft conditional setting.

2. After Beta, hide the BetaDraft conditional setting.

Index

T

U

W