PIGEON POINT SHELF MANAGER

External Interface Reference

Release 3.2.0
May 31, 2013

PIGEON POINT SYSTEMS

Pigeon Point Shelf Manager, ShMM-500 and ShMM-1500

The Pigeon Point Shelf Manager uses an implementation of the MD5 Message-Digest algorithm that is derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm.

Table of Contents

Tables

1 About This Document

This document describes how to use the Shelf Manager command line interface, the web interface, the Simple Network Management Protocol (SNMP) interface and the Remote Management Control Protocol (RMCP) interface.

1.1 Shelf Manager Documentation

This document is one of two in the Shelf Manager documentation set. These documents are available in PDF file format. The complete set includes:

Table 1 Shelf Manager Documentation

DOCUMENT	DESCRIPTION
• ***Pigeon Point Shelf Manager User Guide***	This document describes the overall configuration and use of the Pigeon Point Shelf Manager.
• ***Pigeon Point Shelf Manager External Interface Reference***	This document describes how to use the Shelf Manager command line interface, web interface, Simple Network Management Protocol (SNMP) interface and Remote Management Control Protocol (RMCP) interface.

1.2 Conventions Used in this Document

This table describes the textual conventions used in this document.

Table 2 Conventions Used in this Document

CONVENTION SAMPLE	DESCRIPTION
`setenv`	This 10 point bold Courier font is used for text entered at keyboard in example dialogues, which typically occur as one or more separate lines.
`ARMboot 1.0.2 (Apr 18 2003 - 14:58:54)`	This 10 point normal Courier font is used ShMM output in example dialogues.
`addmisc`	This 12 point bold Courier font is used for special text within normal paragraphs. The types of such special text include command names, file names, configuration parameters and command parameters, plus other text that could be entered by or displayed to a Shelf Manager user. This font is also used for command syntax definitions.
"Get Device ID"	IPMI commands defined by the IPMI specification or as PICMG extensions are shown in the normal font, surrounded by double quotes. This matches the corresponding convention used in PICMG specifications.

1.3 Additional Resources

For more information about Pigeon Point products, go to the Pigeon Point Web site: http://www.pigeonpoint.com/products.html.

2 Introduction

The Pigeon Point Shelf Manager external interfaces include support for a command line interface, a web interface, a Simple Network Management Protocol (SNMP) interface and a Remote Management Control Protocol (RMCP) interface. The following chapters describe how to use each of these interfaces.

The Pigeon Point Shelf Manager User Guideprovides an introduction to shelf management, the Shelf Manager and the Shelf Management Mezzanine (ShMM) on which the Shelf Manager runs; familiarity with that introduction is assumed in this document.

In this document, references to ShMM cover both ShMM-500 and ShMM-500R; the latter model complies with the Restriction of Hazardous Substances (RoHS) directive, but is software equivalent with its non-RoHS predecessor from a Shelf Manager perspective. References to ShMM also cover ShMM-1500R; the Shelf Manager running on ShMM-1500R is interface-compatible with the Shelf Manager running on ShMM-500, though it is implemented by a different binary executable.

3 Command Line Interface

The Command Line Interface (CLI) is used to communicate with the intelligent management controllers of the shelf, with boards, and with the Shelf Manager itself, via textual 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 **`telnet`** or the ShMM's 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.

3.1 Starting the Command Line Interface

To use the CLI, a user should first log on to the Linux system on which the Shelf Manager (ShM) runs. Once logged in, a user runs the executable **`clia`** (for Command Line Interface Agent) 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 ShMM. 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

Pigeon Point Shelf Manager Command Line Interpreter

20: Entity: (0xf0, 1) Maximum FRU device ID: 0x10
    PICMG Version 2.3
    Hot Swap State: M4, Previous: M3, Last State Change Cause: Normal
State Change (0)

#
```

If it is 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: (0xf0, 1) Maximum FRU device ID: 0x10
    PICMG Version 2.3
    Hot Swap State: M4, Previous: M3, Last State Change Cause: Normal
State Change (0)

CLI> exit
#
```

3.2 Command Line Interface Summary Table

The command line interface implements the commands shown in the following table, with a designated subset of them available for use on the backup Shelf Manager.

The commands are described in detail in the subsequent subsections in alphabetical order of the command names.

Table 3 Supported Commands

COMMAND	PARAMETERS	DESCRIPTION	USEABLE ON BACKUP SHELF MANAGER
activate	IPMB address FRU device ID	Activates the specified FRU.	No
airfilterreplaced	dd.mm.yyyy (optional)	Sets the date when the Air Filter is to be replaced.	No
alarm	alarm type	Activates or clears TELCO alarms.	No
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.	No
axie	subcommand, with its parameters	Shows AXIe-related information.	No
board	slot number (optional)	Shows information about boards.	No
boardreset	slot number	Resets the specified board.	No
busres	subcommand, with its parameters	Performs the specified operation on the Bused E-Keying-managed resources.	No
deactivate	IPMB address FRU device ID	Deactivates the specified FRU.	No
debuglevel	new debug level (optional)	Gets current debug levels for the Shelf Manager (for both the system log and the console) or sets new debug levels.	Yes
dhcp	subcommand	Manages the DHCP client embedded in the Shelf Manager.	No

COMMAND	PARAMETERS	DESCRIPTION	USEABLE ON BACKUP SHELF MANAGER
`exit/quit`		Exits from the interpreter in interactive mode.	Yes
`fans`	IPMB address (optional) FRU device ID (optional)	Shows information about fans.	No
`firewall`	subcommand, with its optional parameters	Shows firmware firewall state, enabled/disabled state, support and configurability for IPMI commands and functions, starts and stops firmware firewall.	No
`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.	Yes; on the backup Shelf Manager, reports information only about FRUs that are local to that backup.
`frucontrol`	IPMB address FRU device ID Option	Sends “FRU Control” command to specific FRU.	Yes
`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.	Yes; on the backup Shelf Manager, reports information only about FRUs that are local to that backup.
`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.	Yes; on the backup Shelf Manager, reports information only about FRUs that are local to that backup.
`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.	Yes; on the backup Shelf Manager, reports information only about FRUs that are local to that backup.

COMMAND	PARAMETERS	DESCRIPTION	USEABLE ON BACKUP SHELF MANAGER
`fruinfo`	IPMB address FRU device ID	Provides user friendly FRU Information output.	Yes; on the backup Shelf Manager, reports information only about FRUs that are local to that backup.
`getbootdev`	IPMB address FRU device ID or AMC address	Shows system boot device parameter.	No
`getfanlevel`	IPMB address (optional) FRU device ID (optional)	Shows the current level of the fan controlled by the specified FRU.	No
`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).	No
`getfruledstate`	IPMB address (optional) FRU device ID (optional) LED ID or ALL(optional)	Shows the FRU LED state.	Yes; on the backup Shelf Manager, reports information only about FRU LEDs that are local to that backup.

COMMAND	PARAMETERS	DESCRIPTION	USEABLE ON BACKUP SHELF MANAGER
`gethysteresis`	IPMB address (optional) sensor name (optional) sensor number (optional)	Shows both the positive and negative hysteresis of the specified sensor.	Yes; on the backup Shelf Manager, reports information only about sensors that are local to that backup.
`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 IPM controller is an IPMB hub, information about a specific link is shown.	Yes; on the backup Shelf Manager, reports the current state of IPMB-0 links that are local to that backup.
`getlanconfig`	channel number parameter name or number (optional) set selector (optional)	Shows a LAN configuration parameter for a specific channel.	No
`getpefconfig`	parameter name or number (optional) set selector (optional)	Shows a PEF configuration parameter.	No
`getsensoreventenable`	IPMB address (optional) sensor name (optional) sensor number (optional)	Shows the current sensor event mask values for the supported events of the specified sensor(s).	Yes; on the backup Shelf Manager, reports information only about sensors that are local to that backup.
`getthreshold`, `threshold`	IPMB address (optional) sensor name (optional) sensor number (optional)	Shows threshold information about a specific sensor.	Yes; on the backup Shelf Manager, reports information only about sensors that are local to that backup.
`help`		Shows the list of supported commands.	Yes

COMMAND	PARAMETERS	DESCRIPTION	USEABLE ON BACKUP SHELF MANAGER
ipmc	IPMB address (optional)	Shows information about one or all IPM controllers in the shelf.	Yes; on the backup Shelf Manager, reports information only about IPM Controllers that are local to that backup.
localaddress		Retrieves the IPMB address of the current Shelf Manager.	Yes
minfanlevel	fan level (optional)	Shows or sets the minimum fan level.	No
networkelementid	Index (optional) Network Element Identifier (optional)	Allows getting or setting the Network Element Identifier	No
poll		Initiates a poll of the IPM controllers on IPMB-0.	No
sel	IPMB address (optional) number of items (optional)	Shows the most recent items from the System Event Log maintained on the target IPM controller.	No
sendamc	IPMB address AMC address or FRU ID LUN (optional) Network function 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.	Yes
sendcmd	IPMB address LUN (optional) Network function Command Code Byte1 (optional) … (optional) ByteN (optional)	Transparently sends an arbitrary IPMI command to the target IPMC.	Yes

COMMAND	PARAMETERS	DESCRIPTION	USEABLE ON BACKUP SHELF MANAGER
`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.	Yes; on the backup Shelf Manager, reports information only about sensors that are local to that backup.
`sensordata`	IPMB address (optional) sensor name (optional) sensor number (optional)	Shows value information for a specific sensor.	Yes; on the backup Shelf Manager, reports information only about sensors that are local to that backup.
`sensorread`	IPMB address sensor number	Shows raw value information for a specific sensor (ignoring any Sensor Data Record describing the sensor). It does not check the presence of the target IPM controllers or validity of the sensor number, but just sends the request directly via IPMB.	Yes; on the backup Shelf Manager, reports information only about sensors that are local to that backup.
`session`		Shows information about active RMCP sessions.	No
`setbootdev`	IPMB address FRU device ID or AMC address boot device parameter	Sets system boot device parameter.	No
`setcommandpolicy`	action network function command code, channel (optional) LUN (optional)	Enables/disables (via the action parameter) a specific command for execution, using the IPMI 2.0 firmware firewall functionality	No
`setextracted`	IPMB address FRU device ID	Notifies the Shelf Manager that the specified FRU has been physically extracted from the shelf.	No

COMMAND	PARAMETERS	DESCRIPTION	USEABLE ON BACKUP SHELF MANAGER
`setfanpolicy`	IPMB address FRU device ID action to be taken: ENABLE or DISABLE timeout (optional) site type (optional) site number (optional)	Enables or disables Shelf Manager control over fan trays for cooling management purposes.	No
`setfanlevel`	IPMB address FRU device ID level	Sets a new level for the fan controlled by the specified FRU.	No
`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.	Yes; on the backup Shelf Manager, reports information only about FRU LEDs that are local to that backup.
`setfunctionpolicy`	mask network function command code channel (optional) LUN (optional)	Enables/disables specific subfunctions of a specific command for execution, using the IPMI 2.0 firmware firewall functionality	No
`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.	Yes; on the backup Shelf Manager, reports information only about sensors that are local to that backup.
`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.	Yes; on the backup Shelf Manager, reports information only about IPMB-0 links that are local to that backup.

Command	Parameters	Description	Useable on Backup Shelf Manager
`setlanconfig`	channel parameter name or number additional parameters	Sets the value of the LAN configuration parameter on the specified channel.	No
`setlocked`	IPMB address FRU device ID State	Sets the Locked bit for the specified FRU to the specified state (0 – unlock, 1 – lock).	Yes; on the backup Shelf Manager, reports information only about FRUs that are local to that backup.
`setpefconfig`	parameter name or number set selector (optional) parameter value	Sets a new value of a PEF configuration parameter.	No
`setpowerlevel`	IPMB address FRU device ID Power level Copy flag (optional)	Sets the power level of a board/FRU.	No
`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.	Yes; on the backup Shelf Manager, reports information only about sensors that are local to that backup.
`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.	Yes; on the backup Shelf Manager, reports information only about sensors that are local to that backup.

Command	Parameters	Description	Useable on Backup Shelf Manager
`Shelf`	subcommand, with its parameters	Shows general information about the shelf; several subcommands allow setting shelf attributes and getting additional information about specific areas.	No
`shelfaddress`	Shelf Address string (optional)	Gets or sets the Shelf Address field of the Address Table within Shelf FRU Information.	No
`shmstatus`		Shows the Shelf Manager active/backup status	Yes
`showunhealthy`		Shows the unhealthy components of the shelf	No
`switchover`		Initiates a switchover to the backup Shelf Manager.	Yes
`terminate`		Terminates the Shelf Manager, optionally without rebooting the ShMM.	Yes
`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.	No
`version`		Shows the Shelf Manager version information.	Yes

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`".

For more information about documentation conventions, see Conventions Used in this Document.

3.3 Shorthand Notations

The next chapters provide the details of the individual commands of the CLI and the syntax and usage of each of the available commands. The CLI supports both AdvancedTCA and CompactPCI 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 solely on an IPMB address and numerical FRU identifier:

- **`board <N>`**
- **`power_supply <N>`**
- **`fan_tray <N>`**
- **`pem <N>`**
- **`<IPMB-address> amc <M>`**
- **`board <N> amc <M>`**

In all the above convenience notations, **`<N>`** and **`<M>`** are Site Numbers of the component, as described in the Address Table for the shelf. Site Type 00h ("PICMG Board") corresponds to **`board`**, 01h ("Power Entry") – to **`pem`** and 04h ("Fan Tray") – to **`fan_tray`**. In both ATCA and CompactPCI systems, the OEM-defined site type C5h ("CompactPCI Power Supply") corresponds to power_supply.

This notation enables a user to designate a specific AMC by its number, using the syntax **`<IPMB-address> amc <M>`** or **`board <N> amc <M>`**. In that case, **`<IPMB-address>`** or **`board <N>`** identifies the corresponding AMC carrier board.

Revision 2.0 of the AMC.0 specification defines how the AMC slots are numbered, according to the following principles:

- AMC slot numbers 1-4 (which are also referenced as A1-A4, according to the specification) are only present on AMC carriers that support two layers of AMC slots (each of which can hold two compact size AMCs). These slots are actually embedded through the main board of the carrier, which is therefore called a "cutaway" carrier.
- AMC slot numbers 5-8 (which are also referenced as B1-B4, according to the specification) usually refer to either mid-size or full-size slots, but on cutaway carriers that support two layers of slots, these slots are "above" the A slots (that is, farther away from the main board of the carrier).
- For each layer, slot numbering starts from the slot that is closest to the Zone 3 end of the board (at the top of a vertical slot). The first layer A slot is numbered 1 or A1 and the first layer B slot is numbered 5 or B1.

Furthermore, the CLI supports the following abbreviations:

- **`board <N>`** can be abbreviated to **`b <N>`**
- **`power_supply <N>`** can be abbreviated to **`ps <N>`**

- **`fan_tray <N>`** can be abbreviated to **`ft <N>`**

The special abbreviations **`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 Information. **`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 CLI commands are supported by the backup Shelf Manager. Table 3 in Section 3.2 provides a list of all the CLI commands, including identification of which commands are supported by the backup Shelf Manager.

3.4 activate

3.4.1 Syntax

```
activate <IPMB-address> <fru_id>
activate board <N>
activate shm <N>
activate fan_tray <N>
activate power_supply <N>
activate pem <N>
activate <IPMB-address> amc <M>
activate board <N> amc <M>
```

3.4.2 Purpose

This command activates the specified FRU. To achieve that, it clears the “Activation Locked” flag on the target FRU by sending the IPMI command “Set FRU Activation Policy (Clear Locked)”, and then sends the IPMI command “Set FRU Activation (Activate FRU)” to the target FRU.

The first step allows the command to activate FRUs that are in the state M1. To allow the FRU some time to transition from the state M1 to the state M2, if the command “Set FRU Activation” returns the completion code “COMMAND NOT SUPPORTED IN PRESENT STATE”, it is repeated up to the number of times indicated by the value of the Shelf Manager configuration parameter **TASKLET_RETRIES**.

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 PICMG 2.x contexts, the Shelf Manager emulates this command in the best possible way for each specific type of FRU.

In the PICMG 3.0 context, this command is primarily useful for those FRUs that are not listed in the power management table in the Shelf FRU Information, or for which the Shelf Manager Controlled Activation attribute is set to **FALSE**.

These FRUs are not automatically activated by the Shelf Manager and stay in the state M2. 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.

3.4.3 Examples

Activate the IPM controller proper at address 9Ch.

```
# clia activate 9c 0

Pigeon Point Shelf Manager Command Line Interpreter

    Command executed successfully

#
```

3.5 airfilterreplaced

3.5.1 Syntax

airfilterreplaced [<dd.mm.yyyy>]

3.5.2 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 Filter 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 (i.e. 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.

3.5.3 Examples

```
# clia airfilterreplaced 25.12.2006

Pigeon Point Shelf Manager Command Line Interpreter
#
```

3.6 alarm

3.6.1 Syntax

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

3.6.2 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; the critical alarm output cannot be cleared. The action **`info`** displays information about the last alarm that occurred in the shelf.

Command invocation without parameters returns the status of the TELCO alarm outputs.

3.6.3 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 2006
        Alarm source: Remote request
        Alarm reason: On-demand setting alarms mask: 0x02
#
```

The following example assumes that the last alarm setting was done by the Platform Event Filter (PEF); the command reports that, along with details of the event that triggered the action by PEF.

```
# clia alarm info

Pigeon Point Shelf Manager Command Line Interpreter

Last saved alarm information:
        Alarm mask: 0x06
        Alarm date/time: Sat May  2 23:44:38 2009
        Alarm source: PEF
        Alarm reason: event from: (0x10,0,0); sensor: (0x02,1); event
data: 0x57, 0x18, 0x50
#
```

3.7 amcportstate

3.7.1 Syntax

amcportstate [-v] <IPMB-address> [<fru_id> | amc <M>]
amcportstate [-v] shm <N> [amc <M>]
amcportstate [-v] board <N> amc <M>]

3.7.2 Purpose

This command shows AMC port state information for a specific AMC. If a FRU ID or AMC number is omitted and an **<IPMB-address>** argument is specified, the AMC port state information is reported for all active AMCs for the designated IPM controller. If a FRU ID or AMC number is omitted and either a **shm <N>** or **board <N>** argument is specified, the AMC port state information is reported for AMC 0 on the designated IPM controller.

3.7.3 Example

```
# 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)
```

3.8 axie

3.8.1 Syntax

`axie <subcommand>`

The following subcommands are supported:

`sequencing`

3.8.2 Purpose

This command shows AXIe-related information. Currently the only supported subcommand is **`sequencing`** that shows information about the AXIe sequencing state machine.

3.8.3 sequencing

3.8.3.1 Syntax

`axie sequencing`

3.8.3.2 Purpose

This command displays information about the current state of the AXIe sequencing state machine, the current AXIe power monitor state of the shelf, and the IPMB addresses of current PCI Express (PCIe) hosts.

The command takes no parameters. The IPMI sensors associated with this facility (such as “all modules are in M4”) are defined on a shelf-specific basis by the HPDL (Hardware Platform Description Language) for the shelf.

The states for the AXIe sequencing state machine are defined as follows:

Table 4 AXIe Sequencing States

STATE	DESCRIPTION
`0 = Initial`	Initial state; the Shelf Manager is in this state after the initial power up or a power cycle caused by the Power Switch. On entry to this state, the Shelf Manager turns on the “Enumeration Not Ready” signal, turns off the “All modules are in M4” signal, sets to “deasserted” the “all Modules are in M4” sensor for the System Manager. In this state, AXIe modules are not powered but can be E-Keyed; the system module is not E-Keyed in this state.
`1 = E-Keying System Module`	The system module is being E-Keyed. The Shelf Manager transitions to this state after all other modules are E-Keyed and then proceeds to E-Key the system module.

State	Description
`2 = Power On_AXIe FRUs`	All AXIe modules are being powered (except M9536[1] modules E-Keyed as PCIe hosts). The Shelf Manager transitions to this state after the system module is E-Keyed.
`3 = Wait for AXIe FRUs in M4`	The Shelf Manager waits for all modules (except M9536 modules E-Keyed as PCIe hosts) to achieve state M4. The Shelf Manager transitions to this state after all modules (except M9536 modules E-Keyed as PCIe hosts) are powered. The duration of this state is controlled by a user-configurable timeout; after the expiration of the timeout, the Shelf Manager transitions to the next state even if the requisite modules are not in M4.
`4 = Startup Delay`	The Shelf Manager turns off the “Enumeration Not Ready” output signal, turns on the “All modules are in M4” signal, sets to “asserted” the “all Modules are in M4” sensor for the System Manager. After that, the Shelf Manager stays in this state for the number of milliseconds indicated by the “startup delay” configuration parameter.
`5 = Wait for Enumeration Ready`	The Shelf Manager waits for the “Enumeration Not Ready” input signal to be deasserted. The duration of this state is controlled by a user-configurable timeout; after the expiration of the timeout, the Shelf Manager transitions to the next state even if that input signal has not been deasserted.
`6 = Wait for System Manager Ready`	The Shelf Manager waits for the System Manager to issue the PPS-specific command “AXIe Allow Enable PCIe Hosts”. The duration of this state is controlled by a user-configurable timeout; after the expiration of the timeout, the Shelf Manager transitions to the next state even if that command has not been received.
`7 = Sending Set PCIe Host State`	The Shelf Manager sends the AXIe-specific command “AXIe Set PCIe Host State” to every module that has been E-Keyed as a PCIe host.
`8 = Power on M9536 Hosts`	The Shelf Manager powers on the M9536 modules that have been E-Keyed as PCIe hosts.
`9 = Finished`	Final state; in this state, power sequencing has been completed.

The AXIe power monitor states are defined as follows:

[1] “M9536 module” refers to the Agilent M9536A AXIe Embedded Controller.

Table 5 AXIe Power Monitor States

STATE	DESCRIPTION
0 = Power Up Requested	The shelf has received a power up request. This may be the initial power up or power up request after a power down. The shelf stays in this state during initial power up and normal operation.
1 = Power Down Requested	This is a transitional state, indicating that a hardware request to power down the shelf has been received and the shelf is currently powering down, starting with the PCIe hosts.
2 = Host Power Down Complete	This is a transitional state, indicating that all PCIe hosts have been powered down and now the Shelf Manager is powering down the rest of the FRUs.
3 = Power Down Complete	This state indicates that all FRUs have been powered down as a result of a previous power down request. The Shelf Manager is now in the idle state waiting for a possible power up request.

3.8.3.3 Examples

Get information about the current AXIe sequencing, power monitor states and PCI Express hosts, if any.

```
# clia axie sequencing

Pigeon Point Shelf Manager Command Line Interpreter

AXIe Sequencing State: 5 (Wait for Enumeration Ready)
AXIe Power Monitor State: 0 (Power Up Requested)
PCIe Host(s): 82
#
```

3.9 board

3.9.1 Syntax

`board [-v] [<physical-slot-address>]`

3.9.2 Purpose

This command and the **`boardreset`** command are different from most of the rest of the command set in that they work with ATCA or CompactPCI boards and take as arguments physical slot numbers, instead of IPM controller addresses and FRU device IDs.

This makes them easier for the end user and allows their use in CompactPCI contexts, where boards may not include an IPM controller and therefore, are not easily addressable using the IPMB address – FRU device ID pair.

This command shows information about each IPM controller in the range of IPMB addresses allocated to CompactPCI/ATCA slots, and about each additional FRU represented by those controllers. This command is a short-hand version of the **`fru`** command and the information displayed is the same.

The range of IPMB addresses is 82h-A0h for PICMG 3.0 systems and B0h-ECh for PICMG 2.9 systems, where boards have IPM controllers on them. In generic PICMG 2.x systems, where boards do not necessarily have IPM controllers on them, there may be no IPM controller address or FRU device ID defined. In that case, only the **`board`** and **`boardreset`** commands among the CLI commands are applicable.

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, (which would only occur in a non-compliant ATCA shelf) the following mapping table (mapping of logical slot numbers) is used.

Table 6 Mapping Between ATCA Logical Slot Numbers and IPMB Addresses

Logical Slot Number	IPMB Address	Logical Slot Number	IPMB Address
1	82	9	92
2	84	10	94
3	86	11	96
4	88	12	98
5	8A	13	9A
6	8C	14	9C
7	8E	15	9E
8	90	16	AE

For PICMG 2.9 based systems, the following mapping between Slot Number and IPMB address is used. "Slot Number" refers to the PICMG 2.x concept of "Physical Slot Number".

Table 7 Mapping Between CompactPCI Physical Slot Numbers and IPMB Addresses

PHYSICAL SLOT NUMBER	IPMB ADDRESS	PHYSICAL SLOT NUMBER	IPMB ADDRESS
1	B0	16	D0
2	B2	17	D2
3	B4	18	D4
4	B6	19	D6
5	B8	20	D8
6	BA	21	DA
7	BE	22	DC
8	BC	23	DE
9	C0	24	E0
10	C2	25	E2
11	C4	26	E4
12	C6	27	E6
13	C8	28	E8
14	CC	29	EA
15	CE	30	EC

3.9.3 Example

Get standard information about all boards in the shelf (where only physical slots 1 and 14 happen to be occupied).

```
# clia board

Pigeon Point Shelf Manager Command Line Interpreter

Physical Slot # 1

82: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x08
    PICMG Version 2.2
    Hot Swap State: M4 (Active), Previous: M3 (Activation In Process),
Last State Change Cause: Normal State Change (0x0)

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: Pigeon Point 6

Physical Slot # 14

9c: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x08
    PICMG Version 2.2
    Hot Swap State: M4 (Active), Previous: M3 (Activation In Process),
Last State Change Cause: Normal State Change (0x0)
```

```
9c: 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: Pigeon Point 6

#
```

Get verbose information about a board in physical slot 14.

```
# clia board -v 14

Pigeon Point Shelf Manager Command Line Interpreter

Physical Slot # 14

9c: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x08
    PICMG Version 2.2
    Hot Swap State: M4 (Active), Previous: M3 (Activation In Process),
Last State Change Cause: Normal State Change (0x0)
    Device ID: 0x00, Revision: 0, Firmware: 1.01, IPMI ver 1.5
    Manufacturer ID: 00315a (PICMG), Product ID: 0000, Auxiliary Rev:
01ac1014
    Device ID String: Pigeon Point 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

9c: 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: Pigeon Point 6
    Site Type: 0x00, Site Number: 14
    Current Power Level: 0x01, Maximum Power Level: 0x01, Current Power
Allocation: 20.0 Watts

#
```

3.10 boardreset

3.10.1 Syntax

`boardreset <physical-slot-address>`

3.10.2 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, (which would only be true in a non-compliant ATCA shelf) the following mapping is used. FRU device ID is 0.

Table 8 Mapping PICMG 3.0 Logical Slot Numbers

Logical Slot Number	IPMB Address	Logical Slot Number	IPMB Address
1	82	9	92
2	84	10	94
3	86	11	96
4	88	12	98
5	8A	13	9A
6	8C	14	9C
7	8E	15	9E
8	90	16	A0

In PICMG 2.x contexts, the Shelf Manager uses the radial board reset signal line, if available. Otherwise, if the radial BD_SEL# line is available, the Shelf Manager uses that mechanism to power cycle the board. In generic PICMG 2.x systems, where boards do not have IPM controllers on them, there may be no direct association between the physical slot number and the IPM controller and FRU device ID pair. In that case, CLI commands other than **`board`** and **`boardreset`** are not applicable to boards in physical slots.

For PICMG 2.9 based systems, the following CompactPCI Peripheral address mapping table is used:

Table 9 Mapping Between CompactPCI Physical Slot Numbers and IPMB Addresses

Physical Slot Number	IPMB Address	Physical Slot Number	IPMB Address
1	B0	16	D0
2	B2	17	D2
3	B4	18	D4
4	B6	19	D6

Physical Slot Number	IPMB Address	Physical Slot Number	IPMB Address
5	B8	20	D8
6	BA	21	DA
7	BC	22	DC
8	BE	23	DE
9	C0	24	E0
10	C4	25	E2
11	C6	26	E4
12	C8	27	E6
13	CA	28	E8
14	CC	29	EA
15	CE	30	EC

3.10.3 Examples

Reset the board in physical slot 14 (IPMB address 9Ch, FRU 0).

```
# clia boardreset 14

Pigeon Point Shelf Manager Command Line Interpreter

Board 14 reset, status returned 0

#
```

3.11 busres

3.11.1 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>
```

3.11.2 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, as follows:

- **`0 | mtb1`** --Metallic Test Bus pair 1
- **`1 | mtb2`** --Metallic Test Bus pair 2
- **`2 | clk1`** --Synch Clock group 1
- **`3 | clk2`** --Synch Clock group 2

The following subsections describe the syntax of the several variations of the **`busres`** command.

3.11.3 info

3.11.3.1 Syntax

`busres info [<resource>]`

3.11.3.2 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 resource ID is either a 0-based resource number or a short resource name, as follows:

- **`0 | mtb1`** --Metallic Test Bus pair 1

- **1 | mtb2** --Metallic Test Bus pair 2
- **2 | clk1** --Synch Clock group 1
- **3 | clk2** --Synch Clock group 2

3.11.3.3 Examples

Get information about the state of Metalic 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

#
```

3.11.4 release | force <resource>

3.11.4.1 Syntax

busres release | force <resource>

3.11.4.2 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 is **busres force <resource>**, the “Bused Resource Control (Force)” command is sent. See section 3.7.3.4 of the PICMG 3.0 specification for a detailed description of these ATCA commands.

The parameter **<resource>** is the resource ID. The resource ID is either a 0-based resource number or a short resource name, as follows:

- **0 | mtb1** --Metallic Test Bus pair 1
- **1 | mtb2** --Metallic Test Bus pair 2
- **2 | clk1** --Synch Clock group 1
- **3 | clk2** --Synch Clock group 2

3.11.4.3 Examples

Force releasing Metalic Test Bus pair 2 by the current owner.

```
# clia busres force mtb2

Pigeon Point Shelf Manager Command Line Interpreter

Force operation succeeded

#
```

3.11.5 lock | unlock

3.11.5.1 Syntax

`busres lock | unlock <resource>`

3.11.5.2 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.

Please note that only the resources that are owned by some IPM Controller can be locked. Also, 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 resource ID is either a 0-based resource number or a short resource name, as follows:

- **`0 | mtb1`** --Metallic Test Bus pair 1
- **`1 | mtb2`** --Metallic Test Bus pair 2
- **`2 | clk1`** --Synch Clock group 1
- **`3 | clk2`** --Synch Clock group 2

3.11.5.3 Examples

Lock Synch Clock group 3.

```
# clia busres lock clk3

Pigeon Point Shelf Manager Command Line Interpreter

Lock operation succeeded

#
```

3.11.6 query

3.11.6.1 Syntax

`busres [-v] query <resource> [<target> [noupdate]]`

3.11.6.2 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 resource ID is either a 0-based resource number or a short resource name, as follows:

- **0 | mtb1** --Metallic Test Bus pair 1
- **1 | mtb2** --Metallic Test Bus pair 2
- **2 | clk1** --Synch Clock group 1
- **3 | clk2** --Synch Clock group 2

The parameter **<target>** specifies the IPM Controller to which the request will be sent. It can either be an IPMB address of the IPM Controller, or a symbolic name: **board <N>**, **fan_tray <N>**, or **power_supply <N>**, where **<N>** is the number of the board, fan tray, or power supply respectively, exactly as for **clia ipmc** command.

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.

3.11.6.3 Examples

Send query for Metalic Test Bus pair 1 to the IPM Controller with address 82h.
Don't 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

#
```

3.11.7 setowner

3.11.7.1 Syntax

busres setowner <resource> <target>

3.11.7.2 Purpose

Warning: This command is for experienced users. Use it with care and only when you know what you are doing!

This command directly sets the owner of the specified resource in the resource table. It doesn't send a "Bused Resource Control" command, 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 resource ID is either a 0-based resource number or a short resource name, as follows:

- **0 | mtb1** --Metallic Test Bus pair 1
- **1 | mtb2** --Metallic Test Bus pair 2
- **2 | clk1** --Synch Clock group 1
- **3 | clk2** --Synch Clock group 2

The parameter **<target>** specifies the IPM Controller that is set as the owner of the resource. It can either be an IPMB address of the IPM Controller, or a symbolic name: **board <N>**, **fan_tray <N>**, or **power_supply <N>**, where **<N>** is the number of the board, fan tray, or power supply respectively, exactly as for the **clia ipmc** command. Use 0 as the IPMB address to specify that the resource is not owned by any IPM Controller.

3.11.7.3 Examples

Set board 1 as the new owner for Metalic Test Bus pair 1.

```
# clia busres setowner mtb1 board 1

Pigeon Point Shelf Manager Command Line Interpreter

New owner is set successfully

#
```

3.11.8 sendbusfree

3.11.8.1 Syntax

busres sendbusfree <resource> <target>

3.11.8.2 Purpose

Warning: This command is for experienced users. Use it with care and only when you know what you are doing!

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 resource ID is either a 0-based resource number or a short resource name, as follows:

- **0 | mtb1** --Metallic Test Bus pair 1
- **1 | mtb2** --Metallic Test Bus pair 2
- **2 | clk1** --Synch Clock group 1
- **3 | clk2** --Synch Clock group 2

The parameter **<target>** specifies the IPM Controller, to which the request is sent. It can either be an IPMB address of the IPM Controller, or a symbolic name: **board <N>**, **fan_tray <N>**, or **power_supply <N>**, where **<N>** is the number of the board, fan tray, or power supply respectively, exactly as for **clia ipmc** command. Use 0 as the IPMB address to specify that the resource is not owned by any IPM Controller.

3.11.8.3 Examples

Send a Bus Free request for Metalic Test Bus pair 1 to the IPM Controller with address 82h.

```
# clia busres sendbusfree mtb1 0x82

Pigeon Point Shelf Manager Command Line Interpreter

IPMC rejected ownership of the resource

#
```

3.12 deactivate

3.12.1 Syntax

```
deactivate <IPMB-address> <fru_id>
deactivate board <N>
deactivate shm <N>
deactivate fan_tray <N>
deactivate power_supply <N>
deactivate pem <N>
deactivate <IPMB-address> amc <M>
deactivate board <N> amc <M>
```

3.12.2 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. In PICMG 2.x contexts, the Shelf Manager emulates this command in the best possible way for each specific type of FRU. Attempting to deactivate an already inactive FRU does nothing.

Note: Programmatic deactivation of the active Shelf Manager (such as via the command **`clia deactivate 0x20`**) does not affect the Shelf Manager functionality and does not cause a switchover to the other Shelf Manager. However, programmatic deactivation of the physical Shelf Manager IPM controller on the active Shelf Manager causes a switchover to the backup Shelf Manager if the configuration variable **`SWITCHOVER_ON_HANDLE_OPEN`** is set to **`TRUE`** (remembering that the default value of this parameter is carrier-dependent) and if the backup Shelf Manager is available for switchover. Please see the section 6.2.15 for more information on this topic.

3.12.3 Examples

Deactivate the IPM controller proper at address 9Ch.

```
# clia deactivate 9c 0

Pigeon Point Shelf Manager Command Line Interpreter

    Command executed successfully

# clia deactivate b 4 amc 1

Pigeon Point Shelf Manager Command Line Interpreter

    Command executed successfully
```

3.13 debuglevel

3.13.1 Syntax

`debuglevel [ <new-value> [<new-console-value>]]`

3.13.2 Purpose

This command shows the current debug levels for the Pigeon Point Shelf Manager (for both the system log and the console), or sets them to new values if new values are 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 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 is 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.

3.13.3 Examples

Get current debug levels, and then set both of them to 0x001F. Here, the command **`debuglevel`** works in a mode compatible with previous releases.

```
# 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 set to 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 mask to 0x0003 (errors and warnings only).

```
# clia debuglevel 7 3

Pigeon Point Shelf Manager Command Line Interpreter

Debug Mask is set to 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

#
```

3.14 dhcp

3.14.1 Syntax

dhcp restart
dhcp status

3.14.2 Purpose

This command is used to control the operation of the DHCP client in the Shelf Manager.

The subcommand **restart** can be used to restart the DHCP client and update the IP addresses and other network parameters of the Shelf Manager via DHCP. For example, this subcommand can be useful if the configuration of the DHCP Server has been changed.

The subcommand **status** allows the user to inspect the current status of the DHCP client. Status is reported separately for each of the two network adapters used by the Shelf Manager.

The output information for the **status** subcommand command is in the following format:

DHCP Client: <adapter-1>: <status>; <adapter-2>: <status>

Where **<adapter-1>** and **<adapter-2>** are the names of the first and the second network adapters used by the Shelf Manager, and **<status>** is one of the following:

- **"completed"** – the DHCP client completed the retrieving of LAN parameters, the new parameters have been applied;
- **"started"** – the DHCP client received a start request and is now in the process of retrieving the LAN parameters from the DHCP server;
- **"not started"** – the DHCP client has not been started yet;
- **"stopped"** – the DHCP client has been stopped;
- **"error occurred"** – an unexpected error has occurred in the interaction between the client and the DHCP server;
- **"not used"** - the DHCP client is not enabled in Shelf Manager configuration and not used for the given adapter.

3.14.3 Examples

This following command restarts the DHCP client:

```
# clia dhcp restart

Pigeon Point Shelf Manager Command Line Interpreter

    DHCP client restarted successfully.
#
```

This command shows current status of the DHCP client:

```
# clia dhcp status

Pigeon Point Shelf Manager Command Line Interpreter

DHCP Client: "eth0": completed; "eth1": completed
#
```

3.15 exit/quit

3.15.1 Syntax

exit | quit

3.15.2 Purpose

The command **exit** or **quit** exits the CLI interactive mode (which is entered by issuing **clia** without parameters). This command can also be issued on the backup Shelf Manager.

3.15.3 Examples

```
CLI> exit

#
```

3.16 fans

3.16.1 Syntax

```
fans [-v] [ <IPMB-address> [ <fru_id> ] ]
fans board <N>
fans shm <N>
fans power_supply <N>
fans pem <N>
fans fan_tray <N>
fans <IPMB-address> amc <M>
fans board <N> amc <M>
```

3.16.2 Purpose

This command shows information about the specified fan FRUs. If the 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
- Current Level (the pair of Override and Local Control levels if both are available)

3.16.3 Examples

Get fan information about all fan FRUs at IPMB address 20h.

```
# clia fans 20

Pigeon Point Shelf Manager Command Line Interpreter

20: FRU # 3
    Current Level: 3
    Minimum Speed Level: 0, Maximum Speed Level: 15
20: FRU # 5
    Current Level: 3
    Minimum Speed Level: 0, Maximum Speed Level: 15
20: FRU # 4
    Current Level: 3
    Minimum Speed Level: 0, Maximum Speed Level: 15
```

Get fan information for the fan residing at AMC 1 for AMC carrier at IPMB address 82h.

```
#clia fans 82 amc 1
```

```
Pigeon Point Shelf Manager Command Line Interpreter

     No known fans at FRU id 0x01 at controller 0x82
#
```

3.17 firewall

3.17.1 Syntax

```
firewall [info
[<channel>:[<LUN>:[<netfn>:[<command_code>]]]]]
firewall [info all]
firewall stop
firewall start
```

The **<channel>**, **<LUN>**, **<netfn>** and **<command_code>** arguments can be used in hexadecimal or decimal forms. Any of those arguments is treated as hexadecimal if it includes a 0x prefix or a hexadecimal letter (A, B, C, D, E, and F); otherwise, an argument is treated as a decimal value.

3.17.2 Purpose

This command can be used to inspect the IPMI 2.0 firmware firewall state, enabled/disabled state, specification and initial version information, plus configurability for IPMI commands and functions. Also, this command allows the user to start and stop the firewall functionality at run-time. This feature (**stop**/**start**) should be used carefully and is mainly for test and debug purposes. Initially, the firewall is started.

The output information for the each command is in the following format:

```
<Spec.Type> <Ver> <Rev> <Err> <C/N> <E/D> <CMD> <Name>
[<cfm> : <efm>]
```

where:

- **<Spec.Type>** - specification type; can be **IPMI**, **IPMB**, **ICMB**, etc., see the IPMI 2.0 specification;
- **<Ver>** - version of the specification when the command was initially supported;
- **<Rev>** - revision of the version when the command was initially supported;
- **<Err>** - errata version of the revision when the command was initially supported;
- **<C/N>** - a flag; **C** - the command is configurable, **N** - the command is not configurable;
- **<E/D>** - a flag; **E** - the command is enabled, **D** - the command is disabled;
- **<CMD>** - hexadecimal number of the command within the given Network Function;
- <**Name>** - name of the command in plain text (which does not necessarily correspond to the name used in the corresponding specification);
- **<cfm>** - 64-bit configurable functions mask (only for commands that have configurable subfunctions, as defined by the IPMI 2.0 specification);
- **<efm>** - 64-bit enabled functions mask (only for commands that have configurable subfunctions, as defined by the IPMI 2.0 specification).

If the command is issued without parameters or with the single parameter **`info`**, information about the overall firewall status and summary firewall information is shown. With the parameter **`info all`**, information about the status of all known commands is shown. If any of the arguments **`<channel>`**, **`<LUN>`**, **`<netfn>`** or **`<command_code>`** are specified, the output information is filtered so that only lines matching the specified values are shown.

3.17.3 Examples

This command with no arguments is equivalent to the command with just an **`info`** argument:

```
# clia firewall

Pigeon Point Shelf Manager Command Line Interpreter

Firewall is running
    Commands database: /var/nvdata/firewall_cmd, 5307 bytes
    Functions database: /var/nvdata/firewall_sfm, 4876 bytes
    Channels in database: 4
    LUNs in database: 4
    Network Functions in database: 128
#
```

Show current firewall status.

```
# clia firewall info

Pigeon Point Shelf Manager Command Line Interpreter

Firewall is running
    Commands database: /var/nvdata/firewall_cmd, 5307 bytes
    Functions database: /var/nvdata/firewall_sfm, 4876 bytes
    Channels in database: 4
    LUNs in database: 4
    Network Functions in database: 128
#
```

Show current firewall settings and status.

```
# clia firewall info all

Pigeon Point Shelf Manager Command Line Interpreter

Firewall is running
    Commands database: /var/nvdata/firewall_cmd, 5307 bytes
    Functions database: /var/nvdata/firewall_sfm, 4876 bytes
    Channels in database: 4
    LUNs in database: 4
    Network Functions in database: 128
LUN: 0
  NetFn: 0x00 Chassis
    IPMI 1.5 1.1 0.0 c e 0x00 Get Chassis Capabilities
    IPMI 1.5 1.1 0.0 c e 0x01 Get Chassis Status
    IPMI 1.5 1.1 0.0 c e 0x02 Chassis Control
[000000000000003F:000000000000003F]
    IPMI 1.5 1.1 0.0 c e 0x03 Chassis Reset
```

```
    IPMI 1.5 1.1 0.0 c e 0x04 Chassis Identify
[0000000000000001:0000000000000001]
    IPMI 1.5 1.1 0.0 c e 0x05 Set Chassis Capabilities
    IPMI 1.5 1.1 0.0 c e 0x06 Set Power Restore Policy
    IPMI 1.5 1.1 0.0 c e 0x07 Get System Restart Cause
    IPMI 1.5 1.1 0.0 c e 0x08 Set System Boot Options
[0000000000001FFF:0000000000001FFF]
    IPMI 1.5 1.1 0.0 c e 0x09 Get System Boot Options
    IPMI 1.5 1.1 0.0 c e 0x0F Get POH Counter
  NetFn: 0x02 Bridge
    IPMI 1.5 1.1 0.0 c e 0x00 Get Bridge State
    IPMI 1.5 1.1 0.0 c e 0x01 Set Bridge State
........
........
#
```

Show current firewall settings for channel 0 and LUN 1.

```
# clia firewall info 0:1

Pigeon Point Shelf Manager Command Line Interpreter

Firewall is running
    Commands database: /var/nvdata/firewall_cmd, 5307 bytes
    Functions database: /var/nvdata/firewall_sfm, 4876 bytes
    Channels in database: 4
    LUNs in database: 4
    Network Functions in database: 128
Channel: 0x00
  LUN: 1
  NetFn: 0x00 Chassis
    IPMI 1.5 1.1 0.0 c e 0x00 Get Chassis Capabilities
    IPMI 1.5 1.1 0.0 c e 0x01 Get Chassis Status
    IPMI 1.5 1.1 0.0 c e 0x02 Chassis Control
[000000000000003F:000000000000003F]
    IPMI 1.5 1.1 0.0 c e 0x03 Chassis Reset
    IPMI 1.5 1.1 0.0 c e 0x04 Chassis Identify
[0000000000000001:0000000000000001]
    IPMI 1.5 1.1 0.0 c e 0x05 Set Chassis Capabilities
    IPMI 1.5 1.1 0.0 c e 0x06 Set Power Restore Policy
    IPMI 1.5 1.1 0.0 c e 0x07 Get System Restart Cause
....
....
#
```

Show current firewall settings for channel 0, LUN 1 and NetFn 0x0A.

```
# clia firewall info 0:1:A

Pigeon Point Shelf Manager Command Line Interpreter

Firewall is running
    Commands database: /var/nvdata/firewall_cmd, 5307 bytes
    Functions database: /var/nvdata/firewall_sfm, 4876 bytes
    Channels in database: 4
    LUNs in database: 4
```

```
    Network Functions in database: 128
Channel: 0x00
  LUN: 1
  NetFn: 0x0A Storage
    IPMI 1.5 1.1 0.0 c e 0x10 Get FRU Inventory Area Info
    IPMI 1.5 1.1 0.0 c e 0x11 Read FRU Data
    IPMI 1.5 1.1 0.0 c e 0x12 Write FRU Data
    IPMI 1.5 1.1 0.0 c e 0x20 Get SDR Repository Info
    IPMI 1.5 1.1 0.0 c e 0x21 Get SDR Repository Allocation Info
    IPMI 1.5 1.1 0.0 c e 0x22 Reserve SDR Repository
    IPMI 1.5 1.1 0.0 c e 0x23 Get SDR
    IPMI 1.5 1.1 0.0 c e 0x24 Add SDR
    IPMI 1.5 1.1 0.0 c e 0x25 Partial Add SDR
    IPMI 1.5 1.1 0.0 c e 0x26 Delete SDR
    IPMI 1.5 1.1 0.0 c e 0x27 Clear SDR Repository
    IPMI 1.5 1.1 0.0 c e 0x28 Get SDR Repository Time
    IPMI 1.5 1.1 0.0 c e 0x29 Set SDR Repository Time
    IPMI 1.5 1.1 0.0 c e 0x2A Enter SDR Repository Update Mode
    IPMI 1.5 1.1 0.0 c e 0x2B Exit SDR Repository Update Mode
    IPMI 1.5 1.1 0.0 c e 0x2C Run Initialization Agent
    IPMI 1.5 1.1 0.0 c e 0x40 Get SEL Info
    IPMI 1.5 1.1 0.0 c e 0x41 Get SEL Allocation Info
    IPMI 1.5 1.1 0.0 c e 0x42 Reserve SEL
    IPMI 1.5 1.1 0.0 c e 0x43 Get SEL Entry
    IPMI 1.5 1.1 0.0 c e 0x44 Add SEL Entry
    IPMI 1.5 1.1 0.0 c e 0x45 Partial Add SEL Entry
    IPMI 1.5 1.1 0.0 c e 0x46 Delete SEL Entry
    IPMI 1.5 1.1 0.0 c e 0x47 Clear SEL
    IPMI 1.5 1.1 0.0 c e 0x48 Get SEL Time
    IPMI 1.5 1.1 0.0 c e 0x49 Set SEL Time
    IPMI 1.5 1.1 0.0 c e 0x5A Get Auxiliary Log Status
    IPMI 1.5 1.1 0.0 c e 0x5B Set Auxiliary Log Status
[0000000000000007:0000000000000007]
#
```

Show current firewall settings for channel 4, LUN 0, NetFn 0x0C and command 27 (treated as decimal).

```
# clia firewall info 2:0:0xC:27

Pigeon Point Shelf Manager Command Line Interpreter

Firewall is running
    Commands database: /var/nvdata/firewall_cmd, 5307 bytes
    Functions database: /var/nvdata/firewall_sfm, 4876 bytes
    Channels in database: 4
    LUNs in database: 4
    Network Functions in database: 128
Channel: 0x02
  LUN: 0
  NetFn: 0x0C Transport
    IPMI 1.5 1.1 0.0 c e 0x1B Get User Callback Options
#
```

Stop firewall.

```
# clia firewall stop

Pigeon Point Shelf Manager Command Line Interpreter

    Firewall has been stopped successfully...
#
```

Enable firewall.

```
# clia firewall start

Pigeon Point Shelf Manager Command Line Interpreter

    Firewall has been started successfully...
#
```

3.18 fru

3.18.1 Syntax

```
fru [-v] [<IPMB-address> [<fru_id> | id=<fru_id> |
type=<site_type>]] | [type=<site_type> [/<site_number>]]
fru board <N>
fru shm <N>
fru power_supply <N>
fru pem <N>
fru fan_tray <N>
fru <IPMB-address> amc <M>
fru board <N> amc <M>
```

3.18.2 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 ShMC
- 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 are reserved.

In CompactPCI systems, the following OEM-defined site types are used to describe CompactPCI sites:
- C4h = CompactPCI Board
- C5h = CompactPCI Power Supply

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 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 allocation in Watts

This command shows information about FRUs in state M1, if they were known previously to the Shelf Manager. This command can also be issued on the backup Shelf Manager; in that case, information is only reported about FRUs that are local to the backup Shelf Manager.

3.18.3 Examples

Get standard information about all FRUs at address 9Ch.

```
# 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: "Pigeon Point 6"

#
```

Get verbose information about all FRUs at address 9Ch.

```
# 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: "Pigeon Point 6"
    Site Type: 0x00, Site Number: 14
    Current Power Level: 0x01, Maximum Power Level: 0x01, Current Power
Allocation: 20.0 Watts

#
```

Get verbose information about FRU 1 at address 20h.

```
# clia fru -v 20 id=1

Pigeon Point Shelf Manager Command Line Interpreter

20: FRU # 1
    Entity: (0x1, 0x1)
    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: "Pigeon Point 1.1"
    Current Power Level: UNKNOWN, Maximum Power Level: UNKNOWN, Current
Power Allocation: UNKNOWN
```

Get information about AMC 1 on the AMC carrier in slot 8.

```
# clia fru board 8 amc 1

Pigeon Point Shelf Manager Command Line Interpreter

84: FRU # 1 (AMC # 1)
    Entity: (0xc1, 0x61)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In Process),
Last State Change Cause: Normal State Change (0x0)
    Device ID String: "AMC Module 1"

#
```

3.19 frudata

3.19.1 Syntax

frudata [<IPMB-address> [<fru_id> [<block_offset>]]]
frudata <IPMB-address> <fru_id> <byte_offset> <byte1> [byte2 … [byte16] …]

<IPMB-address> <fru_id> can be replaced with any of the following alternatives:
board <N>
shm <N>
power_supply <N>
pem <N>
fan_tray <N>
<IPMB-address> amc <M>
board <N> amc <M>

3.19.2 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. The changes will become fully effective only after the restart of the shelf.

This command can also be issued on the backup Shelf Manager; in that case, FRU Information is only displayed for FRUs that are local to the backup Shelf Manager.

3.19.3 Examples

```
# 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
    FRU Info size: 160
fe: FRU # 0 Raw FRU Info Data
    FRU Info size: 160
#
```

```
# 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
#

# clia frudata board 8 amc 1

Pigeon Point Shelf Manager Command Line Interpreter

84: FRU # 1 Raw FRU Info Data
    FRU Info size: 64
 01 00 00 00  00 01 00 FE  C0 02 06 41  F7 5A 31 00
 16 00 1E C0  82 28 20 76  5A 31 00 19  00 00 80 04
 E0 FF FF E1  FF FF E2 FF  FF E3 FF FF  00 51 00 00
 FC 01 51 00  00 FC 02 51  00 00 FC 03  51 00 00 FC
```

3.20 frudatar

3.20.1 Syntax

`frudatar <IPMB-address> <fru_id> <file name>`

`<IPMB-address> <fru_id>` can be replaced with any of the following alternatives:

`board <N>`
`shm <N>`
`power_supply <N>`
`pem <N>`
`fan_tray <N>`
`<IPMB-address> amc <M>`
`board <N> amc <M>`

3.20.2 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.

This command can also be issued on the backup Shelf Manager; in that case, FRU Information is only read from FRUs that are local to the backup Shelf Manager.

3.20.3 Examples

```
# 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
#
```

3.21 frudataw

3.21.1 Syntax

frudataw [-s|-d|-p|-l] <IPMB-address> <fru_id> <file name>
frudataw -s -c <IPMB-address> <fru_id>
frudataw -d -c <IPMB-address> <fru_id>
frudataw -p -c <IPMB-address> <fru_id>
frudataw -l -c <IPMB-address> <fru_id>

<IPMB-address> <fru_id> can be replaced with any the following alternatives:
board <N>
shm <N>
power_supply <N>
pem <N
fan_tray <N>
<IPMB-address> amc <M>
board <N> amc <M>

3.21.2 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. The changes will become fully effective only after the restart of the shelf.

This command can also be issued on the backup Shelf Manager; in that case, FRU Information is only downloaded to FRUs that are local to the backup Shelf Manager.

There are four special options to update HPDL data, Shelf Manager Configuration Parameters and Board Lan Configuration Parameters records in the FRU Information. The option **-d** indicates that HPDL data are to be updated; the option **-s** indicates that SDRs are to be updated; the option **-p** indicates that Shelf Manager Configuration Parameters records are to be updated; the option **-l** indicates that Board/AMC LAN Configuration Parameters records are to be updated. In all cases, the parameter **<file name>** specifies the path to the file that contains binary data to be stored in the target FRU information. The option **-c** (if specified instead of the file name) removes (clears) the specified records from the target FRU Information.

The parameters **<IPMB-address>** and **<fru_id>** specify the IPMB address and FRU device ID of the FRU Information to update. When updating records in the Shelf FRU Information, one can specify either the addresses of actual locations of the Shelf FRU Information or the alias (20h, 254) that refers to the logical Shelf FRU Information storage.

The file that contains binary HPDL data, SDRs, Shelf Manager Configuration or Board/AMC LAN Configuration records can be compressed using the **gzip** compression utility. The compressed data is then stored in the FRU Information. Decompression is performed by the Shelf Manager when reading from the FRU information. The Shelf Manager automatically detects whether the data is compressed or not.

3.21.3 Examples

```
# clia frudataw 20 2 /var/tmp/20.2.orig.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
#

# clia frudataw -d 20 2 /var/nvdata/chassis_data

Pigeon Point Shelf Manager Command Line Interpreter

20: FRU # 2 Reading FRU Info Data, size 8192
    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: 3072, status =
0(0x0)
    Writing 16 bytes to IPM 0x20, FRU # 2, offset: 3088, status =
0(0x0)
    Writing 16 bytes to IPM 0x20, FRU # 2, offset: 3104, status =
0(0x0)
    Writing 16 bytes to IPM 0x20, FRU # 2, offset: 3120, status =
0(0x0)
    Writing 8 bytes to IPM 0x20, FRU # 2, offset: 3136, status = 0(0x0)
    wrote 3144 (of 3144) bytes to the FRU 20#2 #

# clia frudataw -s -c 20 2

Pigeon Point Shelf Manager Command Line Interpreter

20: FRU # 2 Reading FRU Info Data, size 8192
    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: 1136, status =
0(0x0)
    Writing 16 bytes to IPM 0x20, FRU # 2, offset: 1152, status =
0(0x0)
    Writing 16 bytes to IPM 0x20, FRU # 2, offset: 1168, status =
0(0x0)
    Writing 16 bytes to IPM 0x20, FRU # 2, offset: 1184, status =
0(0x0)
    Writing 16 bytes to IPM 0x20, FRU # 2, offset: 1200, status =
0(0x0)
    Writing 13 bytes to IPM 0x20, FRU # 2, offset: 1216, status =
0(0x0)
    wrote 1229 (of 1229) bytes to the FRU 20#2 #
```

3.22 fruinfo

3.22.1 Syntax

fruinfo [-v] [-x] <IPMB-address> <fru_id>

<IPMB-address> <fru_id> can be replaced with any the following alternatives:
board <N>
shm <N>
power_supply <N>
pem <N>
fan_tray <N>
<IPMB-address> amc <M>
board <N> amc <M>

3.22.2 Purpose

This command shows FRU Information in a user-friendly format. This command can also be issued on the backup Shelf Manager; in that case, FRU Information is only shown for FRUs that are local to the backup Shelf Manager.

With the **-v** option, the command shows the contents of records in the Multi-record area in detail. Without this option, only the record headers are shown.

With the **-x** option, a hexadecimal dump of all FRU Information areas and records is shown along with the textual representation.

3.22.3 Examples

```
# 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
    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    =

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 C1 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

#
```

3.23 frucontrol

3.23.1 Syntax

```
frucontrol <IPMB-address> <fru_id> <option>
frucontrol board <N> <option>
frucontrol shm <N> <option>
frucontrol power_supply <N> <option>
frucontrol pem <N>
frucontrol fan_tray <N> <option>
frucontrol <IPMB-address> amc <M> <option>
frucontrol board <N> amc <M> <option>
```

3.23.2 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.

When the **<option>** parameter is **info**, the command “Get FRU Control Capabilities” is sent to the specified FRU. The returned byte indicates what FRU Control commands are supported by the specified FRU. This option only works for FRUs that support ECN-002 to the PICMG 3.0 R2.0 specification.

The parameter **<option>** specifies the option of 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
- **graceful_reboot** (abbreviated as **gr**) – perform graceful reboot of the FRU payload
- **diagnostic_interrupt** (abbreviated as **di**) – issue the diagnostic interrupt
- **info** – get FRU Control capabilities.

This command can also be issued on the backup Shelf Manager; in that case, the “FRU Control” command is only sent to FRUs that are local to the backup Shelf Manager.

According to the PICMG 3.0 specification, the command “FRU Control (Cold Reset)” must be implemented for all FRUs. Most FRUs that are represented by the Shelf Manager (such as fan trays or PEMs) have no payload to which the operations of this command would apply; for these FRUs the Shelf Manager implements the cold reset command as a no-operation that just returns the successful completion code.

3.23.3 Examples

Issue a "Cold Reset" command to the FRU 0 at IPMB address 9Ch.

```
# 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

#
```

Get FRU Control capabilities for the FRU 0 at IPMB address 0Eh.

```
# clia frucontrol 0e 0 info

Pigeon Point Shelf Manager Command Line Interpreter

    FRU Control Capabilities: Controller 0x0e, FRU ID # 0, status
0(0x0)
    Capabilities: 00; Supported commands: Cold Reset
#
#
```

3.24 getbootdev

3.24.1 Syntax

getbootdev <IPMB-0-address> [<fru_id> | <IPMB-L-address>]

3.24.2 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.

3.24.3 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)
```

3.25 getconfigparam

3.25.1 Syntax

getconfigparam [<parameter name>]

3.25.2 Purpose

This command shows the value of the specified Shelf Manager configuration parameter. If no name is specified, all known configuration parameters are shown, with their values.

Names of the configuration parameters are described in the Shelf Manager *User Guide*.

This command can be useful to see the actual values of configuration parameters if configuration parameters are set from multiple sources (multiple configuration files and/or the Shelf FRU Information).

3.25.3 Examples

Get the value of the configuration parameter **SENSOR_POLL_INTERVAL**.

```
# clia getconfigparam SENSOR_POLL_INTERVAL

Pigeon Point Shelf Manager Command Line Interpreter

SENSOR_POLL_INTERVAL = 1
#
```

Show all configuration parameters with their values.

```
# clia getconfigparam

Pigeon Point Shelf Manager Command Line Interpreter

2_X_SYSTEM = FALSE
ACTIVATE_LOCAL_WITHOUT_SHELF_FRU = FALSE
ALARM_CUTOFF_TIMEOUT = 600
ALLOW_ALL_COMMANDS_FROM_IPMB = FALSE
ALLOW_CHANGE_EVENT_RECEIVER = TRUE
ALLOW_CLEARING_CRITICAL_ALARM = FALSE
ALLOW_POWER_UNRELATED_FRU_IN_CRITICAL_STATE = FALSE
ALLOW_RESET_STANDALONE = FALSE
ALTERNATE_CONTROLLER = TRUE
ATCA_TESTER_COMPATIBILITY = FALSE
AUTO_SEND_MESSAGE = TRUE
BOARD_LAN_PARAMETERS_CHANNEL_LIST =
CARRIER = PPS
CARRIER_OPTIONS =
CONSOLE_LOGGING_ENABLED = TRUE
COOLING_FAN_DECREASE_TIMEOUT = 0
COOLING_FAN_INCREASE_TIMEOUT = 0
COOLING_IGNORE_LOCAL_CONTROL = FALSE
```

```
COOLING_KEEP_POWERED_OFF_FRUS_IN_M1 = FALSE
COOLING_MANAGEMENT =
COOLING_POLL_TIMEOUT = 30
CPLD_ACTIVE_WORKAROUND = FALSE
CTCA_FRU_RESET_TIMEOUT = 500
CTCA_HEALTHY_TIMEOUT = 0
CTCA_INITIAL_FAN_LEVEL = 5
DEFAULT_GATEWAY_IP_ADDRESS = 192.168.1.253
DEFAULT_GATEWAY_IP_ADDRESS2 = 0.0.0.0
DEFAULT_RMCP_IP_ADDRESS = 192.168.1.68
DEFAULT_RMCP_IP_ADDRESS2 = 0.0.0.0
DEFAULT_RMCP_NETMASK = 255.255.255.0
DEFAULT_RMCP_NETMASK2 = 0.0.0.0
DEFAULT_VLAN_ID = 0
DEFAULT_VLAN_ID2 = 0
DETECT_DEADLOCKS = TRUE
DEVICE_POLL_TIMEOUT = 15
DHCP_FOR_RMCP_ONLY = FALSE
DHCP_SERVER_ADDRESS = 0.0.0.0
ENABLE_DIRECT_SHELF_FRU_WRITE = FALSE
ENABLE_INTEGRALHPI = TRUE
EXIT_IF_HEALTHY_LOST_IN_STANDALONE_MODE = FALSE
EXIT_IF_NO_SHELF_FRU = FALSE
EXTERNAL_EVENT_HANDLER =
FAN_FULL_SPEED_DELAY = 0
FAN_LEVEL_STEP_DOWN = 1
FAN_LEVEL_STEP_UP = 1
HPDL = FALSE
HPDL_ON_SUBSIDIARY_FRUS = TRUE
IGNORE_FAILED_DIRECTED_POWER_DOWN = TRUE
INITIALIZATION_SCRIPT =
INITIAL_FAN_LEVEL = 5
INITIAL_SLOW_LINK_DELAY = 0
INNER_SEQUENCE_NUMBER_IN_SEND_MSG_RESPONSE = TRUE
IPMB_ADDRESS = 0
IPMB_LINK_ISOLATION_TIMEOUT = -1
IPMB_RETRIES = 3
IPMB_RETRY_TIMEOUT = 4
IPMB_RETRY_TIMEOUT_MSEC = 500
IPMC_PRESERVE_ON_REVISION_CHANGE = TRUE
ISOLATE_MUX_ADDRESS = 112
ISOLATE_MUX_IGNORE_COUNT = 10
ISOLATE_MUX_ON_GPIO8 = FALSE
LOCAL_SHELF_FRU = TRUE
M7_TIMEOUT = -1
MAX_ALERT_POLICIES = 64
MAX_ALERT_STRINGS = 64
MAX_DEFERRED_ALERTS = 32
MAX_EVENT_FILTERS = 64
MAX_EVENT_SUBSCRIBERS = 64
MAX_EVENT_SUBSCRIBER_IDLE_TIME = 60
MAX_INCOMING_IPMB_REQUESTS = 192
MAX_NODE_BUSY_TRANSMISSIONS = 255
MAX_OEM_FILTERS = 16
MAX_PENDING_EVENT_NOTIFICATIONS = 1024
MAX_PENDING_IPMB_REQUESTS = 192
MAX_SEL_ENTRIES = 1024
```

```
MAX_SESSIONS = 32
MAX_USERS = 32
MICRO_TCA = FALSE
MIN_FAN_LEVEL = 1
MIN_SHELF_FRUS = 1
NORMAL_STABLE_TIME = 3600
PEF_USE_KEYED_ALARMS = FALSE
PET_FORMAT = 0
POWER_UNLISTED_FRUS = TRUE
PROPAGATE_RMCP_ADDRESS = FALSE
REAPPLY_POWER_MAX_COOLING_STATE = NORMAL
REDUNDANCY_ENABLED = TRUE
REDUNDANCY_NET_ADAPTER = eth1
REDUNDANCY_NET_ADAPTER2 =
REDUNDANCY_NETMASK = 0.0.0.0
REDUNDANCY_PORT = 1040
REDUNDANT_IP_ADDRESS = 192.168.0.62
RESERVATION_RETRIES = 10
RMCP_NET_ADAPTER = eth0
RMCP_NET_ADAPTER2 =
RMCP_WITHOUT_SHELF_FRU = FALSE
SDR_READ_RETRIES = 3
SEL_FILE_COMPRESSION_ENABLED = TRUE
SEL_FILE_JOURNALING_ENABLED = TRUE
SEL_FILE_WRITE_DELAY = 3
SEL_HIGH_WATERMARK = 10
SEL_LOW_WATERMARK = 50
SENSOR_POLL_INTERVAL = 1
SESSION_SEQUENCE_WINDOW = 128
SHELF_FRU_IN_EEPROM = TRUE
SHELF_FRU_IPMB_SOURCE1 = 0
SHELF_FRU_IPMB_SOURCE2 = 0
SHELF_FRU_TIMEOUT = 20
SHELF_MANAGER_CONFIGURATION_IN_SHELF_FRU_INFO = FALSE
SHORT_SEND_MSG_RESPONSE = TRUE
SWAPPED_CROSS_CONNECTS = FALSE
SWITCHOVER_ON_HANDLE_OPEN = FALSE
SWITCHOVER_TIMEOUT_ON_BROKEN_LINK = 11
SYSLOG_LOGGING_ENABLED = TRUE
SYSTEM_MANAGER_TRUNCATES_SEL = FALSE
TACHOMETER_THRESHOLD_UPDATE_DELAY = 15
TASKLET_RETRIES = 3
TIMEPROTO =
TIMESERVER = 0.0.0.0
TURBO_MODE_MIN_FAN_FAILURES = 1
UNCONDITIONAL_SDR_REREAD_ON_VERSION_CHANGE = FALSE
USE_DHCP = FALSE
USE_SECOND_CHANNEL = FALSE
VERBOSITY = 7
VERBOSITY_CONSOLE = -1
VERIFY_SHELF_FRU_CHECKSUM = TRUE
WATCHDOG_ENABLED = TRUE
```

3.26 getfanlevel

3.26.1 Syntax

```
getfanlevel <IPMB-address> <fru_id>
getfanlevel shm <N>
getfanlevel board <N>
getfanlevel power_supply <N>
getfanlevel pem <N>
getfanlevel fan_tray <N>
getfanlevel <IPMB-address> amc <M>
getfanlevel board <N> amc <M>
```

3.26.2 Purpose

This command shows the current level of the fan controlled by the FRU specified in the command parameters.

3.26.3 Examples

Get fan level for the fan residing at FRU #2 at IPMB address 20h.

```
# clia getfanlevel 20 2

Pigeon Point Shelf Manager Command Line Interpreter

20: FRU # 2 Override Fan Level: 1, Local Fan Level: 255

#
```

3.27 getfanpolicy

3.27.1 Syntax

getfanpolicy [<fan tray addr> [<fan tray fru_id>]] [-s <addr>|<site_type> [<fru_id>|<site_number>]]

3.27.2 Purpose

This command 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 be issued with 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**.

3.27.3 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
```

3.28 getfruledstate

3.28.1 Syntax

```
getfruledstate [-v] [<IPMB-address> [<fru_id>
[<LedId>|ALL]]]
getfruledstate [-v] shm <N>[<LedId>|ALL]
getfruledstate [-v] board <N> [<LedId>|ALL]
getfruledstate [-v] power_supply <N>[<LedId>|ALL]
getfruledstate [-v] pem <N> [<LedId>|ALL]
getfruledstate [-v] fan_tray <N>[<LedId>|ALL]
getfruledstate [-v] <IPMB-address> amc <M> [<LedId>|ALL]
getfruledstate [-v] board <N> amc <M> [<LedId>|ALL]
```

3.28.2 Purpose

This command shows the current FRU LED state on all levels of control that are enabled for the LED(s).

In verbose mode, the Shelf Manager shows additional information about the colors supported by the LED(s). Also, if a LED Description record is present in the FRU Information of the target FRU, the Shelf Manager shows the legend, symbol and description information from that record for the target LED(s).

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.

This command can also be issued on the backup Shelf Manager; in that case, the FRU LED state is only shown for FRU LEDs that are local to the backup Shelf Manager.

3.28.3 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 LED state for the FRU 3 at IPMB address 20h:

```
# clia getfruledstate -v 20 3

Pigeon Point Shelf Manager Command Line Interpreter

20: FRU # 3, 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
    Legend: "H/S"
    Description: "Blue LED"

20: FRU # 3, 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
    Legend: "OOS"
    Description: "Out of Service LED"

20: FRU # 3, Led # 2 ("LED 2"):
    Local Control LED State: LED ON, color: GREEN
    LED's color capabilities:
        Colors supported(0x08): GREEN
        Default LED Color in Local Control State(0x03): GREEN
        Default LED Color in Override State(0x03): GREEN
    Symbol: "IEC-60417-5867"
    Description: "Green. Should not blink"

#
```

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 verbose 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
```

3.29 gethysteresis

3.29.1 Syntax

```
gethysteresis [<IPMB-address> [ [ <lun>: ] <sensor id> |
<sensor name>] ]
gethysteresis [board <N> [ [ <lun>: ] <sensor id> | <sensor
name>] ]
gethysteresis [shm <N> [ [ <lun>: ] <sensor id> | <sensor
name>] ]
gethysteresis <IPMB-address> -f <fru_id>
gethysteresis <IPMB-address> -f amc <amc_number>
gethysteresis board <N> -f <fru_id>
gethysteresis board <N> -f amc <amc_number>
gethysteresis shm <N> -f <fru_id>
gethysteresis shm <N> -f amc <amc_number>
```

3.29.2 Purpose

This command shows the current hysteresis values for the specified sensor(s). The sensor(s) must be threshold-based. Both raw and processed values are shown.

The option **-f** allows the user to select all sensors that belong to a specific FRU, designated either with its **<fru_id>** or, if it is an AMC, with the **amc <amc_number>** notation.

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.

This command can also be issued on the backup Shelf Manager; in that case, the current hysteresis values are only shown for sensors that are local to the backup Shelf Manager.

3.29.3 Examples

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
```

Show the hysteresis values for sensors that belong to FRU #5 on the IPM controller at IPMB address 20h

```
# clia gethysteresis  20 -f 5

Pigeon Point Shelf Manager Command Line Interpreter

20: LUN: 0, Sensor # 126 ("Temp_In Right")
    Type: Threshold (0x01), "Temperature" (0x01)
        Positive hysteresis, Raw Data: 0x00    Processed data: 0.000000
degrees C
        Negative hysteresis, Raw Data: 0x00    Processed data: 0.000000
degrees C

#
```

3.30 getipmbstate

3.30.1 Syntax

`getipmbstate <IPMB-address> [ <link> ]` (in radial IPMB-0 environment)
`getipmbstate <IPMB-address>` (in bused IPMB-0 environment)

3.30.2 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 IPM controller (sensor type F1). Information about both buses A and B is displayed.

The command works differently in bused and radial contexts. In a bused shelf, or in a radial shelf if the target IPM controller 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 shelf if the target IPM Controller is an IPMB hub, the command works as follows:

- If **`<link>`** is omitted, the command shows 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 **`<link>`** is present, the command shows 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.

This command can also be issued on the backup Shelf Manager; in that case, the current state of IPMB-0 is only reported for IPM controllers that are local to the backup Shelf Manager.

3.30.3 Examples

Show the current state of IPMB-0 on the IPM controller 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 IPMB link 8 on the Shelf Manager (the IPM controller at 20h).

```
# 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)
```

3.31 getlanconfig

3.31.1 Syntax

getlanconfig <channel> [<parameter-name> [<additional-parameters>]]
getlanconfig <channel> [<parameter-number> [<additional-parameters>]]

3.31.2 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.

The following table lists the names and numbers of LAN configuration parameters supported by the **getlanconfig** command:

Table 10 Names and Numbers of LAN Configuration Parameters Supported by the getlanconfig Command

PARAMETER NAME	NUMBER	DESCRIPTION
auth_support	1	An 8-bit value that contains authentication type support flags for the LAN channel.
auth_enables	2	Five 8-bit values that contain authentication 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 (e.g. 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 6 hexadecimal byte values delimited by ':' symbols (e.g. 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 (e.g. 255.255.255.0).
ipv4_hdr_param	7	Three 8-bit values that contain various IPv4 header parameters for sending RMCP packets: Time-to-live IP header flags (bits [7:5]) Precedence (bits [7:5]) and type of service (bits [4:1])

Parameter Name	Number	Description
`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 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 6 hexadecimal byte values delimited by ':' symbols.
`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 6 hexadecimal byte values delimited by ':' symbols.
`community`	16	A string value (up to 18 symbols) that is put into the "Community String" field in PET Traps.
`destination_count`	17	The maximum number of LAN alert destinations supported on the LAN channel.
`destination_type`	18	The destination type identified by the specified set selector. If no set selector is given, all destination types are shown. Each destination type entry contains the following fields: destination type (0-7) alert acknowledge flag alert acknowledge timeout / retry interval in seconds (1-256) number of retries (0-7)
`destination_address`	19	The destination addresses associated with the specified set selector. If no set selector is given, all destination addresses are shown. Each destination address entry contains the following fields: gateway selector: 0 – use default, 1 – use backup IP address (string in dotted decimal format) MAC address (string of 6 hexadecimal byte values delimited by ':' symbols)

PARAMETER NAME	NUMBER	DESCRIPTION
vlan_id	20	The flag that indicates whether Virtual LANs are enabled for the channel, and the Virtual LAN ID (a number in the range 1 to 4095, or the value 0, indicating that Virtual LAN IDs are not used for the channel)
vlan_priority	21	A number in the range 0 to 7, that specifies the packet Priority field, according to 802.1q
cs_entry_count	22	A number in the range 0 to 16, that specifies how many Cipher Suites are supported by the Shelf Manager
cs_entries	23	A list of the Cipher Suite IDs supported by the Shelf Manager, in hexadecimal, separated by commas. The standard Cipher Suite IDs are defined in the IPMI specification, version 2.0, section 22.15.2.
cs_priv_levels	24	A list of the currently assigned Cipher Suite privilege levels. Each item of the list includes a Cipher Suite ID and the maximum privilege level associated with that Cipher Suite.
destination_vlan_tag	25	The alert destination VLAN tag identified by the specified set selector. If no set selector is specified, all destination VLAN tags are shown. Each destination VLAN tag contains the following fields: destination selector (0-15) address format VLAN ID (0-4095) VLAN priority (0-7) For the Pigeon Point Shelf Manager, VLAN tags are read-only and are the same for all alert destinations.

The following subsections provide more detailed information about each of the supported parameters.

3.31.3 Examples

Get and show the whole 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
Destination Type:
    N/A
Destination Address:
    N/A
802.1q VLAN ID: 4 (enabled)
VLAN priority: 0
Cipher Suite Entry count: 15
Supported Cipher Suite IDs: 0h, 1h, 2h, 3h, 4h, 5h, 6h, 7h, 8h, 9h, Ah,
Bh, Ch, Dh, Eh
Cipher Suite Privilege Levels:
    ID 00h, Priv.Level 'User'            (2); ID 01h, Priv.Level 'User'
(2);
    ID 02h, Priv.Level 'Administrator'   (4); ID 03h, Priv.Level 'OEM
Proprietary' (5);
    ID 04h, Priv.Level 'OEM Proprietary' (5); ID 05h, Priv.Level 'OEM
Proprietary' (5);
    ID 06h, Priv.Level 'User'            (2); ID 07h, Priv.Level
'Administrator'   (4);
    ID 08h, Priv.Level 'OEM Proprietary' (5); ID 09h, Priv.Level 'OEM
Proprietary' (5);
    ID 0Ah, Priv.Level 'OEM Proprietary' (5); ID 0Bh, Priv.Level
'Administrator'   (4);
    ID 0Ch, Priv.Level 'OEM Proprietary' (5); ID 0Dh, Priv.Level 'OEM
Proprietary' (5);
    ID 0Eh, Priv.Level 'OEM Proprietary' (5);
Destination Address VLAN TAGs:
    DST VLAN TAGs # 0, Address Format: 802.1q VLAN TAG, VLAN Tag: ID 5,
CFI 0, Priority 0, (hex: 0x100500)
    DST VLAN TAGs # 1, Address Format: 802.1q VLAN TAG, VLAN Tag: ID 5,
CFI 0, Priority 0, (hex: 0x100500)
    DST VLAN TAGs # 2, Address Format: 802.1q VLAN TAG, VLAN Tag: ID 5,
CFI 0, Priority 0, (hex: 0x100500)
    DST VLAN TAGs # 3, Address Format: 802.1q VLAN TAG, VLAN Tag: ID 5,
CFI 0, Priority 0, (hex: 0x100500)
    DST VLAN TAGs # 4, Address Format: 802.1q VLAN TAG, VLAN Tag: ID 5,
CFI 0, Priority 0, (hex: 0x100500)
    DST VLAN TAGs # 5, Address Format: 802.1q VLAN TAG, VLAN Tag: ID 5,
CFI 0, Priority 0, (hex: 0x100500)
```

```
    DST VLAN TAGs # 6, Address Format: 802.1q VLAN TAG, VLAN Tag: ID 5,
CFI 0, Priority 0, (hex: 0x100500)
    DST VLAN TAGs # 7, Address Format: 802.1q VLAN TAG, VLAN Tag: ID 5,
CFI 0, Priority 0, (hex: 0x100500)
    DST VLAN TAGs # 8, Address Format: 802.1q VLAN TAG, VLAN Tag: ID 5,
CFI 0, Priority 0, (hex: 0x100500)
    DST VLAN TAGs # 9, Address Format: 802.1q VLAN TAG, VLAN Tag: ID 5,
CFI 0, Priority 0, (hex: 0x100500)
    DST VLAN TAGs # 10, Address Format: 802.1q VLAN TAG, VLAN Tag: ID
5, CFI 0, Priority 0, (hex: 0x100500)
    DST VLAN TAGs # 11, Address Format: 802.1q VLAN TAG, VLAN Tag: ID
5, CFI 0, Priority 0, (hex: 0x100500)
    DST VLAN TAGs # 12, Address Format: 802.1q VLAN TAG, VLAN Tag: ID
5, CFI 0, Priority 0, (hex: 0x100500)
    DST VLAN TAGs # 13, Address Format: 802.1q VLAN TAG, VLAN Tag: ID
5, CFI 0, Priority 0, (hex: 0x100500)
    DST VLAN TAGs # 14, Address Format: 802.1q VLAN TAG, VLAN Tag: ID
5, CFI 0, Priority 0, (hex: 0x100500)
    DST VLAN TAGs # 15, Address Format: 802.1q VLAN TAG, VLAN Tag: ID
5, CFI 0, Priority 0, (hex: 0x100500)#
```

3.31.4 auth_support

3.31.4.1 Syntax

getlanconfig <channel> auth_support
getlanconfig <channel> 1

3.31.4.2 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 following meaning of the bits:

- 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.

3.31.4.3 Examples

```
# clia getlanconfig 1 auth_support

Pigeon Point Shelf Manager Command Line Interpreter

Authentication Type Support: 0x15 ( None MD5 Straight Password/Key )

#
```

3.31.5 auth_enables

3.31.5.1 Syntax

getlanconfig <channel> auth_enables
getlanconfig <channel> 2

3.31.5.2 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 following meanings of the bits:

- 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.

3.31.5.3 Examples

Show the types of authentication supported by LAN channel 1.

```
# 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
#
```

3.31.6 ip

3.31.6.1 Syntax

getlanconfig <channel> ip
getlanconfig <channel> 3

3.31.6.2 Purpose

This command shows the current IP address used by the channel, in dotted decimal notation.

3.31.6.3 Examples

```
# clia getlanconfig 1 ip

Pigeon Point Shelf Manager Command Line Interpreter

IP Address: 172.16.2.203

#
```

3.31.7 ip_source

3.31.7.1 Syntax

getlanconfig <channel> ip_source
getlanconfig <channel> 4

3.31.7.2 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.

3.31.7.3 Examples

```
# clia getlanconfig 1 ip_source

Pigeon Point Shelf Manager Command Line Interpreter

IP Address Source: Static Address (Manually Configured) (0x01)

#
```

3.31.8 mac

3.31.8.1 Syntax

getlanconfig <channel> mac
getlanconfig <channel> 5

3.31.8.2 Purpose

This command shows the current MAC address used by the channel, in the form of six hexadecimal bytes separated by colons.

3.31.8.3 Examples

```
# clia getlanconfig 1 mac

Pigeon Point Shelf Manager Command Line Interpreter

MAC Address: 90:91:91:91:91:91

#
```

3.31.9 subnet_mask

3.31.9.1 Syntax

getlanconfig <channel> subnet_mask
getlanconfig <channel> 6

3.31.9.2 Purpose

This command shows the current IP subnet mask used by the channel, in dotted decimal notation.

3.31.9.3 Examples

```
# clia getlanconfig 1 subnet_mask

Pigeon Point Shelf Manager Command Line Interpreter

Subnet Mask: 255.255.255.0

#
```

3.31.10 ipv4_hdr_param

3.31.10.1 Syntax

getlanconfig <channel> ipv4_hdr_param
getlanconfig <channel> 7

3.31.10.2 Purpose

This command shows the current IP 4 header parameters. They are represented as 3 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.

3.31.10.3 Examples

```
# clia getlanconfig 1 ipv4_hdr_param

Pigeon Point Shelf Manager Command Line Interpreter

IPv4 Header Parameters: 0x40:0x40:0x10
```

```
#
```

3.31.11 *pri_rmcp_port*

3.31.11.1 Syntax

getlanconfig <channel> pri_rmcp_port
getlanconfig <channel> 8

3.31.11.2 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.

3.31.11.3 Examples

```
# clia getlanconfig 1 pri_rmcp_port

Pigeon Point Shelf Manager Command Line Interpreter

Primary RMCP Port Number: 0x026f

#
```

3.31.12 *sec_rmcp_port*

3.31.12.1 Syntax

getlanconfig <channel> sec_rmcp_port
getlanconfig <channel> 9

3.31.12.2 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.

3.31.12.3 Examples

```
# clia getlanconfig 1 sec_rmcp_port

Pigeon Point Shelf Manager Command Line Interpreter

Secondary RMCP Port Number: 0x0298

#
```

3.31.13 *arp_control*

3.31.13.1 Syntax

getlanconfig <channel> arp_control
getlanconfig <channel> 10

3.31.13.2 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 following meaning of the bits:

- 0x01 Enable Shelf Manager-generated Gratuitous ARPs
- 0x02 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.

3.31.13.3 Examples

```
# clia getlanconfig 1 arp_control

Pigeon Point Shelf Manager Command Line Interpreter

BMC-generated ARP Control: 02
    Enable BMC-generated Gratuitous Response

#
```

3.31.14 arp_interval

3.31.14.1 Syntax

`getlanconfig <channel> arp_interval`
`getlanconfig <channel> 11`

3.31.14.2 Purpose

This command shows the current ARP interval used by the channel. The value is shown as a number of seconds in fixed-point numeric format.

3.31.14.3 Examples

```
# clia getlanconfig 1 arp_interval

Pigeon Point Shelf Manager Command Line Interpreter

Gratuitous ARP Interval: 2.0 seconds

#
```

3.31.15 dft_gw_ip

3.31.15.1 Syntax

`getlanconfig <channel> dft_gw_ip`
`getlanconfig <channel> 12`

3.31.15.2 Purpose

This command shows the IP address of the default gateway used by the channel, in dotted decimal notation.

3.31.15.3 Examples

```
# clia getlanconfig 1 dft_gw_ip

Pigeon Point Shelf Manager Command Line Interpreter

Default Gateway Address: 0.0.0.0

#
```

3.31.16 *dft_gw_mac*

3.31.16.1 Syntax

getlanconfig <channel> dft_gw_mac
getlanconfig <channel> 13

3.31.16.2 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.

3.31.16.3 Examples

```
# clia getlanconfig 1 dft_gw_mac

Pigeon Point Shelf Manager Command Line Interpreter

Default Gateway MAC Address: N/A

#
```

3.31.17 *backup_gw_ip*

3.31.17.1 Syntax

getlanconfig <channel> backup_gw_ip
getlanconfig <channel> 14

3.31.17.2 Purpose

This command shows the IP address of the backup gateway used by the channel, in dotted decimal notation.

3.31.17.3 Examples

```
# clia getlanconfig 1 backup_gw_ip

Pigeon Point Shelf Manager Command Line Interpreter

Backup Gateway Address: 0.0.0.0
```

```
#
```

3.31.18 backup_gw_mac

3.31.18.1 Syntax

getlanconfig <channel> backup_gw_mac
getlanconfig <channel> 15

3.31.18.2 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.

3.31.18.3 Examples

```
# clia getlanconfig 1 backup_gw_mac

Pigeon Point Shelf Manager Command Line Interpreter

Backup Gateway MAC Address: N/A

#
```

3.31.19 community

3.31.19.1 Syntax

getlanconfig <channel> community
getlanconfig <channel> 16

3.31.19.2 Purpose

This command shows the community string parameter used in PET traps.

3.31.19.3 Examples

```
# clia getlanconfig 1 community

Pigeon Point Shelf Manager Command Line Interpreter

Community String: "public"

#
```

3.31.20 destination_count

3.31.20.1 Syntax

getlanconfig <channel> destination_count
getlanconfig <channel> 17

3.31.20.2 Purpose

This command shows the maximum number of alert destinations available for the channel. This is a configuration parameter for the Pigeon Point Shelf Manager and can be changed only through the shelfman configuration file.

3.31.20.3 Examples

```
# clia getlanconfig 1 destination_count

Pigeon Point Shelf Manager Command Line Interpreter

Number of Destinations: 16

#
```

3.31.21 destination_type

3.31.21.1 Syntax

getlanconfig <channel> destination_type [<set-selector>]
getlanconfig <channel> 18 [<set-selector>]

3.31.21.2 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:

- the destination selector
- the 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.

3.31.21.3 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
#
```

3.31.22 destination_address

3.31.22.1 Syntax

getlanconfig <channel> destination_address [<set-selector>]
getlanconfig <channel> 19 [<set-selector>]

3.31.22.2 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:

- the destination selector
- address format (IP+MAC by default)
- the destination IP address
- the 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.

3.31.22.3 Examples

```
# 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
#
```

3.31.23 vlan_id

3.31.23.1 Syntax

getlanconfig <channel> vlan_id
getlanconfig <channel> 20

3.31.23.2 Purpose

This command shows whether VLANs are currently enabled for the channel and the VLAN ID that is currently assigned to the designated channel. The Shelf Manager activates the Virtual LAN facility for a channel only if both VLANs are enabled and VLAN ID is not 0.

3.31.23.3 Examples

```
# clia getlanconfig 1 vlan_id

Pigeon Point Shelf Manager Command Line Interpreter

802.1q VLAN ID: 4 (enabled)

#
```

For a channel with disabled VLAN functionality:

```
# clia getlanconfig 1 vlan_id

Pigeon Point Shelf Manager Command Line Interpreter

802.1q VLAN ID: 0 (disabled)

#
```

3.31.24 vlan_priority

3.31.24.1 Syntax

getlanconfig <channel> vlan_priority
getlanconfig <channel> 21

3.31.24.2 Purpose

This command shows the VLAN priority that is currently assigned to the designated channel; this is the value of the Priority field used in the VLAN 802.1q network packet headers.

3.31.24.3 Examples

```
# clia getlanconfig 1 vlan_priority

Pigeon Point Shelf Manager Command Line Interpreter

VLAN priority: 0

#
```

3.31.25 cs_entry_count

3.31.25.1 Syntax

getlanconfig <channel> cs_entry_count
getlanconfig <channel> 22

3.31.25.2 Purpose

This command shows the number of Cipher Suites supported by the Shelf Manager (this number is read-only and has the value 15 for the Pigeon Point Shelf Manager).

3.31.25.3 Examples

```
# clia getlanconfig 1 cs_entry_count
```

```
Pigeon Point Shelf Manager Command Line Interpreter

Cipher Suite Entry count: 15

#
```

3.31.26 cs_entries

3.31.26.1 Syntax

getlanconfig <channel> cs_entries
getlanconfig <channel> 23

3.31.26.2 Purpose

This command shows the list of Cipher Suites IDs supported by the Shelf Manager. (The standard Cipher Suite IDs are listed in the IPMI specification version 2.0, section 22.15.2.)

3.31.26.3 Examples

```
# clia getlanconfig 1 cs_entries

Pigeon Point Shelf Manager Command Line Interpreter

Supported Cipher Suite IDs: 0h, 1h, 2h, 3h, 4h, 5h, 6h, 7h, 8h, 9h, Ah,
Bh, Ch, Dh, Eh

#
```

3.31.27 cs_priv_levels

3.31.27.1 Syntax

getlanconfig <channel> cs_priv_levels
getlanconfig <channel> 24

3.31.27.2 Purpose

This command shows the list of maximum privilege levels for the Cipher Suites supported by the Shelf Manager. Each item of the list contains a Cipher Suite ID and the maximum privilege level associated with that Cipher Suite. (The standard Cipher Suite IDs are listed in the IPMI specification version 2.0, section 22.15.2.) The maximum privilege level associated with a specific Cipher Suite can be changed via the CLI command **setlanconfig** or via the IPMI command “Set LAN Configuration Parameters”.

3.31.27.3 Examples

```
# clia getlanconfig 1 cs_priv_levels

Pigeon Point Shelf Manager Command Line Interpreter

Cipher Suite Privilege Levels:
    ID 00h, Priv.Level 'User'            (2); ID 01h, Priv.Level 'User'
(2);
```

```
    ID 02h, Priv.Level 'Administrator'   (4); ID 03h, Priv.Level 'OEM
Proprietary' (5);
    ID 04h, Priv.Level 'OEM Proprietary' (5); ID 05h, Priv.Level 'OEM
Proprietary' (5);
    ID 06h, Priv.Level 'User'            (2); ID 07h, Priv.Level
'Administrator'   (4);
    ID 08h, Priv.Level 'OEM Proprietary' (5); ID 09h, Priv.Level 'OEM
Proprietary' (5);
    ID 0Ah, Priv.Level 'OEM Proprietary' (5); ID 0Bh, Priv.Level
'Administrator'   (4);
    ID 0Ch, Priv.Level 'OEM Proprietary' (5); ID 0Dh, Priv.Level 'OEM
Proprietary' (5);
    ID 0Eh, Priv.Level 'OEM Proprietary' (5);
#
```

3.31.28 destination_vlan_tag

3.31.28.1 Syntax

getlanconfig <channel> destination_vlan_tag [<set-selector>]
getlanconfig <channel> 25 [<set-selector>]

3.31.28.2 Purpose

This command shows the element of the destination VLAN tag table with the index **<set-selector>**. Indices are 0-based. Selector 0 is used to address the volatile destination. The following information is shown about the designated destination:

- destination selector
- address format (“802.1q VLAN TAG” is the only one currently supported)
- VLAN ID
- Canonical format indicator (currently always 0)
- VLAN priority.

If the set selector is omitted, all destination VLAN tags are shown, with their associated destination numbers. However, if VLANs are not in use for the designated channel, the single line N/A is shown.

In the current implementation, VLAN tags are read-only and the same for all destinations on the same channel. In fact, the VLAN tag contents are derived from the VLAN-related channel configuration parameters.

3.31.28.3 Examples

```
# clia getlanconfig 1 destination_vlan_tag 2

Pigeon Point Shelf Manager Command Line Interpreter

Destination Addresse VLAN TAGs:
    DST VLAN TAGs # 2, Address Format: 802.1q VLAN TAG, VLAN Tag: ID 4,
CFI 0, Priority 0 (hex: 0x100004)
#
```

3.32 getpefconfig

3.32.1 Syntax

```
getpefconfig
getpefconfig <parameter-name> [ <additional-parameters> ]
getpefconfig <parameter-number> [ <additional-parameters> ]
```

3.32.2 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.

The following table lists names and numbers of PEF configuration parameters:

Table 11 Names and Numbers of PEF Configuration Parameters Supported by the getpefconfig Command

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

Parameter Name	Number	Description
`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.
`pet_format`	98	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 following subsections provide more detailed information about each of the supported parameters.

3.32.3 Examples

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 OEM Filters:
    0x01: OEM range boundary 0xff:0xff, alert policy # 1
    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
#
```

3.32.4 control

3.32.4.1 Syntax

`getpefconfig control`
`getpefconfig 1`

3.32.4.2 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 following meaning of the bits:

- 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.

3.32.4.3 Examples

```
# 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

#
```

3.32.5 action_control

3.32.5.1 Syntax

`getpefconfig action_control`
`getpefconfig 2`

3.32.5.2 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 following meaning of the bits:

- 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.

3.32.5.3 Examples

```
# 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
#
```

3.32.6 *startup_delay*

3.32.6.1 Syntax

`getpefconfig startup_delay`
`getpefconfig 3`

3.32.6.2 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.

3.32.6.3 Examples

```
# clia getpefconfig startup_delay

Pigeon Point Shelf Manager Command Line Interpreter

    PEF Startup Delay: 60 seconds

#
```

3.32.7 *alert_startup_delay*

3.32.7.1 Syntax

`getpefconfig alert_startup_delay`
`getpefconfig 4`

3.32.7.2 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.

3.32.7.3 Examples

```
# clia getpefconfig alert_startup_delay

Pigeon Point Shelf Manager Command Line Interpreter
```

```
    PEF Alert Startup Delay: 60 seconds

#
```

3.32.8 event_filter_count

3.32.8.1 Syntax

`getpefconfig event_filter_count`
`getpefconfig 5`

3.32.8.2 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 Pigeon Point Shelf Manager and can be changed only through the shelfman configuration file.

3.32.8.3 Examples

```
# clia getpefconfig event_filter_count

Pigeon Point Shelf Manager Command Line Interpreter

    PEF Number of Event Filters: 64

#
```

3.32.9 event_filter

3.32.9.1 Syntax

`getpefconfig event_filter [ <set-selector> ]`
`getpefconfig 6 [ <set-selector> ]`

3.32.9.2 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 and Compare 2 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.

3.32.9.3 Examples

```
# 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


#
```

3.32.10 event_filter_data1

3.32.10.1 Syntax

```
getpefconfig event_filter_data1 [ <set-selector> ]
getpefconfig 7 [ <set-selector> ]
```

3.32.10.2 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.

3.32.10.3 Examples

```
# 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")
```

```
#
```

3.32.11 alert_policy_count

3.32.11.1 Syntax

getpefconfig alert_policy_count
getpefconfig 8

3.32.11.2 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 Pigeon Point Shelf Manager and can be changed only through the shelfman configuration file.

3.32.11.3 Examples

```
# clia getpefconfig alert_policy_count

Pigeon Point Shelf Manager Command Line Interpreter

    Alert Policies Count: 64

#
```

3.32.12 alert_policy

3.32.12.1 Syntax

getpefconfig alert_policy [<set-selector>]
getpefconfig 9 [<set-selector>]

3.32.12.2 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.

3.32.12.3 Examples

```
# 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

#
```

3.32.13 system_guid

3.32.13.1 Syntax

getpefconfig system_guid
getpefconfig 10

3.32.13.2 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).

3.32.13.3 Examples

```
# clia getpefconfig system_guid

Pigeon Point Shelf Manager Command Line Interpreter

    PEF GUID: 23662f7f-ba1b-4b65-8808-94ca09c9bbb0

#
```

3.32.14 alert_string_count

3.32.14.1 Syntax

getpefconfig alert_string_count
getpefconfig 11

3.32.14.2 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 Pigeon Point Shelf Manager and can be changed only through the shelfman configuration file.

3.32.14.3 Examples

```
# clia getpefconfig alert_string_count

Pigeon Point Shelf Manager Command Line Interpreter

    Alert Strings Count: 64

#
```

3.32.15 alert_string_key

3.32.15.1 Syntax

getpefconfig alert_string_key [<set-selector>]
getpefconfig 12 [<set-selector>]

3.32.15.2 Purpose

This command shows the element of the alert string key table with index **<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.

3.32.15.3 Examples

```
# 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

#
```

3.32.16 alert_string

3.32.16.1 Syntax

getpefconfig alert_string [<set-selector>]
getpefconfig 13 [<set-selector>]

3.32.16.2 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.

3.32.16.3 Examples

```
# clia getpefconfig alert_string 2

Pigeon Point Shelf Manager Command Line Interpreter

    Alert Strings:
```

```
        0x02: "This is the alert string"

#
```

3.32.17 oem_filter_count

3.32.17.1 Syntax

getpefconfig oem_filter_count
getpefconfig 96

3.32.17.2 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 Pigeon Point 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, also to OEM timestamped and non-timestamped SEL entries (record type range C0h-FFh).

3.32.17.3 Examples

```
# clia getpefconfig oem_filter_count

Pigeon Point Shelf Manager Command Line Interpreter

    PEF Number of OEM Filters: 16

#
```

3.32.18 oem_filter

3.32.18.1 Syntax

getpefconfig oem_filter [<set-selector>]
getpefconfig 97 [<set-selector>]

3.32.18.2 Purpose

The OEM filter table is a Pigeon Point-defined OEM extension of the IPMI specification. It allows PEF to be applied, in addition to platform events, also 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.

3.32.18.3 Examples

```
# clia getpefconfig oem_filter

Pigeon Point Shelf Manager Command Line Interpreter

    Active OEM Filters:
    0x01: OEM range boundary 0xff:0xff, alert policy # 1
```

3.32.19 *pet_format*

3.32.19.1 Syntax

```
getpefconfig pet_format
getpefconfig 98
```

3.32.19.2 Purpose

Reports 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 following format types are defined:

0 = IPMI default format
1 = Plain Text format
2 = Multi-variable format.

3.32.19.3 Examples

```
# clia getpefconfig pet_format

Pigeon Point Shelf Manager Command Line Interpreter

    Platform Event Trap format: 0 (IPMI default)

# clia getpefconfig pet_format

Pigeon Point Shelf Manager Command Line Interpreter

    Platform Event Trap format: 1 (Plain text)

# clia getpefconfig pet_format

Pigeon Point Shelf Manager Command Line Interpreter

    Platform Event Trap format: 2 (Multi OID)
```

3.33 getsensoreventenable

3.33.1 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> ] ]
getsensoreventenable <IPMB-address> -f <fru_id>
getsensoreventenable <IPMB-address> -f amc <amc_number>
getsensoreventenable board <N> -f <fru_id>
getsensoreventenable board <N> -f amc <amc_number>
getsensoreventenable shm <N> -f <fru_id>
getsensoreventenable shm <N> -f amc <amc_number>
```

3.33.2 Purpose

This command shows the current event enable mask values of the specified sensor(s).

The option **-f** allows the user to select all sensors that belong to a specific FRU, designated either with its **<fru_id>** or, if it is an AMC, with the **amc <amc_number>** notation.

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 sensor(s). 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

This command can also be issued on the backup Shelf Manager; in that case, the current event enable mask values are only shown for sensors that are local to the backup Shelf Manager.

3.33.3 Examples

Get event enable values for a temperature sensor “Local Temp” on IPM controller FEh.

```
# 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

#
```

3.34 getthreshold/threshold

3.34.1 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> ] ]
getthreshold <IPMB-address> -f <fru_id>
getthreshold <IPMB-address> -f amc <amc_number>
getthreshold board <N> -f <fru_id>
getthreshold board <N> -f amc <amc_number>
getthreshold shm <N> -f <fru_id>
getthreshold shm <N> -f amc <amc_number>
```

The verb **threshold** can also be used instead of **getthreshold**.

3.34.2 Purpose

This command shows the current threshold values for the supported thresholds of the specified sensor(s). 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

The option **-f** allows the user to select all sensors that belong to a specific FRU, designated either with its **<fru_id>** or, if it is an AMC, with the **amc <amc_number>** notation.

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.

If a sensor has a decrementing linearization function according to its SDRs, its upper thresholds in raw format are essentially lower thresholds in processed format, and vice versa. This is reflected explicitly in the command output.

This command can also be issued on the backup Shelf Manager; in that case, the current threshold values are only shown for sensors that are local to the backup Shelf Manager.

3.34.3 Examples

Get threshold values for a temperature sensor “Local Temp” on IPM controller FEh.

```
# 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
```

Get threshold values for sensors that belong to FRU #5 on IPM controller 20h.

```
# clia getthreshold 20  -f 5

Pigeon Point Shelf Manager Command Line Interpreter

20: LUN: 0, Sensor # 126 ("Temp_In Right")
    Type: Threshold (0x01), "Temperature" (0x01)
        Upper Critical Threshold, Raw Data: 0x32    Processed data:
50.000000 degrees C
        Upper Non-Recoverable Threshold, Raw Data: 0x41    Processed
data: 65.000000 degrees C
```

Get threshold values for a sensor with a decrementing linearization function (“1/x” in this example).

```
# clia getthreshold 20 240

Pigeon Point Shelf Manager Command Line Interpreter

20: LUN: 0, Sensor # 240 ("Fan Tach. 0")
    Type: Threshold (0x01), "Fan" (0x04)
        Upper Critical Threshold, Raw Data: 0xfe ;
            Processed data (Lower Critical Threshold): 667.289470 RPM
```

3.35 help

3.35.1 Syntax

help [<command> [<sub command>]]

3.35.2 Purpose

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

This command can also be issued on the backup Shelf Manager.

3.35.3 Examples

```
# 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 notation)
        -v turns on verbose output

  activate <addr> <fru_id>
  alarm <alarm status/action>
  amcportstate [-v] <ipmc> [ amc <N> | <fru_id> ]
  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>
  firewall [<info [[[[<channel>]:<LUN>]:<NetFn>]:<command>] | <[all]>>
| <stop> | <start>]
  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> [-s|-d] [<file name>|-c]
```

```
  fruinfo <addr> <fru_id>
  getbootdev <addr> [<fru-id> | <amc-addr>]
  getconfigparam [<parameter name>]
  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> ] ]
  gethysteresis [ <addr> -f <fru id> ]
  gethysteresis [ <addr> -f amc <amc number> ]
  getipmbstate <addr> [<link>]
  getlanconfig <channel number> <parameter number> | <parameter name>
  getpefconfig <parameter name> | <parameter number> [<set selector>]
  getsensoreventenable [ <addr> [ [ lun: ]<sensor_id> | <sensor name>
] ]
  getsensoreventenable [ <addr> -f <fru id> ]
  getsensoreventenable [ <addr> -f amc <amc number> ]
  getthreshold [ <addr> [ [ lun: ]<sensor id> | <sensor name> ] ]
  getthreshold [ <addr> -f <fru id> ]
  getthreshold [ <addr> -f amc <amc number> ]
  help [<command>]
  ipmc [-v] [-x] [<addr>]
  localaddress
  minfanlevel <addr> <fru_id> <min fan level>
  minfanlevel [<min fan level>]
  poll
  quit
  sel [clear] [ <addr> [ <number of items> [<number of first item>] ] ]
  sel info [<addr>]
  sendamc <addr> <amc> [<lun>:]<netfn> <command> [<parameters ...>]
  sendcmd <addr> [<lun>:]<netfn> <command> [<parameters ...>]
  sensor [ <addr> [ [ lun: ]<sensor id> | <sensor name> ] ]
  sensor [ <addr> -f <fru id> ]
  sensor [ <addr> -f amc <amc number> ]
  sensordata [-t] [-d <state>] [ <addr> [ [ lun: ]<sensor id> | <sensor
name> ] ]
  sensordata [-t] [-d <state>] [ <addr> -f <fru id> ]
  sensordata [-t] [-d <state>] [ <addr> -f amc <amc number> ]
  sensorread <addr> [ lun: ]<sensor id>
  session
  setbootdev <addr> <fru-id | amc-addr> <boot-device>
  setcommandpolicy <Enable|Disable> <NetFn> <CMD> [<channel>] [<LUN>]
  setextracted <addr> <fru_id>
  setfanlevel <addr> <fru_id> <state>
  setfanpolicy <addr> <fru_id> <ENABLE|DISABLE [timeout]> [-s
<addr>|site_type <fru_id>|site_number]
  setfruledstate <addr> <fru_id> <LedId>|ALL <LedOp|tail> [LedColor]
  setfunctionpolicy <EnableDisableMask> <NetFn> <CMD> [<channel>]
[<LUN>]
  sethysteresis <addr> [ lun: ]<sensor_id> | <sensor name> pos | neg
<value>
  setipmbstate <addr> A|B [<link>] 0|1
  setlanconfig <channel number> <parameter number> | parameter name
<parameters ...>
  setlocked <addr> <fru_id> <value>
  setpefconfig <parameter name> | <parameter number> [<set selector>]
<parameters ...>
```

```
  setpowerlevel <addr> <fru_id> [<pwr_lvl>|OFF] [Copy]
  setsensordata <addr> [ lun: ]<sensor_id> | <sensor name> [ reading [-
r] <value> ]
        [ assertion <mask> ] [ deassertion <mask> ]
        [ event_data <b1> <b2> <b3> | event_data_no_offset <b1> <b2>
<b3>]
  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
  shelf <parameters>
  shelf board_lan_cfg_params
  shelf cooling_state
  shelf cs
  shelf address_table
  shelf at
  shelf fans_state
  shelf fs
  shelf h110_connectivity
  shelf h110c
  shelf ha_connectivity
  shelf hac
  shelf pci_connectivity
  shelf pcic
  shelf point-to-point_connectivity
  shelf ppc
  shelf power_distribution
  shelf pd
  shelf power_management
  shelf pm
  shelf shm_cfg_params
  shelfaddress [-x] ["<shelf address>"]
  shmstatus
  showunhealthy
  switchover [-force]
  terminate [-reboot]
  threshold [ <addr> [ [ lun: ]<sensor id> | <sensor name> ] ]
  threshold [ <addr> -f <fru id> ]
  threshold [ <addr> -f amc <amc number> ]
  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 enable <user id> 1|0
  user name <user id> <user name>
  user passwd <user id> <user password>
  version
#
# clia help shelf

Pigeon Point Shelf Manager Command Line Interpreter

  "shelf" command provides access to the dedicated records of the Shelf
FRU Info
        Activation <hw-addr> <fru_id> 1/0
        address_table
        Allowance <seconds>
```

```
        BDSelGrounded <slot number> 1/0
                1 means Enabled, 0 means Disabled
        board_lan_cfg_params
        cooling_state
        Deactivation <hw-addr> <fru_id> 1/0
        fans_state
        h110_connectivity
        ha_connectivity
        info_refresh
        info_force_update
        MaxCurrent [feed] <Amps>
        MinVoltage [feed] <Volts>
        pci_connectivity
        point-to-point_connectivity
        power_distribution
        power_management
        PwrCapability <hw-addr> <fru_id> <Watts>
        PwrDelay <hw-addr> <fru_id> <10ths_of_second>
        PwrReorder <hw-addr1> <fru_id1> before/after <hw-addr2>
<fru_id2>
        shm_cfg_params
        cold_sensitive <hw-addr> <fru_id> <enable>

  shelf <parameters>
```

```
# clia help shelf pwrreorder

Pigeon Point Shelf Manager Command Line Interpreter

  Change the order of FRU Activation and Power Descriptors
  instead of <addr> <fru_id> user may use:
        board <N>
        shm <N>
        power_supply <N>
        pem <N>
        fan_tray <N>
        board <M> amc <N>
        <addr> amc <N>

  PwrReorder <addr1> <fru_id1> before/after <addr2> <fru_id2>
```

3.36 hpi

3.36.1 Syntax

hpi <subcommand> [<additional-parameters>]

The following subcommands are supported:

- **session**
- **fru**
- **resource**

3.36.2 Purpose

The command **hpi** shows information related to the IntegralHPI implementation of the Hardware Platform Interface (HPI) in the Shelf Manager. The following subsections describe separate subcommands of the **hpi** command and the information that can be retrieved using these subcommands.

Note: this command works properly only if IntegralHPI is enabled, otherwise an error message is output.

3.36.3 Displaying HPI Session Information

3.36.3.1 Syntax

hpi session

3.36.3.2 Purpose

This variant of the **hpi** command shows HPI sessions opened to the IntegralHPI subsystem in the Shelf Manager. For each session, the command output includes the client's IP address and the number of HPI events queued to that client.

3.36.3.3 Examples

Get HPI session information for the current logical Shelf Manager.

```
# clia hpi session

Pigeon Point Shelf Manager Command Line Interpreter

HPI sessions:
 Session 0, Peer IP address: 192.168.1.56, Queued events: 0

#
```

3.36.4 Displaying HPI Resource Information

3.36.4.1 Syntax

hpi fru <addr> <fru id>

<addr> <fru_id> can be replaced with any the following alternatives:

power_supply <N>
pem <N>
fan_tray <N>
board <M> amc <N>
<addr> amc <N>
board <N>
shm <N>

hpi resource <resource id>

3.36.4.2 Purpose

These variants of the **hpi** command show information about a specific HPI resource. The resource can be specified using a FRU ID or a resource ID. In the first case, the command output describes the resource that corresponds to the specified FRU. In the second case, information about an arbitrary resource can be requested.

3.36.4.3 Examples

Get information about an HPI resource that represents FRU 3 on the logical Shelf Manager:

```
# clia hpi fru 20 3

Pigeon Point Shelf Manager Command Line Interpreter

20: FRU # 3
   HPI Resource Id: 400
   HPI Entity: {0x1e,1}{0x94,1}{0x10003,1}
   HPI Hot Swap State: ACTIVE
   Policy Execution Time: N/A
   HPI Resource Tag: "FanTray1"
```

Get information about an HPI resource that represents the first fan tray in the shelf:

```
#
# clia hpi fru fan_tray 1

Pigeon Point Shelf Manager Command Line Interpreter

20: FRU # 3
   HPI Resource Id: 400
   HPI Entity: {0x1e,1}{0x94,1}{0x10003,1}
   HPI Hot Swap State: ACTIVE
   Policy Execution Time: N/A
   HPI Resource Tag: "FanTray1"
```

Get information about an arbitrary HPI resource by its resource ID:

```
#
# clia hpi resource 400

Pigeon Point Shelf Manager Command Line Interpreter

20: FRU # 3
   HPI Resource Id: 400
   HPI Entity: {0x1e,1}{0x94,1}{0x10003,1}
   HPI Hot Swap State: ACTIVE
   Policy Execution Time: N/A
   HPI Resource Tag: "FanTray1"
```

3.37 ipmc

3.37.1 Syntax

`ipmc [-v] [-x] [<IPMB-address>]`
`ipmc [-v] [-x] board <N>`
`ipmc [-v] [-x] shm <N>`

3.37.2 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.

This command can operate in standard mode (both options **`-v`** and **`-x`** are omitted), in verbose mode (only option **`-v`** is present) or in verbose extended mode (only option **`-x`** is present or both options **`-v`** and **`-x`** are present).

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.X 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 (in both major-minor format and three-part format) 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 HPM.1 components of the target IPM controller, with their versions and descriptions
- The list of ports subject to E-Keying, with their states (Enabled/Disabled), including both PICMG and AXIe ports

The verbose extended mode differs from the extended mode in that the information about E-Keyed ports is shown in a different, more detailed format.

This command shows information about IPM controllers in state M1, if they were known previously to the Shelf Manager.

This command can also be issued on the backup Shelf Manager; in that case, the information is only reported for IPM controllers that are local to the backup Shelf Manager.

3.37.3 Examples

Get information about the IPM controller at address 9Ch.

```
# clia ipmc 9c

Pigeon Point Shelf Manager Command Line Interpreter

9c: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x08
    PICMG Version 2.2
    Hot Swap State: M4 (Active), Previous: M3 (Activation In Process),
Last State Change Cause: Normal State Change (0x0)

#
```

Get verbose information about the IPM controller at address 84h.

```
# clia ipmc -v 84

Pigeon Point Shelf Manager Command Line Interpreter

84: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x03
    PICMG Version 2.2
    Hot Swap State: M4 (Active), Previous: M3 (Activation In Process),
Last State Change Cause: Normal State Change (0x0)
    Device ID: 0x12, Revision: 0, Firmware: 1.51 (ver 1.5.1), IPMI ver
1.5
    Manufacturer ID: 00400a, Product ID: baba, Auxiliary Rev: 00000000
    Device ID String: "BMR-H8S BTP"
    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"
    HPM.1 Components:
        0: "H8S-AMCc F/W": Version 1.51, Aux 00000000
        1: "H8S-AMCc B/L": Version 1.51, Aux 00000000
    Links:
    84: Base Interface (0x00), Channel: 1
```

```
        Link: Disabled Ports: 1
    84: Base Interface (0x00), Channel: 2
        Link: Disabled Ports: 1
    84: Fabric Interface (0x01), Channel: 1
        Link: Disabled Ports: 1
    84: Fabric Interface (0x01), Channel: 2
        Link: Disabled Ports: 1
    84: Update Channel Interface (0x02), Channel: 1
        Link: Disabled Ports: 1

#
```

Get verbose extended information about the IPM controller at address 84h.

```
# clia ipmc -x 84

Pigeon Point Shelf Manager Command Line Interpreter

84: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x03
    PICMG Version 2.2
    Hot Swap State: M4 (Active), Previous: M3 (Activation In Process),
Last State Change Cause: Normal State Change (0x0)
    Device ID: 0x12, Revision: 0, Firmware: 1.51 (ver 1.5.1), IPMI ver
1.5
    Manufacturer ID: 00400a, Product ID: baba, Auxiliary Rev: 00000000
    Device ID String: "BMR-H8S BTP"
    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"
    HPM.1 Components:
        0: "H8S-AMCc F/W": Version 1.51, Aux 00000000
        1: "H8S-AMCc B/L": Version 1.51, Aux 00000000
    Links:
    84: Base Interface (0x00), Channel: 1
       Link Type: PICMG 3.0 Base Interface 10/100/1000 BASE-T
       Link Type Extension: 0 (10/100/1000 BASE-T)
       Link Grouping ID: 0x00
       Link Ports: 1
    84: Base Interface (0x00), Channel: 2
       Link Type: PICMG 3.0 Base Interface 10/100/1000 BASE-T
       Link Type Extension: 0 (10/100/1000 BASE-T)
       Link Grouping ID: 0x00
       Link Ports: 1
    84: Fabric Interface (0x01), Channel: 1
       Link Type: PICMG 3.1 Ethernet Fabric Interface
       Link Type Extension: 0 (1000BASE-BX)
       Link Grouping ID: 0x00
       Link Ports: 1
    84: Fabric Interface (0x01), Channel: 2
       Link Type: PICMG 3.1 Ethernet Fabric Interface
       Link Type Extension: 0 (1000BASE-BX)
       Link Grouping ID: 0x00
       Link Ports: 1
    84: Update Channel Interface (0x02), Channel: 1
```

```
       Link Type: OEM GUID-based (0xf0)
       Link Type Extension: 0
       Link Grouping ID: 0x00
       Link Ports: 1
```

Get verbose information about the IPM controller at address 10h (the physical Shelf Manager).

```
# clia ipmc -v 10

Pigeon Point Shelf Manager Command Line Interpreter

10: Entity: (0xf0, 0x60) Maximum FRU device ID: 0x08
    PICMG Version 2.3
    Hot Swap State: M4 (Active), Previous: M3 (Activation In Process),
Last State Change Cause: Normal State Change (0x0)
    Device ID: 0x00, Revision: 0, Firmware: 3.20 (ver. 3.2.0), IPMI ver
2.0
    Manufacturer ID: 00400a, Product ID: 0000, Auxiliary Rev: 00000000
    Device ID String: "ShMM-500"
    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"
    Links:
    10: Base Interface (0x00), Channel: 1
        Link: Disabled Ports: 1
    10: Base Interface (0x00), Channel: 2
        Link: Disabled Ports: 1
```

3.38 localaddress

3.38.1 Syntax

`localaddress`

3.38.2 Purpose

This command shows the IPMB address of the current Shelf Manager, based on its hardware address (as opposed to its generic BMC address 20h). These addresses will be different between redundant Shelf Managers (while the BMC address is shared between them).

This command can also be issued on the backup Shelf Manager.

3.38.3 Examples

```
# clia localaddress

Pigeon Point Shelf Manager Command Line Interpreter

Local IPMB Address = 0xFC

#
```

3.39 minfanlevel

3.39.1 Syntax

`minfanlevel [<level>]`
`minfanlevel <IPMB-address> <fru_id> [<level>]`[2]

3.39.2 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 won't 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 "Set Fan Level" 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.

In the shelves where zoned cooling is implemented, an alternative variant of this command is available that includes the parameters **`<IPMB-address>`** and **`<fru_id>`**. This syntax allows setting of the minimum fan level on a per-zone basis. The **`<IPMB-address>`** and **`<fru_id>`** in that case designate the Fan tray FRU for which the minimum fan level is set or queried. The command without parameters in such shelves shows the current minimum fan levels for all fan trays; the command with a single **`<level>`** parameter in such systems sets the same minimum fan level to all fan trays.

3.39.3 Examples

In a shelf that does not implement zoned cooling:

```
# clia minfanlevel 3

Pigeon Point Shelf Manager Command Line Interpreter

Minimal Fan Level is set to 3
```

---

[2] This variant of the syntax is valid only for shelves that implement zoned cooling

```
# clia minfanlevel

Pigeon Point Shelf Manager Command Line Interpreter

Minimal Fan Level is 3
Dynamic Minimum Fan Level is 3
```

In a shelf with zoned cooling:

```
# clia minfanlevel

Pigeon Point Shelf Manager Command Line Interpreter

Minimal Fan Level is 3
20: FRU # 3
    Dynamic Minimum Fan Level is 3
20: FRU # 4
    Dynamic Minimum Fan Level is 3
20: FRU # 5
    Dynamic Minimum Fan Level is 3
#
# clia minfanlevel 5

Pigeon Point Shelf Manager Command Line Interpreter

Minimal Fan Level is set to 5 for all fan trays
#
# clia minfanlevel

Pigeon Point Shelf Manager Command Line Interpreter

Minimal Fan Level is 5
20: FRU # 3
    Dynamic Minimum Fan Level is 5
20: FRU # 4
    Dynamic Minimum Fan Level is 5
20: FRU # 5
    Dynamic Minimum Fan Level is 5
#
# clia minfanlevel 20 4 7

Pigeon Point Shelf Manager Command Line Interpreter

Minimal Fan Level for (20, 4) is set to 7

# clia minfanlevel 20 4

Pigeon Point Shelf Manager Command Line Interpreter

Minimal Fan Level is 5
20: FRU # 4
    Dynamic Minimum Fan Level is 7
```

3.40 networkelementid

3.40.1 Syntax

`networkelementid [-i <index>] ["<id>"]`

3.40.2 Purpose

This command is carrier-specific and is not necessarily supported on all ShMM carriers.

This command shows or sets a Network Element Identifier if this parameter is supported by the current carrier. The superuser (UID 0) privilege is required for setting a Network Element Identifier.

Up to three Network Element Identifiers are supported for a single shelf. The index of the Network Element Identifier to retrieve or set is specified by the command line parameter **`<index>`**. This parameter can take values 1, 2 or 3; if omitted, the default index value is 1.

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 first Network Element Identifier is displayed.

3.40.3 Examples

```
#clia networkelementid

Pigeon Point Shelf Manager Command Line Interpreter

Network Element ID1: "0123456789A"

#clia networkelementid "01234567890"

Pigeon Point Shelf Manager Command Line Interpreter

Network Element ID1 is set successfully to "01234567890"
```

Retrieve the second Network Element Identifier for this shelf:

```
#clia networkelementid -i 2

Pigeon Point Shelf Manager Command Line Interpreter

Network Element ID2: "0123456789A"
```

3.41 poll

3.41.1 Syntax

`poll`

3.41.2 Purpose

This command initiates re-discovery 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.

3.41.3 Examples

```
# clia poll

Pigeon Point Shelf Manager Command Line Interpreter

IPMB polling thread started
```

3.42 sel

3.42.1 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.

3.42.2 Purpose

This command shows the contents of the 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, starting 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 show, 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
- Timestamp (for timestamped records) in the local time zone defined by the **TZ** environment variable
- Source address parameters: IPMB address, LUN and channel number
- Type and number of the sensor that generated the event
- Event/reading type code
- 3 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.

3.42.3 Examples

Reading 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
0x002A: 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
#

# clia sel b4 5

Pigeon Point Shelf Manager Command Line Interpreter

0x00A4: Event: at Nov 19 01:24:25 2003; from:(0x20,0,0);
sensor:(0x02,4); event:0x1(asserted): "Lower Non-Critical", Threshold:
0xb3, Reading: 0xb3
0x00B8: Event: at Nov 19 00:04:11 2003; from:(0x20,0,0);
sensor:(0x02,4); event:0x1(asserted): "Lower Non-Critical", Threshold:
0xb3, Reading: 0xb3
0x00CC: Event: at Nov 19 00:36:32 2003; from:(0x20,0,0);
sensor:(0x02,7); event:0x1(asserted): "Lower Non-Critical", Threshold:
0xae, Reading: 0x94
0x00E0: Event: at Nov 19 00:36:32 2003; from:(0x20,0,0);
sensor:(0x02,7); event:0x1(asserted): "Lower Critical", Threshold:
0xac, Reading: 0x94
0x00F4: Event: at Nov 19 00:02:37 2003; from:(0x20,0,0);
sensor:(0x01,2); event:0x1(asserted): "Upper Critical", Threshold:
0x13, Reading: 0x1c

# clia sel -v board 3 5

Pigeon Point Shelf Manager Command Line Interpreter

0x00A4: Event: at: Nov 19 01:24:25 2003; from IPM Controller: 0x20,
LUN: 0, Channel: 0
    "Voltage" (0x02) sensor # 4
    "Threshold" (0x01) event Asserted
    "Lower Non-Critical Going Low"
    Reading value: 0xb3
    Threshold value: 0xb3
```

```
0x00B8: Event: at: Nov 19 00:04:11 2003; from IPM Controller: 0x20,
LUN: 0, Channel: 0
    "Voltage" (0x02) sensor # 4
    "Threshold" (0x01) event Asserted
    "Lower Non-Critical Going Low"
    Reading value: 0xb3
    Threshold value: 0xb3

0x00CC: Event: at: Nov 19 00:36:32 2003; from IPM Controller: 0x20,
LUN: 0, Channel: 0
    "Voltage" (0x02) sensor # 7
    "Threshold" (0x01) event Asserted
    "Lower Non-Critical Going Low"
    Reading value: 0x94
    Threshold value: 0xae

0x00E0: Event: at: Nov 19 00:36:32 2003; from IPM Controller: 0x20,
LUN: 0, Channel: 0
    "Voltage" (0x02) sensor # 7
    "Threshold" (0x01) event Asserted
    "Lower Critical Going Low"
    Reading value: 0x94
    Threshold value: 0xac

0x00F4: Event: at: Nov 19 00:02:37 2003; from IPM Controller: 0x20,
LUN: 0, Channel: 0
    "Temperature" (0x01) sensor # 2
    "Threshold" (0x01) event Asserted
    "Upper Critical Going High"
    Reading value: 0x1c
    Threshold value: 0x13
#
```

Getting 5 sel entries, starting with 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
#
```

Clearing 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

#
```

3.43 sendamc

3.43.1 Syntax

sendamc <addr> <AMC-address> [<lun>:]<netfn> <command-code> [<parameter1> …<parameterN>]

3.43.2 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 target LUN of the command on the IPMB-L is specified by the **<lun>** parameter. (The default is LUN 0; if specified, **<lun>** is separated from **<netfn>** with a colon and no spaces.) The code of the command is specified by the **<command-code>** parameter. The request data bytes of the command are represented by **<parameter1>**, **<parameter2>**, etc.

The command reports the completion code of the IPMI command and the response data are displayed as hexadecimal bytes.

3.43.3 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 doesn’t 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. LUN 0 is explicitly specified. Since this command doesn’t require request data, no **<parameter1>**, **<parameter2>**, … are specified.

```
# clia sendamc 84 72 0:6 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
#
```

3.44 sendcmd

3.44.1 Syntax

sendcmd <IPMB-address> [<lun>:]<netfn> <command-code> [<parameter1> …<parameterN>]

3.44.2 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 0 – FF. 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 target LUN of the command is specified by the **<lun>** parameter. (The default is LUN 0; if specified, **<lun>** is separated from **<netfn>** with a colon and no spaces.) The code of the command is specified by the **<command-code>** parameter. The request data bytes of the command are represented by **<parameter1>**, **<parameter2>**, etc.

The command reports the completion code resulting from the IPMI command and the response data, all are displayed as hexadecimal bytes.

3.44.3 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 doesn’t require request data, no **<parameter1>**, **<parameter2>**, … are specified.

```
# clia sendcmd 20 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
#
```

Send the same command as above, but with LUN 0 explicitly specified.

```
# clia sendcmd 20 0:6 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
#
```

3.45 sensor

3.45.1 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> ] ]
sensor <IPMB-address> -f <fru_id>
sensor <IPMB-address> -f amc <amc_number>
sensor board <N> -f <fru_id>
sensor board <N> -f amc <amc_number>
sensor shm <N> -f <fru_id>
sensor shm <N> -f amc <amc_number>
```

3.45.2 Purpose

This command shows information about specific sensor(s). 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 also shows information about event-only sensors (a new category of sensors defined in the version 2.0 of the IPMI specification; these sensors have special reduced SDRs and support event generation only).

The option **–f** allows the user to select all sensors that belong to a specific FRU, designated either with its **<fru_id>** or, if it is an AMC, with the **amc <amc_number>** notation.

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)

In verbose mode, additional information about the sensor may be shown depending on the sensor category (threshold-based, discrete or event-only).

For threshold-based sensors, the following information is shown:

- Assertion mask
- Deassertion mask
- Settable/readable mask for sensor thresholds
- 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.

For discrete sensors, the following information is shown:

- Assertion mask
- Deassertion mask
- Settable/readable mask for sensor states

For event-only sensors, no additional information is shown.

This command can also be issued on the backup Shelf Manager; in that case, the information is only shown for sensors that are local to the backup Shelf Manager.

3.45.3 Examples

Get standard information about sensor “FAN 4” on IPM controller FEh.

```
# clia sensor fe "FAN 4"

Pigeon Point Shelf Manager Command Line Interpreter

fe: LUN: 0, Sensor # 14 ("FAN 4")
    Type: Threshold (0x01), "Fan" (0x04)
    Belongs to entity: (0xd0, 0) [FRU # 0]

#
```

Get verbose information about sensor 2 on IPM controller 9Ch.

```
# clia sensor -v 9c 2

Pigeon Point Shelf Manager Command Line Interpreter
```

```
9c: LUN: 0, Sensor # 2 ("emulated temp")
    Type: Threshold (0x01), "Temperature" (0x01)
    Belongs to entity: (0xd0, 0) [FRU # 0]
    Assertion Mask: 0x7a95
        Lower Non-Critical Going Low
        Lower Critical Going Low
        Lower Non-Recoverable Going Low
        Upper Non-Critical Going High
        Upper Critical Going High
        Upper Non-Recoverable Going High
        Upper non-critical threshold is comparison returned
        Upper critical threshold is comparison returned
        Upper non-recoverable threshold comparison is returned
    Deassertion Mask: 0x7a95
        Lower Non-Critical Going Low
        Lower Critical Going Low
        Lower Non-Recoverable Going Low
        Upper Non-Critical Going High
        Upper Critical Going High
        Upper Non-Recoverable Going High
        Upper non-critical threshold is comparison returned
        Upper critical threshold is comparison returned
        Upper non-recoverable threshold comparison is returned
    Settable / Readable Mask: 0x3f3f
        Lower Non-Critical Threshold is Readable
        Lower Critical Threshold is Readable
        Lower Non-Recoverable Threshold is Readable
        Upper Non-Critical Threshold is Readable
        Upper Critical Threshold is Readable
        Upper Non-Recoverable Threshold is Readable
        Lower Non-Critical Threshold is Settable
        Lower Critical Threshold is Settable
        Lower Non-Recoverable Threshold is Settable
        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 = 1, B = 0, K1 = 0, K2 = 0
    Tolerance = 0, Accuracy = 0, Accuracy EXP = 0
    Analog Flags: 0x0
    Nominal: 0 (0x00), Normal max: 0 (0x00), Normal min: 0 (0x00)
    Sensor max: 127 (0x7f), Sensor min: 128 (0x80)
    Upper Thresholds:
        Non-Critical: 70 (0x46) Critical: 80 (0x50) Non-Recoverable: 90
(0x5a)
    Lower Thresholds:
        Non-Critical: 3 (0x03) Critical: 0 (0x00) Non-Recoverable: 251
(0xfb)
    Hysteresis:
        Positive: 2 (0x02), Negative 2 (0x02)

#
```

Same as above, but explicitly specifying the LUN for the sensor.

```
# clia sensor -v 9c 0:2

Pigeon Point Shelf Manager Command Line Interpreter

9c: LUN: 0, Sensor # 2 ("emulated temp")
    Type: Threshold (0x01), "Temperature" (0x01)
    Belongs to entity: (0xd0, 0) [FRU # 0]
    Assertion Mask: 0x7a95
        Lower Non-Critical Going Low
        Lower Critical Going Low
        Lower Non-Recoverable Going Low
        Upper Non-Critical Going High
        Upper Critical Going High
        Upper Non-Recoverable Going High
        Upper non-critical threshold is comparison returned
        Upper critical threshold is comparison returned
        Upper non-recoverable threshold comparison is returned
    Deassertion Mask: 0x7a95
        Lower Non-Critical Going Low
        Lower Critical Going Low
        Lower Non-Recoverable Going Low
        Upper Non-Critical Going High
        Upper Critical Going High
        Upper Non-Recoverable Going High
        Upper non-critical threshold is comparison returned
        Upper critical threshold is comparison returned
        Upper non-recoverable threshold comparison is returned
    Settable / Readable Mask: 0x3f3f
        Lower Non-Critical Threshold is Readable
        Lower Critical Threshold is Readable
        Lower Non-Recoverable Threshold is Readable
        Upper Non-Critical Threshold is Readable
        Upper Critical Threshold is Readable
        Upper Non-Recoverable Threshold is Readable
        Lower Non-Critical Threshold is Settable
        Lower Critical Threshold is Settable
        Lower Non-Recoverable Threshold is Settable
        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 = 1, B = 0, K1 = 0, K2 = 0
    Tolerance = 0, Accuracy = 0, Accuracy EXP = 0
    Analog Flags: 0x0
    Nominal: 0 (0x00), Normal max: 0 (0x00), Normal min: 0 (0x00)
    Sensor max: 127 (0x7f), Sensor min: 128 (0x80)
    Upper Thresholds:
        Non-Critical: 70 (0x46) Critical: 80 (0x50) Non-Recoverable: 90
(0x5a)
    Lower Thresholds:
```

```
        Non-Critical: 3 (0x03) Critical: 0 (0x00) Non-Recoverable: 251
(0xfb)
    Hysteresis:
        Positive: 2 (0x02), Negative 2 (0x02)

#
```

Get standard information about sensors that belong to FRU #1 on IPM controller 20h.

```
# # clia sensor 20  -f 1

Pigeon Point Shelf Manager Command Line Interpreter

20: LUN: 0, Sensor # 2 ("FRU 1 HOT_SWAP")
    Type: Discrete (0x6f), "Hot Swap" (0xf0)
    Belongs to entity (0xf2, 96): [FRU # 1]

20: LUN: 0, Sensor # 194 ("Shelf EEPROM 1")
    Type: Discrete (0x6f), "Entity Presence" (0x25)
    Belongs to entity (0xf2, 96): [FRU # 1]
```

Get verbose information about an event-only sensor.

```
# clia sensor -v 0x82 128

Pigeon Point Shelf Manager Command Line Interpreter

82: LUN: 0, Sensor # 128 ("Memory")
    Type: Discrete (0x6f), "Memory" (0x0c)
    Belongs to entity (0xa0, 96): [FRU # 0]
    This is an Event-Only Sensor
```

3.46 sensordata

3.46.1 Syntax

```
sensordata [-v] [-t] [-d <state>] [ <IPMB-address>
[<sensor-name> | [<lun>:]<sensor-number> ] ]
sensordata [-v] [-t] [-d <state>] board <N> [<sensor-name>
| [<lun>:]<sensor-number> ] ]
sensordata [-v] [-t] [-d <state>] shm <N> [<sensor-name> |
[<lun>:]<sensor-number> ] ]
sensordata [-v] [-t] [-d <state>] <IPMB-address> -f
<fru_id>
sensordata [-v] [-t] [-d <state>] <IPMB-address> -f amc
<amc_number>
sensordata [-v] [-t] [-d <state>] board <N> -f <fru_id>
sensordata [-v] [-t] [-d <state>] board <N> -f amc
<amc_number>
sensordata [-v] [-t] [-d <state>] shm <N> -f <fru_id>
sensordata [-v] [-t] [-d <state>] shm <N> -f amc
<amc_number>
```

3.46.2 Purpose

This command shows the actual value of the specified sensor(s) (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.

The option **-f** allows the user to select all sensors that belong to a specific FRU, designated either with its **<fru_id>** or, if it is an AMC, with the **amc <amc_number>** notation.

The option **-d** allows the user to select discrete sensors that have the state **<state>** (which is a decimal number in the range of 0 to 14) set in their state masks.

If the option **-t** is specified, information is displayed only for threshold-based sensors that have at least one of their thresholds crossed.

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 (e.g., 27 degrees). The discrete value is annotated according to the event/reading code type (e.g. for the event/reading code 2, the asserted state 0 is shown as “Transition to Idle”) and also according to the sensor type if the event/reading code is “Sensor-specific” and the sensor type is known to the Shelf Manager. The processed value is not shown if the sensor is in the state “Initial Update In Progress/Sensor Reading Unavailable” (since the processed value does not make sense in that state).

This command can also be issued on the backup Shelf Manager; in that case, the information is only shown for sensors that are local to the backup Shelf Manager.

3.46.3 Examples

Get sensor data values for a temperature sensor “Local Temp” on IPM controller FEh.

```
# 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
    Current State Mask: 0x00

#
```

Get sensor data values for a discrete (Hot Swap) sensor (#0) on IPM controller 9Ch.

```
# 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

#
```

Get sensor data values for sensors that belong to FRU #1 on IPM controller 20h.

```
# clia sensordata 20 -f 1

Pigeon Point Shelf Manager Command Line Interpreter

20: LUN: 0, Sensor # 2 ("FRU 1 HOT_SWAP")
    Type: Discrete (0x6f), "Hot Swap" (0xf0)
    Belongs to entity (0xf2, 0x60): FRU # 1
    Status: 0xc0
        All event messages enabled from this sensor
        Sensor scanning enabled
        Initial update completed
    Sensor reading: 0x00
    Current State Mask 0x0010

20: LUN: 0, Sensor # 194 ("Shelf EEPROM 1")
    Type: Discrete (0x6f), "Entity Presence" (0x25)
    Belongs to entity (0xf2, 0x60): FRU # 1
    Status: 0xc0
        All event messages enabled from this sensor
        Sensor scanning enabled
        Initial update completed
    Sensor reading: 0x00
    Current State Mask 0x0001
        Entity Present
```

Get sensor data values for discrete sensors on IPM controller 20h that have state #1 set in their state masks.

```
# clia sensordata -d 1 20

Pigeon Point Shelf Manager Command Line Interpreter

20: LUN: 0, Sensor # 138 ("SHM Redundancy")
    Type: Discrete (0x0b), "Management Subsystem Health" (0x28)
    Belongs to entity (0xf0, 0x01): FRU # 0
    Status: 0xc0
        All event messages enabled from this sensor
        Sensor scanning enabled
        Initial update completed
    Sensor reading: 0x00
    Current State Mask 0x0002
        Redundancy Lost
```

Get the current state of the Reboot Reason sensor on the first physical Shelf Manager.

```
#
# clia sensordata 10 129

Pigeon Point Shelf Manager Command Line Interpreter

10: LUN: 0, Sensor # 129 ("Reboot Reason")
    Type: Discrete (0x6f), "OEM reserved" (0xdd)
    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 0x0400
        Reboot caused by ShMM power cycle

#
```

3.47 sensorread

3.47.1 Syntax

`sensorread <IPMB-address> [<lun>:]<sensor-number>`

3.47.2 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 cannot 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.

This command can also be issued on the backup Shelf Manager; in that case, the raw values are only shown for sensors that are local to the backup Shelf Manager.

3.47.3 Examples

Get sensor data values for sensor 4 on IPM controller FCh. 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
```

```
    Current State Mask: 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
```

3.48 session

3.48.1 Syntax

session [-v]

3.48.2 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.

If the option **-v** is specified, detailed information about the active RMCP/RMCP+ sessions is added. The additional information includes session activity timestamp, authentication type and challenge string. If there are no active sessions, this information is provided for the last active session.

3.48.3 Examples

```
# 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
#
# clia session  -v

Pigeon Point Shelf Manager Command Line Interpreter

32 sessions possible, 1 sessions currently active

Session: 1,
   User: ID 1, Name: ""; Privilege Level: "Administrator"; Notifier:
Off
   Channel: 1 ("LAN_802_3"); Peer IP address: 80.240.102.45, Port: 2556

   Session Activity Timestamp: Fri Feb 12 13:28:59
```

```
   Maximum Privilege Level: Administrator; Authentication type: None;
   Challenge: F7 42 2A 9B D3 77 20 90 B6 44 6E 3E 0A AC DB 89
```

```
# clia session -v

Pigeon Point Shelf Manager Command Line Interpreter

32 sessions possible, 0 sessions currently active

Last active session:
   Session: 1, RMCP
   User: ID 1, Name: ""; Privilege Level: "Administrator"; Notifier:
Off
   Channel: 1 ("LAN_802_3"); Peer IP address: 80.240.102.45, Port: 2556

   Session Activity Timestamp: Fri Feb 12 13:28:59 2010
   Maximum Privilege Level: Administrator; Authentication type: None;
   Challenge: 36 5A B7 B0 F5 64 54 92 BB AF 12 3E 43 7A 1E E0
```

3.49 setbootdev

3.49.1 Syntax

setbootdev <IPMB-0-address> <fru_id>| <IPMB-L-address> <boot-device>

3.49.2 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 may be:

- **1** or **pxe** (Pre-Boot Execution Environment)
- **2** or **disk** (Default Hard Drive)
- **3** or s**afe** (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).

3.49.3 Examples

Set the system boot parameter for an IPM controller at IPMB-0 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
```

3.50 setcommandpolicy

3.50.1 Syntax

setcommandpolicy enable <netfn> <command_code> [<channel> [<LUN>]]
setcommandpolicy disable <netfn> <command_code> [<channel> [<LUN>]]

The arguments **<channel>**, **<LUN>**, **<netfn>** and **<command_code>** can be used in hexadecimal or decimal forms. Any of these arguments is treated as hexadecimal if it includes a 0x prefix or a hexadecimal letter (A, B, C, D, E, and F); otherwise, the argument is treated as a decimal. All the arguments are case insensitive.

3.50.2 Purpose

This command allows enabling or disabling a specific command for execution by the Shelf Manager, using the IPMI 2.0 firmware firewall functionality. To discover what commands can be enabled or disabled, use the command **clia firewall info**. If no channel/LUN is specified, the policy change applies to the specified command for all channels and/or LUNs. Command and network function codes are defined in the IPMI specification.

3.50.3 Examples

Enable the command “Set LAN Configuration Parameters” (NetFn = 0xC, command code = 1) for all channels and LUNs.

```
# clia setcommandpolicy enable 0xC 0x1

Pigeon Point Shelf Manager Command Line Interpreter

    Command 0x01(1) has been enabled (enabled previously) for channel
0x00, LUN 0, NetFn 0x0C...
    Command 0x01(1) has been enabled (enabled previously) for channel
0x00, LUN 1, NetFn 0x0C...
    Command 0x01(1) has been enabled (enabled previously) for channel
0x00, LUN 2, NetFn 0x0C...
    Command 0x01(1) has been enabled (disabled previously) for channel
0x00, LUN 3, NetFn 0x0C...
    Command 0x01(1) has been enabled (enabled previously) for channel
0x01, LUN 0, NetFn 0x0C...
    Command 0x01(1) has been enabled (enabled previously) for channel
0x01, LUN 1, NetFn 0x0C...
    Command 0x01(1) has been enabled (enabled previously) for channel
0x01, LUN 2, NetFn 0x0C...
    Command 0x01(1) has been enabled (enabled previously) for channel
0x01, LUN 3, NetFn 0x0C...
    Command 0x01(1) has been enabled (enabled previously) for channel
0x02, LUN 0, NetFn 0x0C...
    Command 0x01(1) has been enabled (enabled previously) for channel
0x02, LUN 1, NetFn 0x0C...
```

```
    Command 0x01(1) has been enabled (enabled previously) for channel
0x02, LUN 2, NetFn 0x0C...
    Command 0x01(1) has been enabled (enabled previously) for channel
0x02, LUN 3, NetFn 0x0C...
    Command 0x01(1) has been enabled (enabled previously) for channel
0x0F, LUN 0, NetFn 0x0C...
    Command 0x01(1) has been enabled (enabled previously) for channel
0x0F, LUN 1, NetFn 0x0C...
    Command 0x01(1) has been enabled (enabled previously) for channel
0x0F, LUN 2, NetFn 0x0C...
    Command 0x01(1) has been enabled (enabled previously) for channel
0x0F, LUN 3, NetFn 0x0C...
#
```

Disable the command “Set LAN Configuration Parameters” (NetFn = 0xC, command code = 1) for channel 0 and LUN 1

```
# clia setcommandpolicy disable 0xC 0x1 0 1

Pigeon Point Shelf Manager Command Line Interpreter

    Command 0x01(1) has been disabled (enabled previously) for channel
0x00, LUN 1, NetFn 0x0C...
#
```

3.51 setextracted

3.51.1 Syntax

```
setextracted <IPMB-address> <fru_id>
setextracted shm <N>
setextracted board <N>
setextracted power_supply <N>
setextracted pem <N>
setextracted fan_tray <N>
setextracted <IPMB-address> amc <M>
setextracted board <N> amc <M>
```

3.51.2 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).

3.51.3 Examples

```
# clia setextracted 9c 0

Pigeon Point Shelf Manager Command Line Interpreter

    Set FRU extracted state successfully

# clia setextracted 96 amc 1

Pigeon Point Shelf Manager Command Line Interpreter

    Set FRU extracted state successfully

#
```

3.52 setfanlevel

3.52.1 Syntax

```
setfanlevel <IPMB-address> <fru_id> <level>
setfanlevel shm <N> <level>
setfanlevel board <N> <level>
setfanlevel power_supply <N> <level>
setfanlevel pem <N> <level>
setfanlevel fan_tray <N> <level>
setfanlevel all <level>
```

3.52.2 Purpose

This command sets a new level for the fan controlled by the FRU specified in the command parameters.

The version of this command with an **`all`** qualifier attempts to set the same level for all known fans in the shelf.

3.52.3 Examples

Set fan level for the fan controlled by FRU #2 at IPMB address 20h 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
#
```

3.53 setfanpolicy

3.53.1 Syntax

setfanpolicy <fan tray addr> <fan tray fru_id> ENABLE|DISABLE [timeout] [-s <addr>|<site_type> <fru_id>|<site_number>]

3.53.2 Purpose

This command enables or disables Shelf Manager control over fan trays for cooling management purposes. This control is enabled by default; it can be disabled temporarily or for an indefinite period of time. In addition, for shelves with zoned cooling management, control can be enabled or disabled with respect to a specific FRU; in that case, the Shelf Manager does not react to thermal events from that FRU by changing the fan level of the specified fan.

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, but rounded to 5 second units in accordance with the PICMG 3.0 specification. The value of the **<timeout>** parameter may not be greater then 21 minutes (**1260** seconds) and the minimum value of **<timeout>** is **5** seconds. If the **<timeout>** variable is not specified, the DISABLE period 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.

3.53.3 Examples

Disable Shelf Manager control over the fan for 60 seconds with respect to a specific FRU. The fan tray is at IPMB address 20h, FRU ID 3. The designated FRU site (which is assumed to be cooled by that 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 Shelf Manager control over the fan for infinite time with respect to a specific site. 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
```

3.54 setfruledstate

3.54.1 Syntax

```
setfruledstate <IPMB-address> <fru_id> <LedId>| ALL
<LedOp> [<LedColor>]
<LedOp> ::= ON | OFF | LOCAL | BLINK <onTime> <offTime> |
TEST <onTime>
<LedColor> ::= BLUE | RED | GREEN | AMBER | ORANGE | WHITE
| NONE | <number>
```

<IPMB-address> <fru_id> can be replaced with any of the following alternatives:

```
board <N>
shm <N>
power_supply <N>
pem <N>
fan_tray <N>
<IPMB-address> amc <M>
board <N> amc <M>
```

3.54.2 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 FRU(s), 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 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** = 5
- **WHITE** = 6
- **NONE** = 14 (doesn't change color).

This command can also be issued on the backup Shelf Manager; in that case, the FRU LED state can only be set for FRU LEDs that are local to the backup Shelf Manager.

3.54.3 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
```

3.55 setfunctionpolicy

3.55.1 Syntax

`setfunctionpolicy <EnableDisableMask> <netfn> <command_code> [<channel> [<LUN>]]`

The arguments **`<channel>`**, **`<LUN>`**, **`<netfn>`** and **`<command_code>`** can be used in hexadecimal or decimal forms. Any of these arguments is treated as hexadecimal if it includes a 0x prefix or a hexadecimal letter (A, B, C, D, E, and F); otherwise, the argument is treated as a decimal. All the arguments are case insensitive.

3.55.2 Purpose

This command allows enabling or disabling specific subfunctions of a specific command for execution, using the IPMI 2.0 firmware firewall functionality. To discover what subfunctions can be enabled or disabled, use the command **`clia firewall info`**. An attempt to disable a non-existent function is silently ignored; an attempt to enable a non-existent function results in an error message.

3.55.3 Examples

Enable functions 0, 6 and 9 and disable all other (existing) functions of the command “Set PEF Configuration Parameters” (NetFn = 0x4, command code = 0x12) for all channels and LUNs. For this specific command, this means allowing the change of only the following PEF configuration parameters: PEF Control, Event Filter Table and Alert Policy Table.

```
# clia setfunctionpolicy 0x241 0x4 0x12

Pigeon Point Shelf Manager Command Line Interpreter

    Command functions 0x0000000000000241 have been set
(0x0000000000003FFF previously) for channel 0x00, LUN 0, command 0x12,
NetFn 0x04...
    Command functions 0x0000000000000241 have been set
(0x0000000000003FFF previously) for channel 0x00, LUN 1, command 0x12,
NetFn 0x04...
    Command functions 0x0000000000000241 have been set
(0x0000000000003FFF previously) for channel 0x00, LUN 2, command 0x12,
NetFn 0x04...
    Command functions 0x0000000000000241 have been set
(0x0000000000003FFF previously) for channel 0x00, LUN 3, command 0x12,
NetFn 0x04...
    Command functions 0x0000000000000241 have been set
(0x0000000000003FFF previously) for channel 0x01, LUN 0, command 0x12,
NetFn 0x04...
    Command functions 0x0000000000000241 have been set
(0x0000000000003FFF previously) for channel 0x01, LUN 1, command 0x12,
NetFn 0x04...
    Command functions 0x0000000000000241 have been set
(0x0000000000003FFF previously) for channel 0x01, LUN 2, command 0x12,
NetFn 0x04...
```

```
    Command functions 0x0000000000000241 have been set
(0x0000000000003FFF previously) for channel 0x01, LUN 3, command 0x12,
NetFn 0x04...
    Command functions 0x0000000000000241 have been set
(0x0000000000003FFF previously) for channel 0x02, LUN 0, command 0x12,
NetFn 0x04...
    Command functions 0x0000000000000241 have been set
(0x0000000000003FFF previously) for channel 0x02, LUN 1, command 0x12,
NetFn 0x04...
    Command functions 0x0000000000000241 have been set
(0x0000000000003FFF previously) for channel 0x02, LUN 2, command 0x12,
NetFn 0x04...
    Command functions 0x0000000000000241 have been set
(0x0000000000003FFF previously) for channel 0x02, LUN 3, command 0x12,
NetFn 0x04...
    Command functions 0x0000000000000241 have been set
(0x0000000000003FFF previously) for channel 0x0F, LUN 0, command 0x12,
NetFn 0x04...
    Command functions 0x0000000000000241 have been set
(0x0000000000003FFF previously) for channel 0x0F, LUN 1, command 0x12,
NetFn 0x04...
    Command functions 0x0000000000000241 have been set
(0x0000000000003FFF previously) for channel 0x0F, LUN 2, command 0x12,
NetFn 0x04...
    Command functions 0x0000000000000241 have been set
(0x0000000000003FFF previously) for channel 0x0F, LUN 3, command 0x12,
NetFn 0x04...
#
```

Enable functions 1, 13 and 15 and disable all other (existing) functions for the command “Set LAN Configuration Parameters” (NetFn = 0xC, command code = 1) for channel 0 and LUN 1. This means allowing only the change of volatile destination type and volatile destination address for the channel 1 (LAN), for the commands arriving from IPMB (channel 0) on LUN 0.

```
# clia setfunctionpolicy 0xA002 0xC 0x1 0 1

Pigeon Point Shelf Manager Command Line Interpreter

    Command functions 0x000000000000A002 have been set
(0x0000000001FFFFFF previously) for channel 0x00, LUN 0, command 0x01,
NetFn 0x0C...
```

3.56 sethysteresis

3.56.1 Syntax

sethysteresis <IPMB-address> [<lun>:] <sensor_id> | <sensor_name > (pos | neg) <value>

3.56.2 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. The **<value>** is always treated as a raw value.

This command can also be issued on the backup Shelf Manager; in that case, the hysteresis values can only be set for sensors that are local to the backup Shelf Manager.

3.56.3 Examples

Set positive hysteresis for sensor #2 of the IPM controller at IPMB address FCh.

```
# clia sethysteresis FC 2 pos 10

Pigeon Point Shelf Manager Command Line Interpreter

     Positive hysteresis set successfully to 0xA, previous: 0x0
```

3.57 setipmbstate

3.57.1 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)

3.57.2 Purpose

This command enables/disables an IPMB link on the target IPM controller. The second argument defines the bus (IPMB-A or IPMB-B) to be enabled/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 context. In a bused environment, and in radial shelf for target IPM controllers other than an IPMB Hub, the argument **<link>** is not used. For an IPMB hub controller in a radial shelf, the argument **<link>** is optional.

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

This command can also be issued on the backup Shelf Manager; in that case, an IPMB link can only be enabled/disabled for an IPM controller that is local to the backup Shelf Manager.

3.57.3 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 an IPMB hub):

```
# clia setipmbstate 20 B 3 1

Pigeon Point Shelf Manager Command Line Interpreter

     Command executed successfully
```

3.58 setlanconfig

3.58.1 Syntax

setlanconfig <channel> <parameter-name> <additional-parameters>
setlanconfig <channel> <parameter-number> <additional-parameters>

3.58.2 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.

The following table lists names and numbers of LAN configuration parameters supported by the **setlanconfig** command:

Table 12 Names and Numbers of LAN Configuration Parameters Supported by the setlanconfig Command

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

PARAMETER NAME	NUMBER	DESCRIPTION
		default gateway in dotted decimal notation.
backup_gw_ip	14	A string value that contains the IP address of the backup gateway in dotted decimal notation.
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 6 hexadecimal bye values delimited by ':' symbols)
vlan_id	20	The flag that indicates whether Virtual LANs are enabled for the channel, and the Virtual LAN ID (a number in the range 1 to 4095, or the value 0, indicating that Virtual LAN IDs are not used for the channel)
vlan_priority	21	A number in the range 0 to 7 that specifies the packet Priority field, according to 802.1q
cs_priv_levels	24	The maximum privilege level associated with the Cipher Suite that is identified by the specified set selector.

3.58.3 *auth_enables*

3.58.3.1 Syntax

```
setlanconfig <channel> auth_enables <value1> <value2>
<value3> <value4> <value5>
setlanconfig <channel> 2 <value1> <value2> <value3>
<value4> <value5>
```

3.58.3.2 Purpose

This command sets the 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 following meaning of the bits:

- 0x01 None
- 0x02 MD2
- 0x04 MD5
- 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.

3.58.3.3 Examples

```
# clia setlanconfig 1 auth_enables 0 1 1 1 0

Pigeon Point Shelf Manager Command Line Interpreter

Authentication Type Enables set successfully

#
```

3.58.4 *ip*

3.58.4.1 Syntax

```
setlanconfig <channel> ip <value>
setlanconfig <channel> 3 <value>
```

3.58.4.2 Purpose

This command sets the IP address used by the channel. The value should represent an IP address in dotted decimal notation.

3.58.4.3 Examples

```
# clia setlanconfig 1 ip 172.16.2.203

Pigeon Point Shelf Manager Command Line Interpreter

IP set successfully

#
```

3.58.5 subnet_mask

3.58.5.1 Syntax

setlanconfig <channel> subnet_mask <value>
setlanconfig <channel> 6 <value>

3.58.5.2 Purpose

This command sets the IP subnet mask used by the channel. The value should represent a subnet mask in dotted decimal notation.

3.58.5.3 Examples

```
# clia setlanconfig 1 subnet_mask 255.255.255.0

Pigeon Point Shelf Manager Command Line Interpreter

Subnet Mask set successfully

#
```

3.58.6 ipv4_hdr_param

3.58.6.1 Syntax

setlanconfig <channel> ipv4_hdr_param <value1> <value2> <value3>
setlanconfig <channel> 7 <value1> <value2> <value3>

3.58.6.2 Purpose

This command sets the IP 4 header parameters for the Shelf Manager. They are represented as 3 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

3.58.6.3 Examples

```
# clia setlanconfig 1 ipv4_hdr_param 37 E0 11

Pigeon Point Shelf Manager Command Line Interpreter

IPv4 Header Parameters set successfully

#
```

3.58.7 arp_control

3.58.7.1 Syntax

setlanconfig <channel> arp_control <value>
setlanconfig <channel> 10 <value>

3.58.7.2 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 following meaning of the bits:

- 0x01 Enable Shelf Manager-generated Gratuitous ARPs
- 0x02 Enable Shelf Manager-generated ARP responses

Other bits are reserved and should be set to 0.

3.58.7.3 Examples

```
# clia setlanconfig 1 arp_control 3

Pigeon Point Shelf Manager Command Line Interpreter

BMC-generated ARP control set successfully

#
```

3.58.8 arp_interval

3.58.8.1 Syntax

setlanconfig <channel> arp_interval <value>
setlanconfig <channel> 11 <value>

3.58.8.2 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.

3.58.8.3 Examples

```
# clia setlanconfig 1 arp_interval 3.5

Pigeon Point Shelf Manager Command Line Interpreter

Gratuitous ARP interval set successfully

#
```

3.58.9 dft_gw_ip

3.58.9.1 Syntax

setlanconfig <channel> dft_gw_ip <value>
setlanconfig <channel> 12 <value>

3.58.9.2 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.

3.58.9.3 Examples

```
# clia setlanconfig 1 dft_gw_ip 172.16.2.100

Pigeon Point Shelf Manager Command Line Interpreter

Default Gateway Address set successfully

#
```

3.58.10 backup_gw_ip

3.58.10.1 Syntax

setlanconfig <channel> backup_gw_ip <value>
setlanconfig <channel> 14 <value>

3.58.10.2 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.

3.58.10.3 Examples

```
# clia setlanconfig 1 backup_gw_ip 172.16.2.100

Pigeon Point Shelf Manager Command Line Interpreter

Backup Gateway Address set successfully

#
```

3.58.11 community

3.58.11.1 Syntax

setlanconfig <channel> community <value>
setlanconfig <channel> 16 <value>

3.58.11.2 Purpose

This command sets the community string parameter used in PET traps. The value should be a string enclosed in double quotes.

3.58.11.3 Examples

```
# clia setlanconfig 1 community "Community"

Pigeon Point Shelf Manager Command Line Interpreter

Community string set successfully

#
```

3.58.12 destination_type

3.58.12.1 Syntax

setlanconfig <channel> destination_type <set-selector> <value1> <value2> <value3>
setlanconfig <channel> 18 <set-selector> <value1> <value2> <value3>

3.58.12.2 Purpose

This command sets the element of the destination table with the index **<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

3.58.12.3 Examples

```
# clia setlanconfig 1 destination_type 2 80 3 5

Pigeon Point Shelf Manager Command Line Interpreter

Destination Type set successfully

#
```

3.58.13 destination_address

3.58.13.1 Syntax

setlanconfig <channel> destination_address <set-selector> <gateway-sel> <IP-address> <MAC-address>
setlanconfig <channel> 19 <set-selector> <gateway-sel> <IP-address> <MAC-address>

3.58.13.2 Purpose

This command sets the element of the destination address table with the index **<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

3.58.13.3 Examples

```
# 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

#
```

3.58.14 vlan_id

3.58.14.1 Syntax

```
setlanconfig <channel> vlan_id disabled
setlanconfig <channel> vlan_id enabled <value>
setlanconfig <channel> 20 disabled
setlanconfig <channel> 20 enabled <value>
```

3.58.14.2 Purpose

This command enables or disables Virtual LAN support for the channel, and if enabled, specifies the Virtual LAN ID to use. If the parameter **disabled** is specified, the VLAN ID in LAN configuration parameters is set to 0.

3.58.14.3 Examples

```
# clia setlanconfig 1 vlan_id enabled 4

Pigeon Point Shelf Manager Command Line Interpreter

VLAN ID set successfully

#
```

3.58.15 vlan_priority

3.58.15.1 Syntax

```
setlanconfig <channel> vlan_priority <value>
setlanconfig <channel> 21 <value>
```

3.58.15.2 Purpose

This command sets the VLAN priority for the channel, that is, the value of the Priority field used in the VLAN 802.1q network packet headers.

3.58.15.3 Examples

```
# clia setlanconfig 1 vlan_priority 1

Pigeon Point Shelf Manager Command Line Interpreter

VLAN priority set successfully

#
```

3.58.16 cs_priv_levels

3.58.16.1 Syntax

setlanconfig <channel> cs_priv_levels <set-selector> <value>
setlanconfig <channel> 24 <set-selector> <value>

3.58.16.2 Purpose

This command sets the maximum privilege level for the Cipher Suite that is identified by the **<set-selector>**. Standard Cipher Suite IDs are defined in the IPMI specification version 2.0, section 22.15.2 and are in the range from 0 to 14. The parameter **<value>** specifies the desired maximum privilege level.

3.58.16.3 Examples

Set the maximum privilege level “Operator” for Cipher Suite 2:

```
# clia setlanconfig 1 cs_priv_levels 2 3

Pigeon Point Shelf Manager Command Line Interpreter

Cipher Suite Privilege Levels set successfully

#
```

3.59 setlocked

3.59.1 Syntax

```
setlocked <IPMB-address> <fru_id> 0 | 1
setlocked board <N> 0 | 1
setlocked shm <N> 0 | 1
setlocked power_supply <N> 0 | 1
setlocked pem <N> 0 | 1
setlocked fan_tray <N> 0 | 1
setlocked <IPMB-address> amc <M> 0 | 1
setlocked board <N> amc <M> 0 | 1
```

3.59.2 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 to state M2. 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 the 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.

3.59.3 Examples

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

```
# clia setlocked 9c 0 0

Pigeon Point Shelf Manager Command Line Interpreter

    Lock set successfully to 0x0

#
```

3.60 setpefconfig

3.60.1 Syntax

```
setpefconfig <parameter-name> <additional-parameters>
setpefconfig <parameter-number> <additional-parameters>
```

3.60.2 Purpose

This command sets a new value of the specified PEF configuration parameter. The following table lists names and numbers of PEF configuration parameters that can be set via this command.

Table 13 Names and Numbers of PEF Configuration Parameters Supported by the setpefconfig Command

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

Parameter Name	Number	Description
`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 17-2 of the IPMI specification version 2.0: 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 5 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.
`alert_string`	13	An alert string identified by the specified set selector.

Parameter Name	Number	Description
oem_filter	97	An OEM filter table entry identified by the specified set selector. Consists of the following 3 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.
pet_format	98	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.

3.60.3 control

3.60.3.1 Syntax

`setpefconfig control <value>`
`setpefconfig 1 <value>`

3.60.3.2 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 following meaning for the bits:

- 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.

3.60.3.3 Examples

```
# clia setpefconfig control 7

Pigeon Point Shelf Manager Command Line Interpreter

PEF control set successfully
```

```
#
```

3.60.4 action_control

3.60.4.1 Syntax

`setpefconfig action_control <value>`
`setpefconfig 2 <value>`

3.60.4.2 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 following meaning for the bits:

- 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

3.60.4.3 Examples

```
# clia setpefconfig action_control 3f

Pigeon Point Shelf Manager Command Line Interpreter

PEF action control set successfully
#
```

3.60.5 startup_delay

3.60.5.1 Syntax

`setpefconfig startup_delay <value>`
`setpefconfig 3 <value>`

3.60.5.2 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.

3.60.5.3 Examples

```
# clia setpefconfig startup_delay 45

Pigeon Point Shelf Manager Command Line Interpreter

PEF startup delay set successfully

#
```

3.60.6 alert_startup_delay

3.60.6.1 Syntax

`setpefconfig startup_delay <value>`
`setpefconfig 4 <value>`

3.60.6.2 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.

3.60.6.3 Examples

```
# clia setpefconfig alert_startup_delay 45

Pigeon Point Shelf Manager Command Line Interpreter

Alert startup delay set successfully

#
```

3.60.7 event_filter

3.60.7.1 Syntax

`setpefconfig event_filter <set-selector> <value1>` … **`<value19>`**
`setpefconfig 6 <set-selector> <value1>` … **`<value19>`**

3.60.7.2 Purpose

This command sets the element of the event filter table with the index **`<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 17-2 of the IPMI specification, version 2.0:

- 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

3.60.7.3 Examples

Setting event filter 2 to trigger an alert action when an IPM Controller at address 9Ch, 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

#
```

3.60.8 event_filter_data1

3.60.8.1 Syntax

`setpefconfig event_filter_data1 <set-selector> <value>`
`setpefconfig 7 <set-selector> <value>`

3.60.8.2 Purpose

This command sets the first byte of the element of the event filter table with the index **`<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 pre-configured 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.

3.60.8.3 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

#
```

3.60.9 alert_policy

3.60.9.1 Syntax

setpefconfig alert_policy <set-selector> <value1> <value2> <value3> <value4> <value5>
setpefconfig 9 <set-selector> <value1> <value2> <value3> <value4> <value5>

3.60.9.2 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 5 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

3.60.9.3 Examples

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

#
```

3.60.10 system_guid

3.60.10.1 Syntax

setpefconfig system_guid <guid-value>
setpefconfig 10 <guid-value>
setpefconfig system_guid none
setpefconfig 10 none

3.60.10.2 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.

3.60.10.3 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

#
```

3.60.11 alert_string_key

3.60.11.1 Syntax

setpefconfig alert_string_key <set-selector> <value1> <value2>
setpefconfig 12 <set-selector> <value1> <value2>

3.60.11.2 Purpose

This command sets the element of the alert string key table with the index **<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 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.

3.60.11.3 Examples

```
# clia setpefconfig alert_string_key 2 10 11

Pigeon Point Shelf Manager Command Line Interpreter

Alert string keys set successfully

#
```

3.60.12 *alert_string*

3.60.12.1 Syntax

setpefconfig alert_string <set-selector> <string-value>
setpefconfig 13 <set-selector> <string-value>

3.60.12.2 Purpose

This command sets the element of the alert string table with the index **<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.

3.60.12.3 Examples

```
# clia setpefconfig alert_string 2 "This string has
> a line feed inside."

Pigeon Point Shelf Manager Command Line Interpreter

Alert string set successfully

#
```

3.60.13 *oem_filter*

3.60.13.1 Syntax

setpefconfig oem_filter <set-selector> <value1> <value2> <value3>
setpefconfig 97 <set-selector> <value1> <value2> <value3>

3.60.13.2 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, also 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 3 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.

3.60.13.3 Examples

```
# 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
```

3.60.14 pet_format

3.60.14.1 Syntax

setpefconfig pet_format <format>
setpefconfig 98 <format>

3.60.14.2 Purpose

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 of **<format>** parameter 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.

3.60.14.3 Examples

```
# clia setpefconfig pet_format 0

Pigeon Point Shelf Manager Command Line Interpreter

Platform Event Trap format set successfully

#
```

3.61 setpowerlevel

3.61.1 Syntax

setpowerlevel <IPMB-address> <fru_id> [<pwr_lvl>|OFF] [COPY]

<IPMB-address> <fru_id> can be replaced with any of the following alternatives:
board <N>
shm <N>
power_supply <N>
pem <N>
fan_tray <N>

3.61.2 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. 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" desired power levels to present power levels (see the AdvancedTCA specification for background). If this parameter is not specified, the command does not copy desired to present power levels.

3.61.3 Examples

Turn off power for the board at 84h:

```
# clia setpowerlevel 84 0 OFF

Pigeon Point Shelf Manager Command Line Interpreter

    Operation completed with status = 0x0
```

3.62 setsensordata

3.62.1 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>]`

`<IPMB-address>` can be replaced with either of the following alternatives:
`board <N>`
`shm <N>`

3.62.2 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.

3.62.3 Examples

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.

```
# 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

#
# clia setsensordata 10 10 assertion 2

Pigeon Point Shelf Manager Command Line Interpreter

    Sensor data set successfully
#
# 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
```

3.63 setsensoreventenable

3.63.1 Syntax

setsensoreventenable <IPMB-address> <sensor-name> <global> [<assertion_events> [<deassertion_events>]]
setsensoreventenable <IPMB-address> [<lun>:]<sensor-number> <global> [<assertion_events> [<deassertion_events>]]

<IPMB-address> can be replaced with any of the following alternatives:
board <N>
shm <N>

3.63.2 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 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 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 Event Enable”.

The parameter **<global>** directly corresponds to the second byte of the command request parameters and is a bit mask with the following meanings of the bits:

- [7] - 0b = disable all Event Messages from this sensor (optional; does not impact individual enable/disable status)
- [6] - 0b = disable scanning on this sensor (optional)
- [5:4] - 00b = do not change individual enables
- 01b = enable selected event messages
- 10b = disable selected event messages
- 11b = reserved
- [3:0] - reserved

The parameters **<assertion_events>** and **<deassertion_events>** are 16-bit bit masks representing individual events to enable or disable, as follows:

For threshold-based sensors:

- [15:12] – reserved, must be 0000

- [11] - 1b = select event for upper non-recoverable going high
- [10] - 1b = select event for upper non-recoverable going low
- [9] - 1b = select event for upper critical going high
- [8] - 1b = select event for upper critical going low
- [7] - 1b = select event for upper non-critical going high
- [6] - 1b = select event for upper non-critical going low
- [5] - 1b = select event for lower non-recoverable going high
- [4] - 1b = select event for lower non-recoverable going low
- [3] - 1b = select event for lower critical going high
- [2] - 1b = select event for lower critical going low
- [1] - 1b = select event for lower non-critical going high
- [0] - 1b = select event for lower non-critical going low

For discrete sensors:

- [15] – reserved, must be 0
- [14] - 1b = select event for state bit 14
- [13] - 1b = select event for state bit 13
- [12] - 1b = select event for state bit 12
- [11] - 1b = select event for state bit 11
- [10] - 1b = select event for state bit 10
- [9] - 1b = select event for state bit 9
- [8] - 1b = select event for state bit 8
- [7] - 1b = select event for state bit 7
- [6] - 1b = select event for state bit 6
- [5] - 1b = select event for state bit 5
- [4] - 1b = select event for state bit 4
- [3] - 1b = select event for state bit 3
- [2] - 1b = select event for state bit 2
- [1] - 1b = select event for state bit 1
- [0] - 1b = select event for state bit 0

This command can be used both to control individual event enables and to disable/enable sensor scanning and event generation globally.

In the first case, the two most significant bits of the parameter **`<global>`** should be set. Typical values are:

- 0xD0 to enable events specified by the masks **`<assertion_events>`** and **`<deassertion_events>`**
- 0xE0 to disable events specified by the masks **`<assertion_events>`** and **`<deassertion_events>`**

In the second case, the parameters **`<assertion_events>`** and **`<deassertion_events>`** can be omitted and the parameter **`<global>`** controls the global attributes of the sensor. Only bits 7 and 6 are used, bits 5:0 are set to 0 in that case. Typical values of the **`<global>`** parameter in that case are:

- 0 to disable both sensor scanning and event generation
- 0x40 to enable sensor scanning but disable event generation
- 0xC0 to enable both sensor scanning and event generation

However the combination of the two cases above is also possible. More information can be found in the section of the IPMI specification that is dedicated to the command “Set Sensor Event Enable”.

This command can also be issued on the backup Shelf Manager; in that case, the event enable mask is only set for sensors that are local to the backup Shelf Manager.

3.63.3 Examples

Enable the “Lower Non-Critical Going Low” event on the temperature sensor "Local Temp" on the IPM controller FEh.

```
# clia setsensoreventenable fe "Local Temp" 0xD0 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)
    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 0xD0 0x01 0x00

Pigeon Point Shelf Manager Command Line Interpreter

    Event enable mask set successfully
#
```

3.64 setthreshold

3.64.1 Syntax

setthreshold <IPMB-address> <sensor-name> <threshold-type> [-r] <value>
setthreshold <IPMB-address> [<lun>:]<sensor-number> <threshold-type> [-r] <value>

<IPMB-address> can be replaced with any of the following alternatives:
board <N>
shm <N>

3.64.2 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 (e.g. in Volts for voltage sensors or in Celsius degrees 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.)

This command can also be issued on the backup Shelf Manager; in that case, threshold values can only be set for sensors that are local to the backup Shelf Manager.

3.64.3 Examples

Set the upper non-critical threshold value for the temperature sensor “emulated temp” on IPM controller 9Ch 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

#
```

3.65 shelf

3.65.1 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**
- **BDSelGrounded <slot number> 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**
- **shm_cfg_params**
- **board_lan_cfg_params**
- **cold_sensitive**

3.65.2 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 it.

3.65.3 Displaying Shelf FRU Information

3.65.3.1 Syntax

shelf [cooling_state | fans_state | address_table | power_distribution | power_management |pci_connectivity | ha_connectivity | h110_connectivity | point-to-point_connectivity | shm_cfg_params | board_lan_cfg_params]

3.65.3.2 Purpose

The variants of the **shelf** command show key Shelf FRU information, plus selected current operating data for the shelf. The type of the information shown is specified in the subcommand.

The following table lists the subcommands and parameters used to display shelf information:

Table 14 Parameters Supported by the shelf Command

Parameter Name	Description
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. In addition, a message is printed if any of the fan trays listed in the Address Table are not operational (missing or deactivated).
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 Type, and Site Number for each of them.

PARAMETER NAME	DESCRIPTION
power_distribution (can be abbreviated to **pd**)	Shows the following information 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
power_management (can be abbreviated to **pm**)	Shows 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)	Shows the Shelf PCI Connectivity record in the Shelf FRU Info. The following information is provided: PCI Slot Descriptor IDSEL Connection Segment ID Extended PCI Slot Descriptor Geographic Address Interface Number System Slot Capable
ha_connectivity (can be abbreviated to ha)	Shows the Shelf HAConnectivity record in the Shelf FRU Info. The following information is provided: Radial Connectivity Support
h110_connectivity (can be abbreviated to **h110c**)	Shows 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**)	Shows 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
shm_cfg_params	Shows the PPS Shelf Manager Configuration Record(s) in the Shelf FRU Info in a human readable format.
board_lan_cfg_params	Shows the PPS Board/AMC LAN Configuration Parameters Record(s) in the Shelf FRU Info in a human readable format.

For the subcommands **`cooling_state`**, **`fans_state`**, and **`power_management`**, the verbosity option **`-v`** is available. It should be entered before the subcommand: **`clia shelf -v cooling_state`**. In verbose mode, the subcommands **`cooling_state`** or **`fans_state`** 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). The verbose variant of the **`power_management`** subcommand displays the amount of power currently assigned to each of the FRUs covered by FRU Power Descriptors in the Shelf FRU Info.

3.65.3.3 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)
    1 fan tray(s) (out of 3) are not operational
#
```

Get address table.

```
# clia shelf address_table

Pigeon Point Shelf Manager Command Line Interpreter

    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
```

```
#
```
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 -v 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
        Currently Assigned Power: 70 Watts

    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
        Currently Assigned Power: 0 Watts
```

```
    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
        Currently Assigned Power: 0 Watts

    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
        Currently Assigned Power: 0 Watts

    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
        Currently Assigned Power: 0 Watts

    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
        Currently Assigned Power: 0 Watts

    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
        Currently Assigned Power: 0 Watts

    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
        Currently Assigned Power: 0 Watts

    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
        Currently Assigned Power: 0 Watts

    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
        Currently Assigned Power: 0 Watts

    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
        Currently Assigned Power: 0 Watts

    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
        Currently Assigned Power: 0 Watts

    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
        Currently Assigned Power: 0 Watts

    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
        Currently Assigned Power: 0 Watts

    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
        Currently Assigned Power: 0 Watts

    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
        Currently Assigned Power: 20 Watts
#
```

Get the PPS Shelf Manager Configuration records.

```
# clia shelf shm_cfg_params

Pigeon Point Shelf Manager Command Line Interpreter

   PPS Shelf Manager Configuration Record

CARRIER = ACB
ALLOW_CLEARING_CRITICAL_ALARM = TRUE
INITIAL_FAN_LEVEL = 8
MIN_FAN_LEVEL = 3
PROPAGATE_RMCP_ADDRESS = TRUE
###SWITCHOVER_TIMEOUT_ON_BROKEN_LINK = 10
IPMB_LINK_ISOLATION_TIMEOUT = 40

DEFAULT_RMCP_IP_ADDRESS2 = 172.16.2.168
GET_SHM_CONF_FROM_SHELF_FRU = TRUE
HPDL = TRUE

TIMESERVER = 192.168.1.253
TIMEPROTO = ntp


BOARD_LAN_PARAMETERS_CHANNEL_LIST = 1245
#
```

Get the PPS Board/AMC LAN Configuration Parameter records.

```
#clia shelf board_lan_cfg_params

Pigeon Point Shelf Manager Command Line Interpreter

   PPS Board/AMC LAN Configuration Parameters Record

2 = 192.168.1.200, 255.255.255.0, 192.168.1.1
2, 1 = 192.168.1.201, 255.255.255.0, 192.168.1.1

#
```

3.65.4 Modifying Maximum External Available Current

3.65.4.1 Syntax

`shelf maxcurrent [<feed>] <current>`

3.65.4.2 Purpose

This command sets the Maximum External Available Current for the specified feed number and updates all known instances of Shelf FRU Info in the shelf. If the **`<feed>`** parameter is omitted, the value is set for the first feed (feed 0) in the Shelf FRU Info.

The parameter **`<feed>`** is a 0-based feed number in the Shelf FRU Info based on the sequential order of the descriptor for that feed.

The parameter **`<current>`** is the desired current value in Amps.

If, as the result of execution of this command, the currently used power exceeds the available power provided by the feed, some FRUs are deactivated until the currently used power becomes less than or equal to the available power. Deactivation takes place in the reverse order of activation (according to the Shelf Activation and Power Management Record).

3.65.4.3 Examples

Changing the Maximum Available External Current for Feed 0 from 50 A to 99 A.

```
# 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: 240.000 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: 4009.500 Watts
        Currently Used Power: 240.000 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

#
```

3.65.5 Modifying Minimum Expected Operating Voltage

3.65.5.1 Syntax

`shelf minvoltage [<feed>] <voltage>`

3.65.5.2 Purpose

This command sets the Minimum Expected Operating Voltage for the specified feed number and updates all known Shelf FRU Info instances in the shelf. If the **`<feed>`** parameter is omitted, the value is set for the first feed (feed 0) in the Shelf FRU Info.

The parameter **`<feed>`** is a 0-based feed number in the Shelf FRU Info based on the sequential order of the description of that feed.

The parameter **`<voltage>`** is the desired value.

If, as the result of execution of this command, the currently used power exceeds the available power provided by the feed, some FRUs are deactivated until the currently used power becomes less than or equal to the available power. Deactivation takes place in the reverse order of activation (according to the Shelf Activation and Power Management Record).

3.65.5.3 Examples

Changing the Minimum Expected Operating Voltage for the Feed 0 -40.5 to -59

```
# 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: 240.000 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: 50.0 Amps
        Maximum Internal Current: Not specified
        Minimum Expected Operating Voltage: -59.0 Volts
        Actual Power Available: 2950.000 Watts
        Currently Used Power: 240.000 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

#
```

3.65.6 Modifying Shelf Manager Controlled Activation Flag

3.65.6.1 Syntax

```
shelf activation <hardware addr> <fru_id> [1|0]
shelf activation board <N> [1|0]
shelf activation board all [1|0]
shelf activation power_supply <N> [1|0]
shelf activation pem <N> [1|0]
```

`shelf activation fan_tray <N> [1|0]`

3.65.6.2 Purpose

These variants of the **`shelf`** command display or change 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. The final parameter enables (when the value is **`1`**) or disables (when the value is **`0`**) Shelf Manager Controlled Activation for the specified FRU of the specified IPM controller.

3.65.6.3 Examples

Enabling Shelf Manager Controlled Activation on an IPM Controller with hardware address 42h (IPMB address 84h).

```
# 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

#
```

3.65.7 Modifying Shelf Manager Controlled Deactivation Flag

3.65.7.1 Syntax

```
shelf deactivation <hardware addr> <fru_id> [1|0]
shelf deactivation board <N> [1|0]
shelf deactivation board all [1|0]
shelf deactivation power_supply <N> [1|0]
shelf deactivation pem <N> [1|0]
shelf deactivation fan_tray <N> [1|0]
```

3.65.7.2 Purpose

These variants of the **`shelf`** command display or change 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 Deactivation 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. The final parameter enables (when the value is **0**) or disables (when the value is **1**) Shelf Manager Controlled Deactivation for the specified FRU of the specified IPM controller. Note the unusual use of a zero parameter value to enable a function; this is the approach used in the ATCA specification for this function. The approach is preserved here for consistency.

3.65.7.3 Examples

Enabling Shelf Manager Controlled Deactivation on an IPM Controller with hardware address 42h (IPMB address 84h).

```
# 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: 2s
    Hw Address: 41, FRU ID: fe, Maximum FRU Power Capabilities: 200
Watts
        Shelf Manager Controlled Auto-Activation: Disabled
        Shelf Manager Controlled Auto-Deactivation: Enabled
        Delay Before Next Power On: 2.2 seconds

    Hw Address: 42, FRU ID: fe, Maximum FRU Power Capabilities: 200
Watts
        Shelf Manager Controlled Auto-Activation: Disabled
        Shelf Manager Controlled Auto-Deactivation: Disabled
        Delay Before Next Power On: 2.2 seconds

#

# clia shelf deactivation 42 0xfe 0

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 Auto-Activation: Disabled
        Shelf Manager Controlled Auto-Deactivation: Enabled
Delay Before Next Power On: 2.2 seconds

    Hw Address: 42, FRU ID: fe, Maximum FRU Power Capabilities: 200
Watts
        Shelf Manager Controlled Auto-Activation: Disabled
        Shelf Manager Controlled Auto-Deactivation: Enabled
        Delay Before Next Power On: 2.2 seconds

#
```

3.65.8 Modifying Shelf Manager BDSelGrounded Flag

3.65.8.1 Syntax

`shelf bdselgrounded <slot number>[1|0]`
`shelf bdselgrounded board <N> [1|0]`
`shelf bdselgrounded board all [1|0]`

3.65.8.2 Purpose

These variants of the **`shelf`** command allow specifying whether the BD_SEL# signal is grounded for a slot. Some shelves may have BD_SEL# lines grounded for some slots, while operational for other slots. If BD_SEL# line is grounded, it is not possible for the Shelf Manager to discover if a board is present in the slot or turn on/off power for this slot. In the case of a grounded BD_SEL# line, the Shelf Manager uses a different control algorithm for the slot; thus it is important to have this information.

The BD SEL# Grounded flags for slots are stored in Shelf FRU Information in the HA Connectivity record. The command modifies this flag only for already existing entries in that record. This command also updates the cached version of the Shelf FRU Information used by the Shelf Manager.

The parameter **`<slot number>`** is the ordinary number that may be specified in either decimal or hexadecimal form.

3.65.8.3 Examples

Configuring normal BD SEL# signal operation for slot 2.

```
# clia shelf bdselgrounded board all

Pigeon Point Shelf Manager Command Line Interpreter

    Slot #  1, "Normal BD_SEL# operation"
    Slot #  2, "BD_SEL# is grounded for this slot by hardware"
    Slot #  3, "Normal BD_SEL# operation"
    Slot #  4, "Normal BD_SEL# operation"
    Slot #  5, "Normal BD_SEL# operation"
    Slot #  6, "BD_SEL# is grounded for this slot by hardware"
```

```
    Slot #  7, "Normal BD_SEL# operation"
    Slot #  8, "Normal BD_SEL# operation"
    Slot #  9, "Normal BD_SEL# operation"
    Slot # 10, "Normal BD_SEL# operation"
    Slot # 11, "Normal BD_SEL# operation"
    Slot # 12, "Normal BD_SEL# operation"
    Slot # 13, "Normal BD_SEL# operation"
    Slot # 14, "Normal BD_SEL# operation"
    Slot # 15, "Normal BD_SEL# operation"
    Slot # 16, "BD_SEL# is grounded for this slot by hardware"
    Slot # 17, "Normal BD_SEL# operation"
    Slot # 18, "Normal BD_SEL# operation"
    Slot # 19, "Normal BD_SEL# operation"
    Slot # 20, "Normal BD_SEL# operation"
    Slot # 21, "BD_SEL# is grounded for this slot by hardware"

#

# clia shelf bdselgrounded b 2 0

Pigeon Point Shelf Manager Command Line Interpreter

    Updating Shelf FRU Info, slot # 2
    Wrote Information to the Shelf FRU

#

# clia shelf bdselgrounded board all

Pigeon Point Shelf Manager Command Line Interpreter

    Slot #  1, "Normal BD_SEL# operation"
    Slot #  2, "Normal BD_SEL# operation"
    Slot #  3, "Normal BD_SEL# operation"
    Slot #  4, "Normal BD_SEL# operation"
    Slot #  5, "Normal BD_SEL# operation"
    Slot #  6, "BD_SEL# is grounded for this slot by hardware"
    Slot #  7, "Normal BD_SEL# operation"
    Slot #  8, "Normal BD_SEL# operation"
    Slot #  9, "Normal BD_SEL# operation"
    Slot # 10, "Normal BD_SEL# operation"
    Slot # 11, "Normal BD_SEL# operation"
    Slot # 12, "Normal BD_SEL# operation"
    Slot # 13, "Normal BD_SEL# operation"
    Slot # 14, "Normal BD_SEL# operation"
    Slot # 15, "Normal BD_SEL# operation"
    Slot # 16, "BD_SEL# is grounded for this slot by hardware"
    Slot # 17, "Normal BD_SEL# operation"
    Slot # 18, "Normal BD_SEL# operation"
    Slot # 19, "Normal BD_SEL# operation"
    Slot # 20, "Normal BD_SEL# operation"
    Slot # 21, "BD_SEL# is grounded for this slot by hardware"

#
```

3.65.9 Modifying Maximum FRU Power Capability

3.65.9.1 Syntax

```
shelf pwrcapability <hardware addr> <fru_id> <value>
shelf pwrcapability shm <N> <value>
shelf pwrcapability board <N> <value>
shelf pwrcapability power_supply <N> <value>
shelf pwrcapability pem <N> <value>
shelf pwrcapability fan_tray <N> <value>
```

3.65.9.2 Purpose

These variants of the **shelf** command change 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 a 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.

If, as the result of execution of this command, the currently used power for the slot exceeds the power limit for that slot, some (or all) FRUs associated with the slot are deactivated until the currently used power becomes less than or equal to the power limit. If several FRUs are associated with the slot, deactivation takes place in the reverse order of FRU IDs.

3.65.9.3 Examples

Setting Maximum FRU Power Capability on an IPM Controller with hardware address 42h (IPMB address 84h) 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
#
```

3.65.10 Modifying Delay Before Next Power On

3.65.10.1 Syntax

shelf pwrdelay <hardware addr> <fru_id> <value>
shelf pwrdelay shm <N> <value>
shelf pwrdelay board <N> <value>
shelf pwrdelay power_supply <N> <value>
shelf pwrdelay pem <N> <value>
shelf pwrdelay fan_tray <N> <value>

3.65.10.2 Purpose

These variants of the **shelf** command change 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 a 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.

3.65.10.3 Examples

Setting Delay Before Next Power On for an IPM Controller with hardware address 42h (IPMB address 84h) 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

#
```

3.65.11 Modifying Allowance for FRU Activation Readiness

3.65.11.1 Syntax

`shelf allowance <value>`

3.65.11.2 Purpose

This variant of the **`shelf`** 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.

3.65.11.3 Examples

Setting 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

#
```

3.65.12 Reorder the FRU Activation and Power Descriptors

3.65.12.1 Syntax

shelf pwrreorder <hardware addr 1> <fru_id 1> before/after <hardware addr 2> <fru_id 2>

As usual, **<hardware addr x> <fru_id x>** can be replaced with any the following alternatives:
shm <N>
board <N>
power_supply <N>
pem <N>
fan_tray <N>

3.65.12.2 Purpose

This variant of the **shelf** 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 a 7-bit hardware address in hexadecimal format of the descriptor that needs to be moved to a new place.

The parameter **<fru_id 1>** is a FRU ID in hexadecimal format of the descriptor that needs to be moved to a new place; 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 or after which the **<hardware addr 1> <fru_id 1>** descriptor should be placed.

3.65.12.3 Examples

Placing the descriptor for an IPM Controller with hardware address 42h (IPMB address 84h) before the descriptor for an IPM Controller with hardware address 41h (IPMB address 82h).

```
# 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

#
```

3.65.13 Refresh the Shelf FRU Info

3.65.13.1 Syntax

shelf info_refresh

3.65.13.2 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, 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 Info (in volatile memory) which is used for any updating of Shelf FRU Info sources and is treated as the sole source of the Shelf FRU information. Thus, all Shelf FRU Info related operations work with the master copy and changes of the master copy are automatically propagated to all Shelf FRU Info 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 the restart of the shelf.

3.65.13.3 Examples

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

#
```

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

#
```

3.65.14 Updating the Shelf FRU Info Storage Devices

3.65.14.1 Syntax

`shelf info_force_update`

3.65.14.2 Purpose

This command causes a check of the Shelf FRU Info source devices and copying the contents of the Shelf FRU Info master copy to all of them. This command is useful in the case of a conflict between the Shelf FRU Info 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 will resume being propagated to the SEEPROMs.

This command initiates an update of the Shelf FRU Info source devices in an asynchronous fashion.

3.65.14.3 Examples

```
# clia shelf info_force_update

Pigeon Point Shelf Manager Command Line Interpreter

Starting the Shelf FRU Info source device update

#
```

3.65.15 Modifiyng Enable Flag for an Entry in the PPS Cold-Sensitive FRU List Record

3.65.15.1 Syntax

```
shelf cold_sensitive <hardware addr> <fru_id> [1|0]
shelf cold_sensitive power_supply <N> [1|0]
```

```
shelf cold_sensitive pem <N> [1|0]
shelf cold_sensitive fan_tray <N> [1|0]
shelf cold_sensitive board <M> amc <N> [1|0]
shelf cold_sensitive <addr> amc <N> [1|0]
shelf cold_sensitive board <N> [1|0]
```

3.65.15.2 Purpose

This variant of the **shelf** command changes the Enable flag for the specified FRU of the specified IPM controller in the PPS Cold-Sensitive FRU List record. The command modifies this flag only for already existing entries in the PPS Cold-Sensitive FRU List record. This command also updates the cached version of the Shelf FRU Information used by the Shelf Manager.

The parameter **<hardware addr>** is a 7-bit hardware address in hexadecimal format. The parameter **<fru_id>** is a FRU ID in hexadecimal format; a value of 0xFE indicates all FRUs at that hardware address. The final parameter enables (when the value is **1**) or disables (when the value is **0**) the entry in the PPS Cold-Sensitive FRU List record for the specified FRU of the specified IPM controller.

3.65.15.3 Examples

Enable an entry in the PPS Cold-Sensitive FRU List record

```
# clia shelf cold_sensitive 0x41 0xfe 1

Pigeon Point Shelf Manager Command Line Interpreter

    Updating Shelf FRU Info
    Cached information updated
```

Disable an entry in the PPS Cold-Sensitive FRU List record

```
# clia shelf cold_sensitive 0x41 0xfe 0

Pigeon Point Shelf Manager Command Line Interpreter

    Updating Shelf FRU Info
    Cached information updated
```

3.66 shelfaddress

3.66.1 Syntax

shelfaddress [<up to 20 characters of the shelf address>]
shelfaddress -x <byte1> … <byteN>

3.66.2 Purpose

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

3.66.3 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 Interpreter

    Shelf Address Info set successfully
```

```
#
# clia shelfaddress

Pigeon Point Shelf Manager Command Line Interpreter

    Shelf Address Info: " 0x01 0x02 0x03 0x04 0x05 "

#
```

3.67 shmstatus

3.67.1 Syntax

shmstatus

3.67.2 Purpose

This command returns the Shelf Manager status in redundant configurations: Active or Backup In verbose mode, the command reports a more detailed picture: status of the Shelf FRU Info, 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 Info and successfully initializes its RMCP interface.
- on the backup Shelf Manager if it successfully received the redundancy state information from the active Shelf Manager.

3.67.3 Examples

```
# clia shmstatus -v

Pigeon Point Shelf Manager Command Line Interpreter

Host: "Active"
Ready For Operation: Yes
Detailed State Flags: "Shelf FRU Found" "RMCP Up" "Backup Healthy"

#
```

3.68 showunhealthy

3.68.1 Syntax

`showunhealthy`

3.68.2 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”. In CompactPCI shelves, this command checks Board, Fan Tray and Power Supply healthy status bits as well.

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.

3.68.3 Examples

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.

#
```

3.69 switchover

3.69.1 Syntax

`switchover [-force]`

3.69.2 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.

This command with the option **`-force`** can be executed only on the backup instance of the Shelf Manager and performs forced switchover. That is, the backup Shelf Manager immediately turns on the hardware Active bit and becomes active without any negotiations with the active Shelf Manager. The active Shelf Manager (if still alive) immediately reboots in that case.

3.69.3 Examples

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.

#
```

3.70 terminate

3.70.1 Syntax

`terminate [-reboot]`

3.70.2 Purpose

This command terminates the Shelf Manager. Also, it causes the ShMM to unconditionally reboot if the option **`-reboot`** is specified.

If the option **`-reboot`** is omitted, this command terminates the Shelf Manager without rebooting the ShMM.

3.70.3 Examples

Terminate the Shelf Manager on ShMM-500 without rebooting the ShMM.

```
# clia terminate
   Terminating the Shelf Manager.

#
```

3.71 user

3.71.1 Syntax

`user [<subcommand>]`

The following subcommands are supported:

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

3.71.2 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.

3.71.3 Displaying User Information

3.71.3.1 Syntax

`user [-v] [<user id>]`

3.71.3.2 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.

3.71.3.3 Examples

```
# clia user -v
```

```
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"
```

3.71.4 Adding a New User

3.71.4.1 Syntax

user add <user id> <user name> <channel access flags> <privilege level> <password>

3.71.4.2 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 returns an error if the specified user does not exist. Command parameters have the following meaning:

<user id> - is a valid user ID;
<user name> - is a user name (which is truncated to the 16 characters without notice);
<channel access flags> - is the first byte of the SetUserInfo commands (only bits 4,5,6 are meaningful)
  bit 6 – IPMI messaging enabled,
  bit 5 – Link authentication enabled,
  bit 4 – Restricted to callback
<privilege level> - is the user privilege level
<password> - is a password (which is truncated to the 16 characters without notice)

3.71.4.3 Examples

Adding 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"
```

3.71.5 Deleting a User

3.71.5.1 Syntax

user delete <user id>

3.71.5.2 Purpose

This command deletes the user specified by the **<user id>**.

3.71.5.3 Examples

Deleting the user with user ID = 10.

```
# 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"

    10: "root2"
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"

# clia user delete 10

Pigeon Point Shelf Manager Command Line Interpreter

    User 10 deleted 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"
```

3.71.6 Enabling and Disabling a User

3.71.6.1 Syntax

`user enable <user id> 1 | 0`

3.71.6.2 Purpose

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

- **0** - disable the specified user
- **1** - enable the specified user

3.71.6.3 Examples

Disabling and enabling user with user ID 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 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"
```

3.71.7 Modifying a User Name

3.71.7.1 Syntax

user name <user id> <user name>

3.71.7.2 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:

<user id> - is a valid user ID ;

<user name> - is a user name (which will be truncated to 16 characters without notice)

3.71.7.3 Examples

Changing 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"
```

3.71.8 Modifying a User's Password

3.71.8.1 Syntax

user passwd <user id> <password>

3.71.8.2 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:

<user id> - is the valid user ID;
<password> - is the user password (which will be truncated to 16 characters without any notice)

3.71.8.3 Examples

Changing the password of user ID 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"
```

3.71.9 Modify Channel Access Settings

3.71.9.1 Syntax

user channel <user id> <channel number> <flags> <privilege level>

3.71.9.2 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:

<user id> - is the valid user ID;
<channel number> - is the channel number;
<flags> - is the first byte of the "Set User Info" commands (only bits 4,5,6 are meaningful)
bit 6 – IPMI messaging enabled,
bit 5 – Link authentication enabled,
bit 4 – Restricted to callback
<privilege level> - is the user privilege level

3.71.9.3 Examples

Changing 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"
```

3.72 variable

3.72.1 Syntax

```
variable <name>|<indexed-name>
variable <name>|<indexed-name> <value>
variable <name> total
```

An indexed name can designate a single array item or a range of array items:

```
<indexed-name> := <name>'[' <number> ']'
<indexed-name> := <name>'[' <number> ':' <number> ']'
```

3.72.2 Purpose

This command shows or sets the value of the HPDL variable specified by **<name>**.

When issued without **<value>**, the command shows the current value of the variable. This value is an integer. It is shown both in decimal and hexadecimal notation. If the variable is an array, values of all elements of the array are shown. Also, an indexed notation (e.g. **'$VAR[2]'**) can be used to get the value of a single array element.

Including a **<value>** parameter allows setting the specified variable to a new value. For arrays, the indexed notation must be used to address a single element of the array.

An extended indexed notation (with two indices separated by a colon) can be used to show a range of elements in an array or assign a single value to a range of elements in an array. An example of the extended notation is **'$VAR[2:4]'**.

If the variable is an array, it is also possible to get the sum of all array elements using the **total** keyword. The indexed notation is not applicable in this case.

Since the variable name normally contains a **$** character according to HPDL rules, it must be enclosed in single quotes in a Linux shell command line, to prevent interpretation of the **$** character by the shell.

The variable name may be qualified by the name of the containing scope (e.g. **'Carrier_FRU.$VAR'**) or may be left unqualified. In the latter case, the command interpreter looks for the variable definition across the entire HPDL module, starting from the top of the object hierarchy, and uses the first definition it finds.

3.72.3 Examples

Assume that the carrier HPDL definition file contains the following lines defining a single scalar variable **$VAR** and a vector **$VEC** of 10 elements (where indexing starts from 0):

```
    FRUS {
        FRU Carrier_FRU {
            ...
            INITIAL_ACTION $VAR = 0, VECTOR( $VEC, 10 );
            ...
        }
    }
```

Then the following command gets the initial value of the variable **$VAR**:

```
# clia variable '$VAR'

Pigeon Point Shelf Manager Command Line Interpreter

$VAR == 0 (0x0)
#
```

Changing the value of the variable and reading it back using the qualified name:

```
# clia variable '$VAR' 2

Pigeon Point Shelf Manager Command Line Interpreter

Operation complete

# clia variable 'Carrier_FRU.$VAR'

Pigeon Point Shelf Manager Command Line Interpreter

Carrier_FRU.$VAR == 2 (0x2)
```

Changing several items of the array **$VEC**, using indexed expressions:

```
# clia variable '$VEC[0]' 1

Pigeon Point Shelf Manager Command Line Interpreter

Operation complete

# clia variable '$VEC[1]' 2

Pigeon Point Shelf Manager Command Line Interpreter

Operation complete

# clia variable '$VEC[2]' 3

Pigeon Point Shelf Manager Command Line Interpreter

Operation complete

# clia variable '$VEC[3:9]' 4

Pigeon Point Shelf Manager Command Line Interpreter

Operation complete
```

Showing the values of all elements of the vector **$VEC**:

```
# clia variable '$VEC'

Pigeon Point Shelf Manager Command Line Interpreter

$VEC: 10 items
$VEC[0] == 1 (0x1)
$VEC[1] == 2 (0x2)
$VEC[2] == 3 (0x3)
$VEC[3] == 4 (0x4)
$VEC[4] == 4 (0x4)
$VEC[5] == 4 (0x4)
$VEC[6] == 4 (0x4)
$VEC[7] == 4 (0x4)
$VEC[8] == 4 (0x4)
$VEC[9] == 4 (0x4)
```

Showing the sum of the values in the vector **$VEC**:

```
# clia variable '$VEC' total

Pigeon Point Shelf Manager Command Line Interpreter

TOTAL($VEC) == 34 (0x22)
#
```

3.73 version

3.73.1 Syntax

version

3.73.2 Purpose

This command shows the version information for the Shelf Manager software. This command can also be issued on the backup Shelf Manager.

3.73.3 Examples

```
# clia version

Pigeon Point Shelf Manager Command Line Interpreter

Pigeon Point Shelf Manager ver. 3.2.0
Pigeon Point and the stylized lighthouse logo are trademarks of Pigeon
Point Systems.
Copyright (c) 2002-2013 Pigeon Point Systems
All rights reserved
Build date/time: May 31, 2013 16:39:37
Carrier: PPS
Carrier subtype: 0; subversion: 0
#
```

If the shelf is HPDL-based, this command also shows the information about the cooling management library that is currently used, its version information and the product ID strings from the carrier and chassis HPDL modules:

```
# clia version

Pigeon Point Shelf Manager Command Line Interpreter

Pigeon Point Shelf Manager ver. 3.2.0
Pigeon Point and the stylized lighthouse logo are trademarks of Pigeon
Point Systems.
Copyright (c) 2002-2013 Pigeon Point Systems
All rights reserved
Build date/time: May 31 2013 16:50:22
Carrier: HPDL/ACB
Carrier subtype: 0; subversion: 0
Cooling Management Library: libcooling_acb.so; Version: 3.2.0
Carrier Product ID: ACB-IV
Chassis Product ID: Standard 14-slot
#
```

Starting with release 2.6.4, support for an optional fourth level of the version number has been added to the Shelf Manager. This is a number in the range from 1 to 127. This number is defined for a specific build of the Shelf Manager; by default, the fourth level is absent. If a specific build of the Shelf Manager has the fourth level of the version defined, the output of this command looks like this:

```
# clia version

Pigeon Point Shelf Manager Command Line Interpreter

Pigeon Point Shelf Manager ver. 3.2.0
Pigeon Point and the stylized lighthouse logo are trademarks of Pigeon
Point Systems.
Copyright (c) 2002-2013 Pigeon Point Systems
All rights reserved
Build date/time: May 31, 2013 16:39:37
Carrier: PPS
Carrier subtype: 0; subversion: 0
#
```

If IntegralHPI is enabled, this command also shows information about it.

```
# clia version

Pigeon Point Shelf Manager Command Line Interpreter

Pigeon Point Shelf Manager ver. 3.2.0
Pigeon Point and the stylized lighthouse logo are trademarks of Pigeon
Point Systems.
Copyright (c) 2002-2013 Pigeon Point Systems
All rights reserved
Build date/time: May 17 2013 18:24:33
Carrier: HPDL/ACB
Carrier subtype: 0; subversion: 0
Cooling management: Default
Carrier Product ID: ACB-IV
Chassis Product ID: Standard 14-slot
IntegralHPI: HPI spec version: B.03.02; Pigeon Point Shelf Manager
3.2.0
```

4 Web Interface

The Web interface can be used to communicate with the intelligent management controllers of the shelf, with boards, and with the Shelf Manager itself remotely over the network, using a Web browser. The Web interface is based on the Command Line Interface (CLI) and is essentially a front-end to the CLI.

In redundant configurations, the external IP address is always maintained by the active Shelf Manager and is switched over to the backup Shelf Manager when the general switchover takes place. Therefore, the client always communicates to the active Shelf Manager via the Web interface in redundant configurations.

4.1 Starting the Web Interface

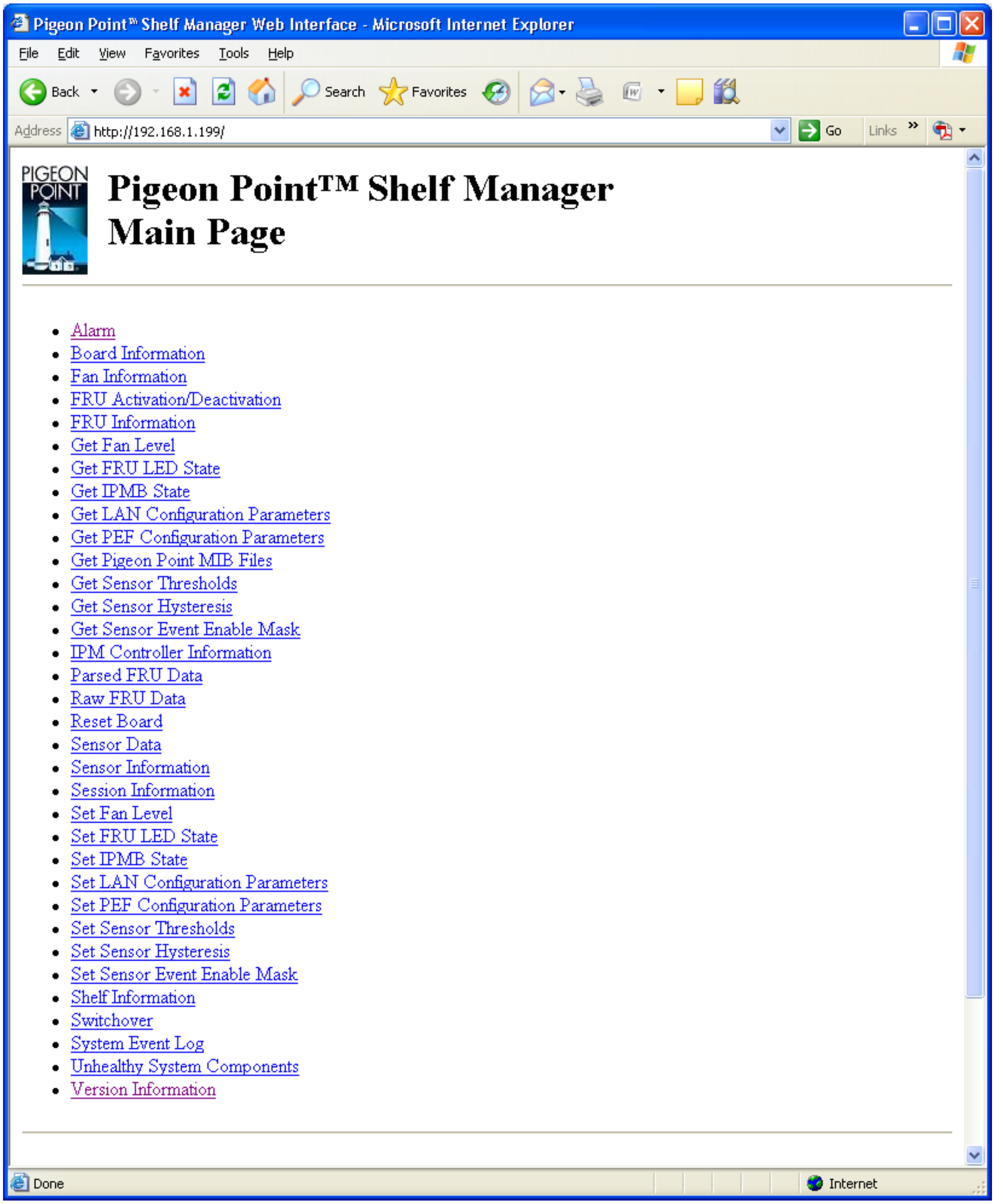
Before using the Web interface, the following prerequisites should be satisfied on the ShMM:

- one of the Ethernet interfaces should be configured and up
- the Web server **`boa`** should be running
- the Shelf Manager software (**`shelfman`**) should be running.

To use the Web interface, start any Web browser (Internet Explorer, Netscape or something else) and point it to URL **`http://<Shelf-Manager-IP-Address>`**. In the case of redundant Shelf Manager instances for a single shelf, the IP address should be the one exported outside the shelf and used for RMCP access to the Shelf Manager (instances). For example, if the Shelf Manager IP address is 192.168.1.204, the URL will look like **`http://192.168.1.204`**. The main page shows up in the browser and provides a menu of choices.

To fill a field of a Web form with a parameter value that includes the space symbol the user should enclose the value in backslashed quotes. For example, sensor “Local Temp” should be entered as **`\"Local Temp\"`** in the field “Sensor Name or LUN:Sensor #:” on the page “Set Sensor Hysteresis”.

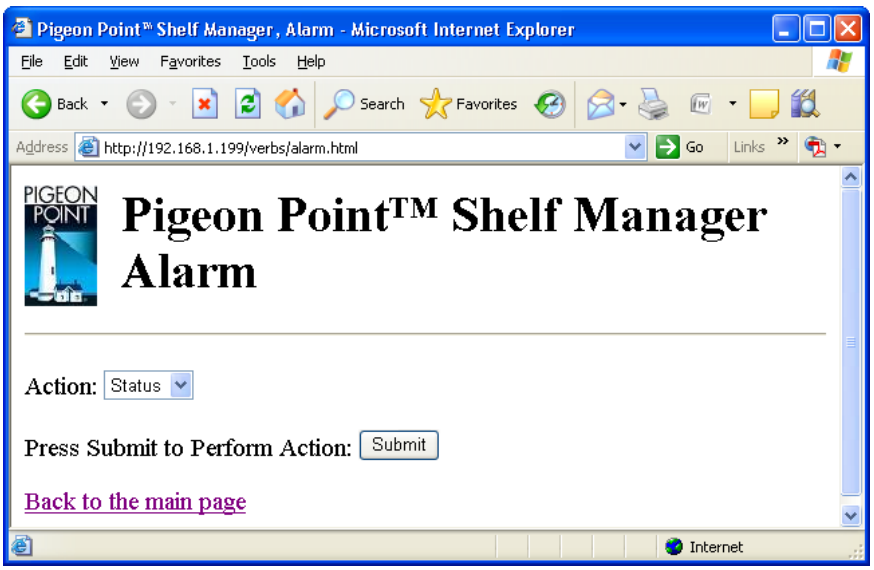
The main page contains a list of links to other pages, each of which corresponds to one of the commands available through the Web interface. These commands and the corresponding pages are described in detail in subsequent sections. The documentation relating to the command line interface can also be very helpful as the web interface provides the same functionality via a Web browser.


Pigeon Point™ Shelf Manager Web Interface - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://192.168.1.199/
Go Links
Pigeon Point™ Shelf Manager
Main Page
Alarm
Board Information
Fan Information
FRU Activation/Deactivation
FRU Information
Get Fan Level
Get FRU LED State
Get IPMB State
Get LAN Configuration Parameters
Get PEF Configuration Parameters
Get Pigeon Point MIB Files
Get Sensor Thresholds
Get Sensor Hysteresis
Get Sensor Event Enable Mask
IPM Controller Information
Parsed FRU Data
Raw FRU Data
Reset Board
Sensor Data
Sensor Information
Session Information
Set Fan Level
Set FRU LED State
Set IPMB State
Set LAN Configuration Parameters
Set PEF Configuration Parameters
Set Sensor Thresholds
Set Sensor Hysteresis
Set Sensor Event Enable Mask
Shelf Information
Switchover
System Event Log
Unhealthy System Components
Version Information
Done
Internet

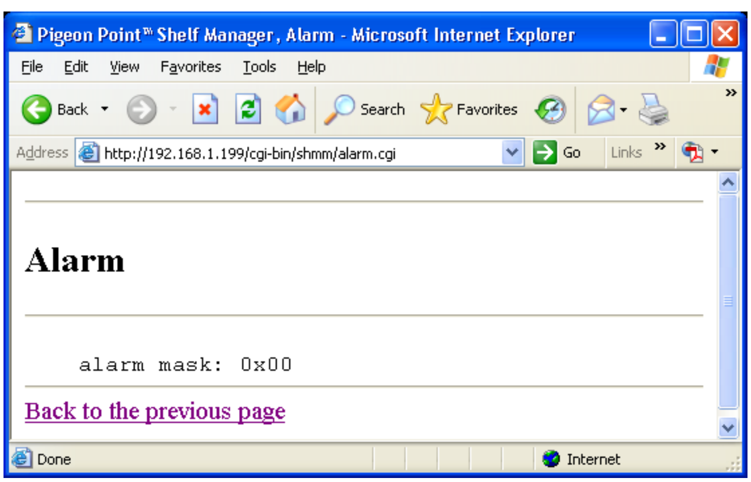
4.2 Alarm

The page "Alarm" allows the user to access to the TELCO alarm outputs. One of the following actions is specified:

- Status
- Major
- Minor
- Critical
- Clear
- Info



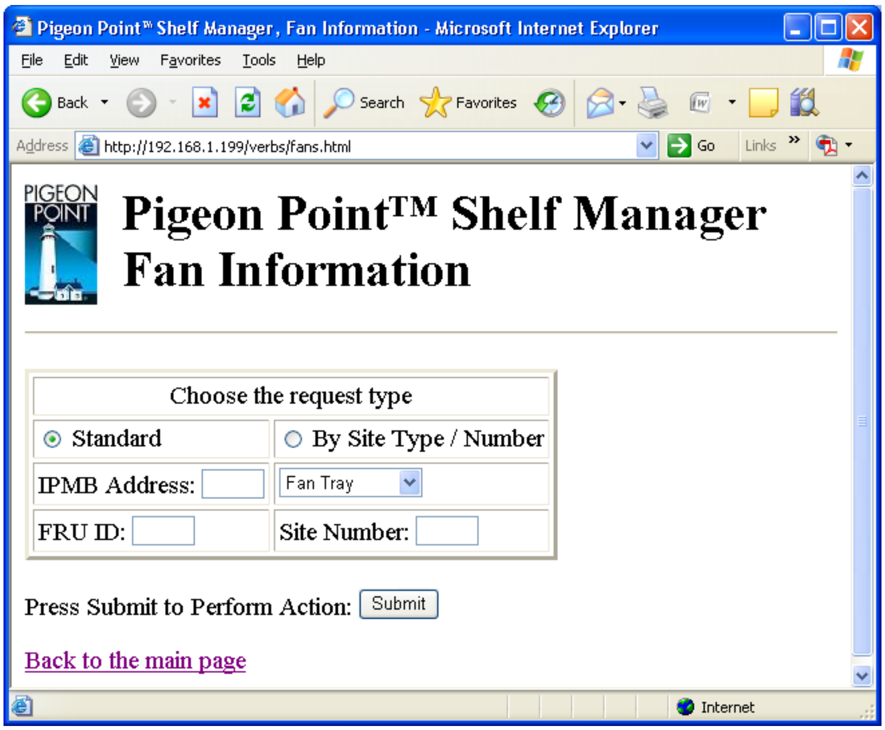
After the user selects one of the actions specified and clicks the "Submit" button, the request is executed and the results page is shown, similar to the one below. The output is essentially equal to the output produced by the CLI command **`alarm`** with a corresponding parameter.



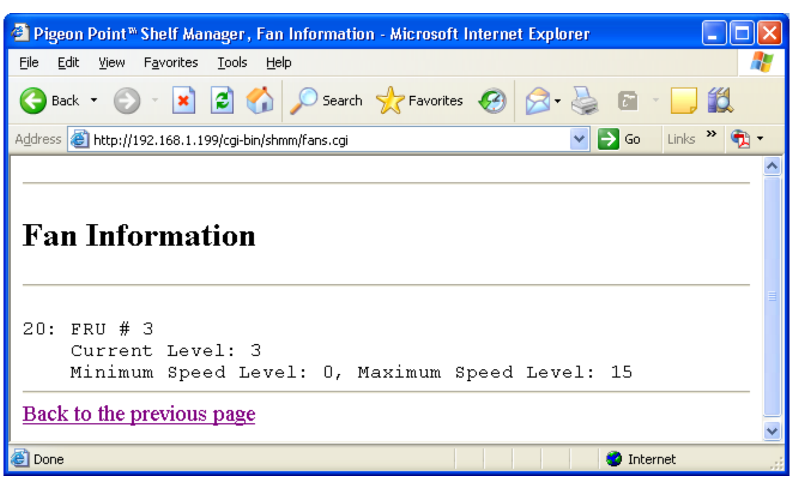
4.3 Fan Information

The page "Fan Information" allows the user to specify the IPM controller address and the FRU device ID for a fan information request. Some of the fields may be left blank; in that case:

- if all of the fields are left blank, information about all known fans in the shelf is provided.
- if only the IPM controller address is specified, information about all fans controlled by the specified IPM controller is provided.



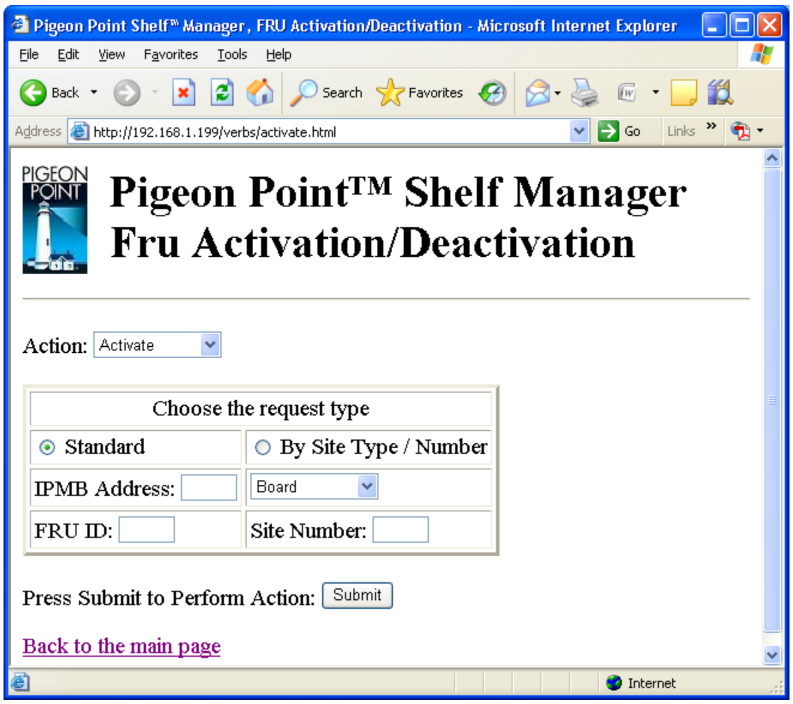
After the user fills in desired fields and clicks the "Submit" button, the request is executed and the results page is shown, similar to the one below. The output is essentially equal to the output produced by the CLI command **`fans`**.



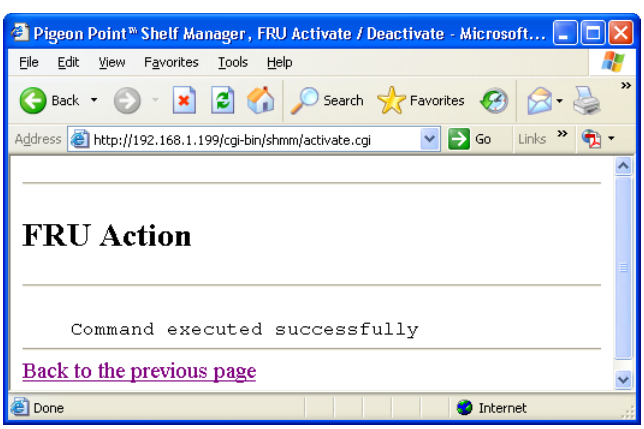
4.4 FRU Activation/Deactivation

The page "FRU Activation/Deactivation" allows the user to request activation/deactivation actions for the specified FRU. The IPM controller address and the FRU device ID identify the FRU. Both fields must be filled in. Additionally, one of the following actions is specified:

- Activate FRU
- Deactivate FRU
- Set Locked Bit
- Clear Locked Bit.



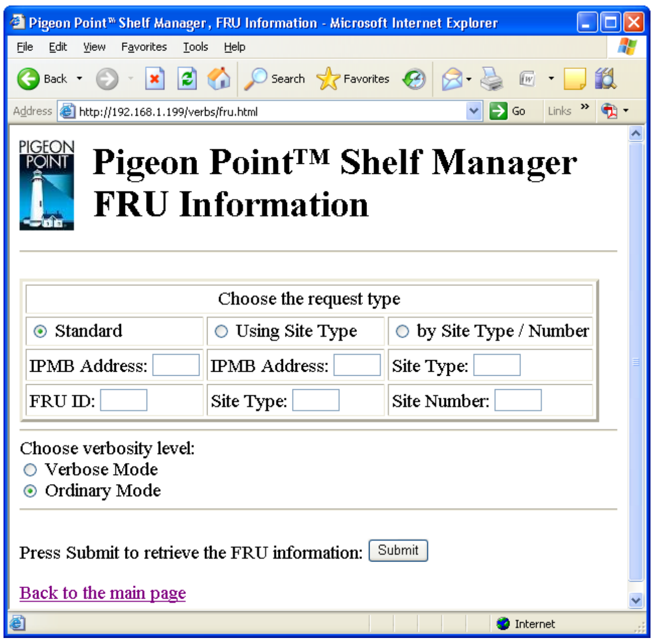
After the user fills in all fields and clicks the "Submit" button, the request is executed and the results page is produced, similar to the one below. This command is essentially equal to one of the CLI commands **`activate`**, **`deactivate`** or **`setlocked`**, depending on the action chosen.



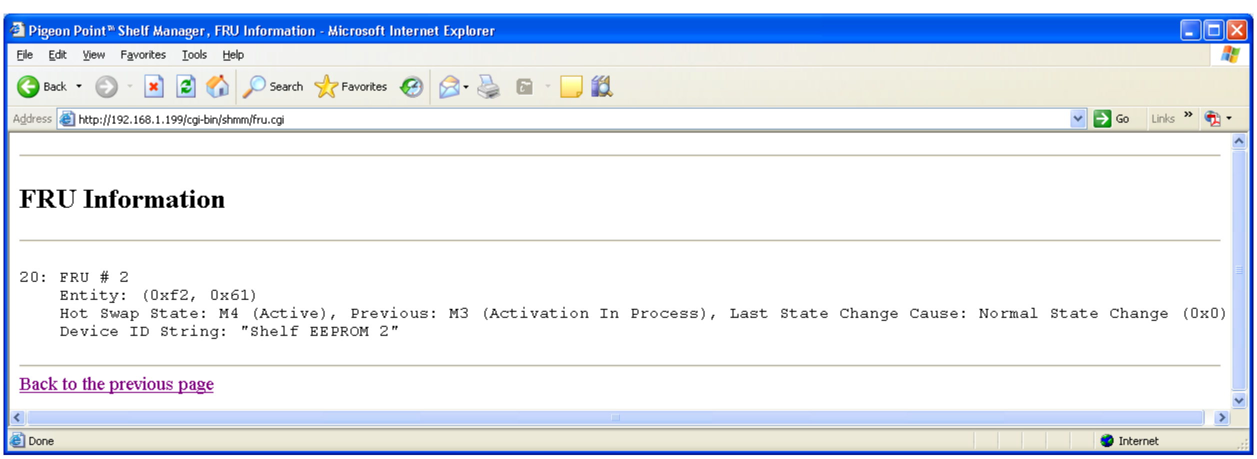
4.5 FRU Information

The page “FRU Information” allows the user to specify the IPM controller address, FRU device ID or site type, and verbosity mode for the FRU information request. Some of the fields may be left blank; in that case:

- if all of the fields are left blank, information about all known FRUs is provided
- if only the IPM controller address is specified, information about all FRUs of the specified IPM controller is provided.
- if only the site type is specified, information about all FRUs with the specified site type is provided.

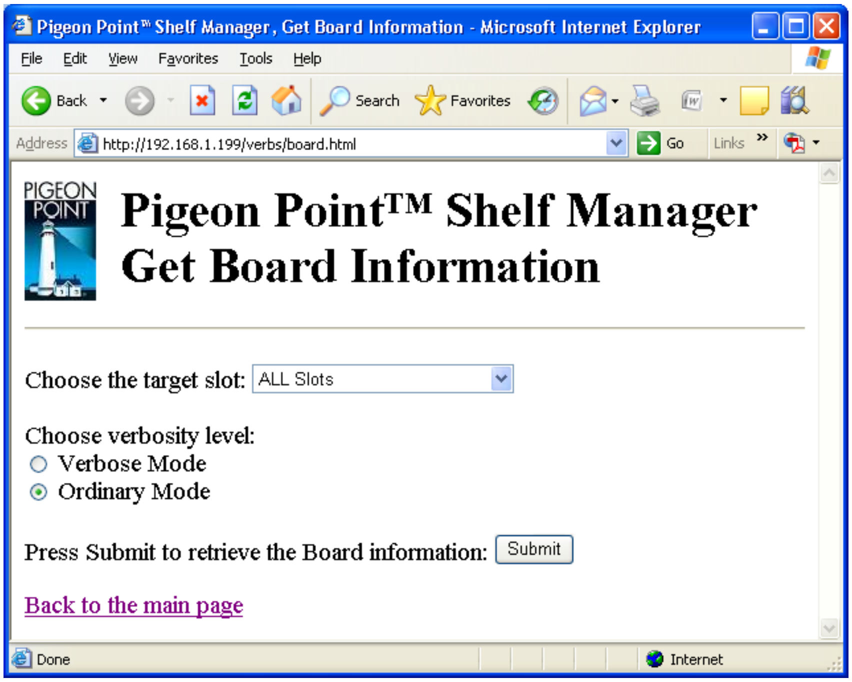


After the user fills in the desired fields and clicks the “Submit” button, the request is executed and the results page is shown, similar to the one below. The output is essentially equal to the output produced by the CLI command **`fru`**.

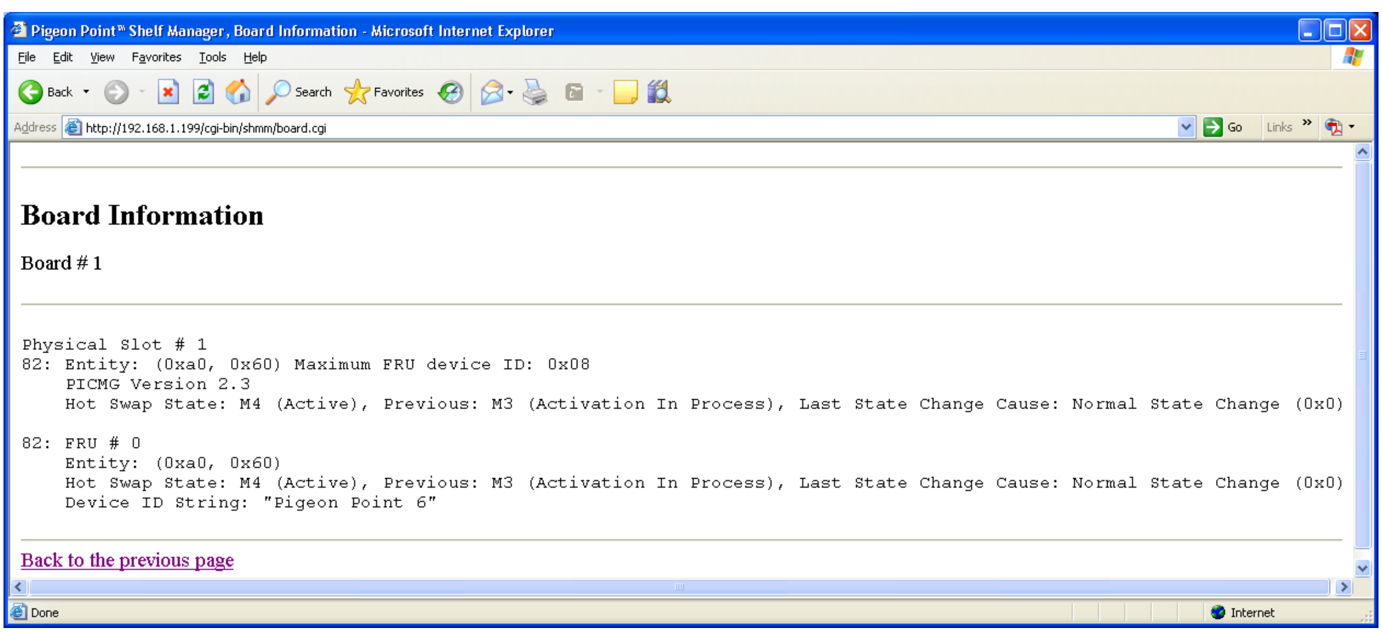


4.6 Get Board Information

The page “Get Board Information” allows the user to specify physical slot number and verbosity mode for the board information request. One of the options for the physical slot number is “all slots”.

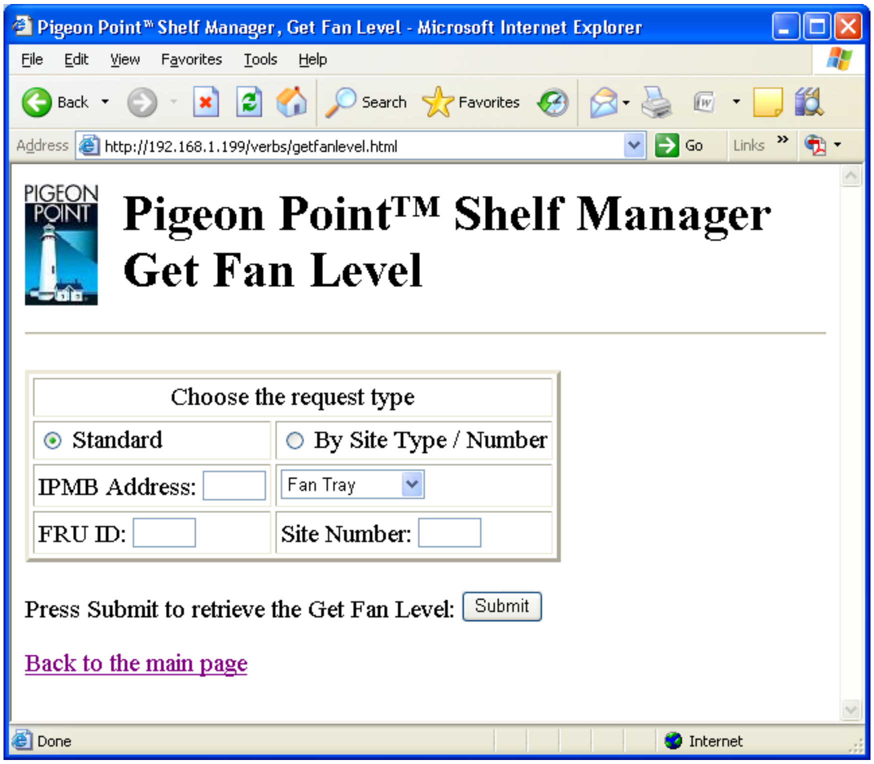


After the user fills in all fields and clicks the “Submit” button, the request is executed and the results page is shown, similar to the one below. The output is essentially equal to the output produced by the CLI command **board**.

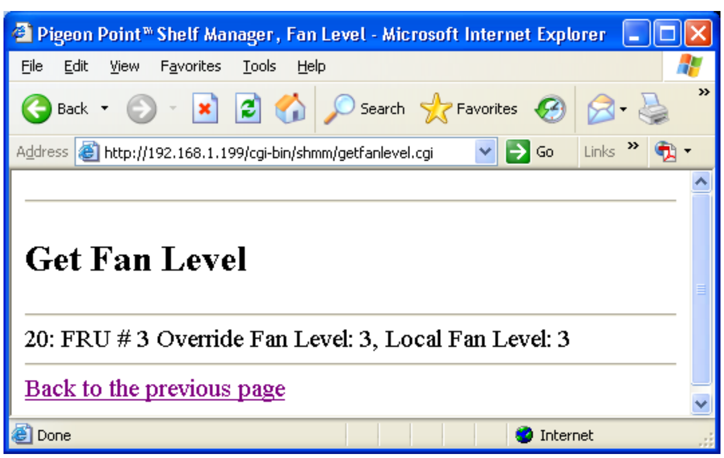


4.7 Get Fan Level

The page "Get Fan Level" allows the user to specify the IPM controller address and the FRU ID to retrieve the fan level of the specified fan.



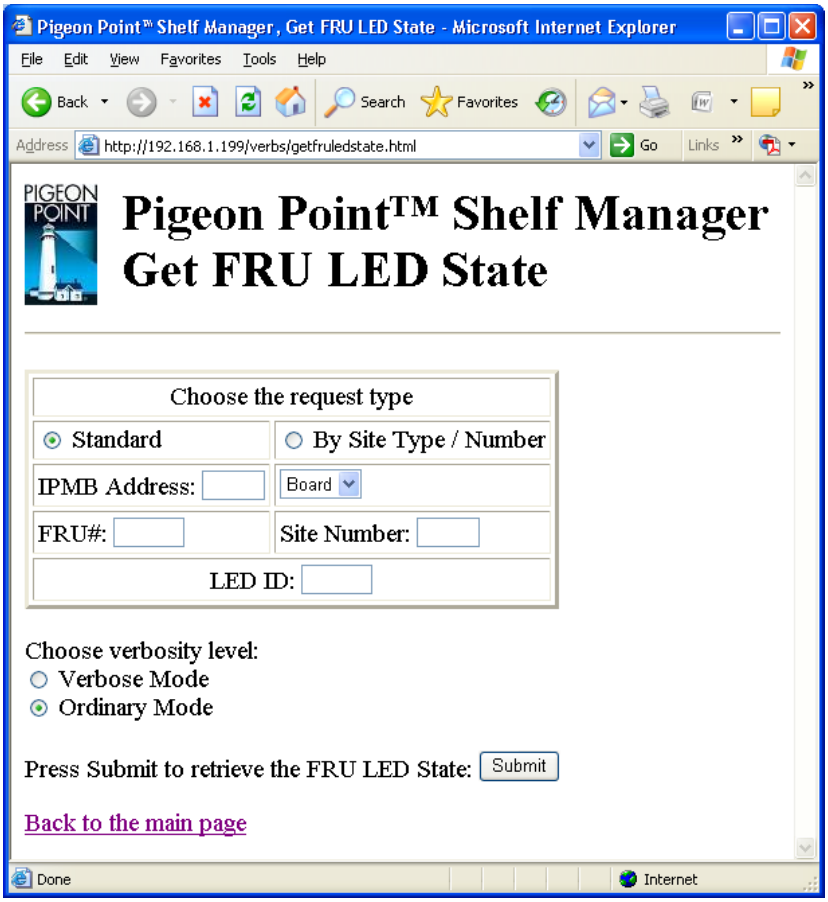
After the user fills in the desired fields and clicks the "Submit" button, the request is executed and the results page is shown, similar to the one below. The output is essentially equal to the output produced by the CLI command **`getfanlevel`**.



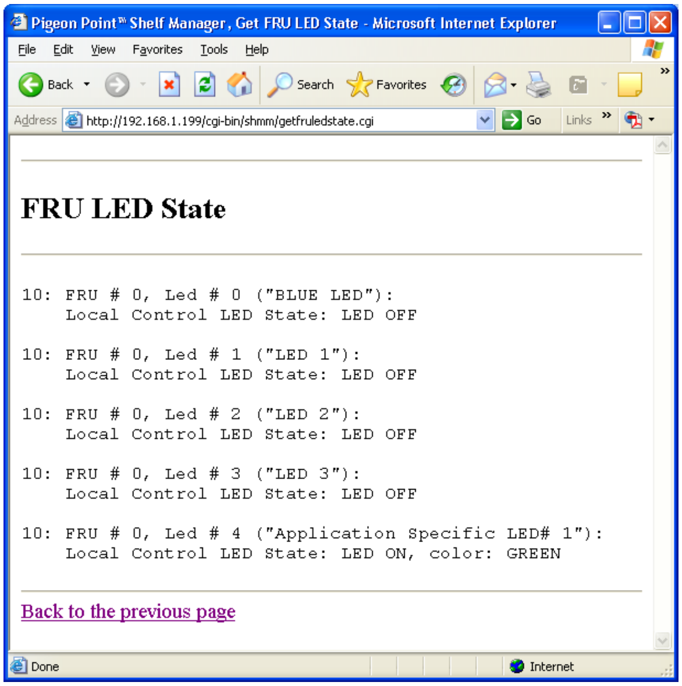
4.8 Get FRU LED State

The page "Get FRU LED State" allows the user to obtain the current FRU LED state on all levels of control that are enabled for the LED(s). In verbose mode, information about the colors supported by the LED(s) is also shown. Information can be shown about a specific LED or all LEDs for the given FRU if the correspondent fields are filled. Some of the fields may be left blank.

In that case, 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.

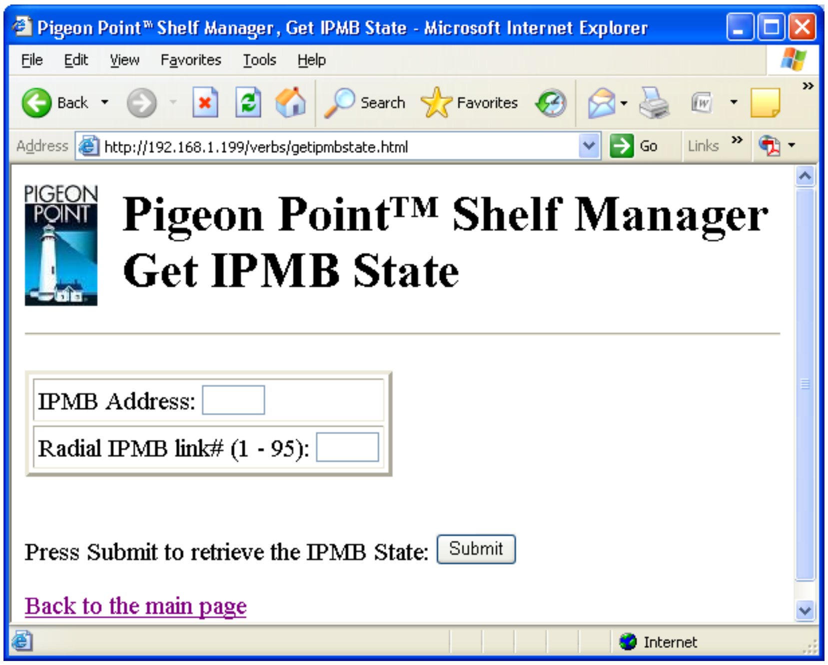


After the user fills in the request information and clicks the "Submit" button, the request is executed and the results page is generated, similar to the one below. The output produced by this command is essentially the same as the output produced by the CLI command **`getfruledstate`**.

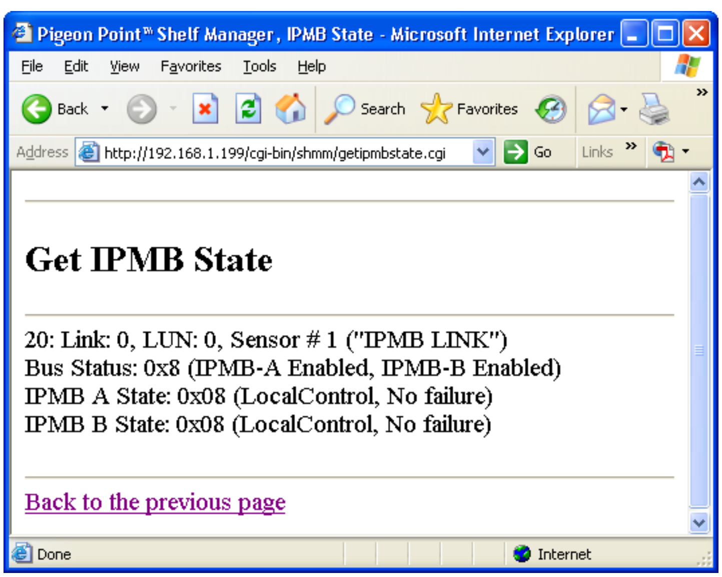

Pigeon Point™ Shelf Manager, Get FRU LED State - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://192.168.1.199/cgi-bin/shmm/getfruledstate.cgi
Go Links
FRU LED State
10: FRU # 0, Led # 0 ("BLUE LED"):
Local Control LED State: LED OFF
10: FRU # 0, Led # 1 ("LED 1"):
Local Control LED State: LED OFF
10: FRU # 0, Led # 2 ("LED 2"):
Local Control LED State: LED OFF
10: FRU # 0, Led # 3 ("LED 3"):
Local Control LED State: LED OFF
10: FRU # 0, Led # 4 ("Application Specific LED# 1"):
Local Control LED State: LED ON, color: GREEN
Back to the previous page
Done
Internet

4.9 Get IPMB State

The page "Get IPMB State" allows the user to obtain the current state of IPMB-0 on the target IPM controller. The IPMB Address must be specified. In a bused environment, or in a radial environment if the target IPM controller is not an IPMB hub, the field "Radial IPMB link# (1-95)" must be left empty. Information about the state of IPMB-A and IPMB-B on the target IPM controller is reported.

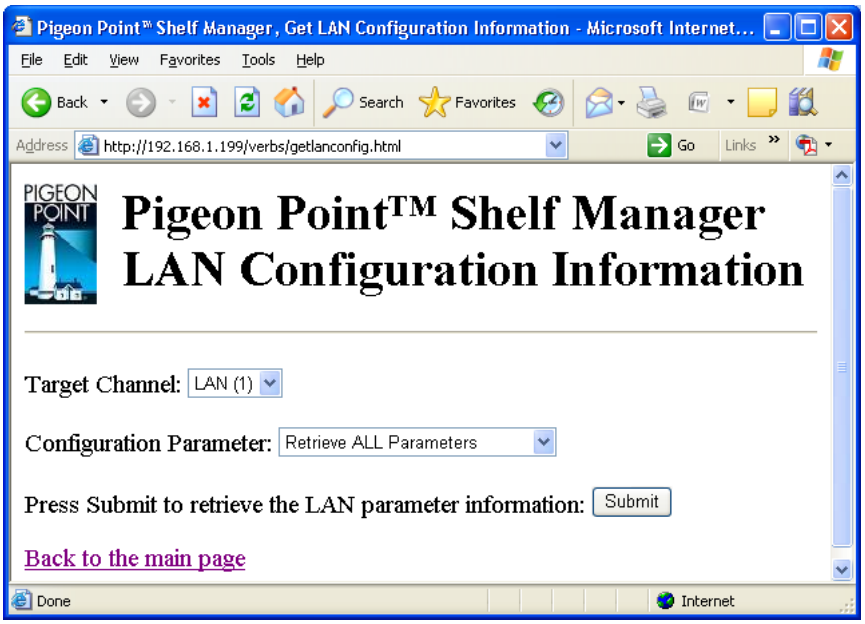


After the user fills in the request information and clicks the "Submit" button, the request is executed and the results page is generated, similar to the one below. The output produced by this command is essentially the same as the output produced by the CLI command **`getipmbstate`**.

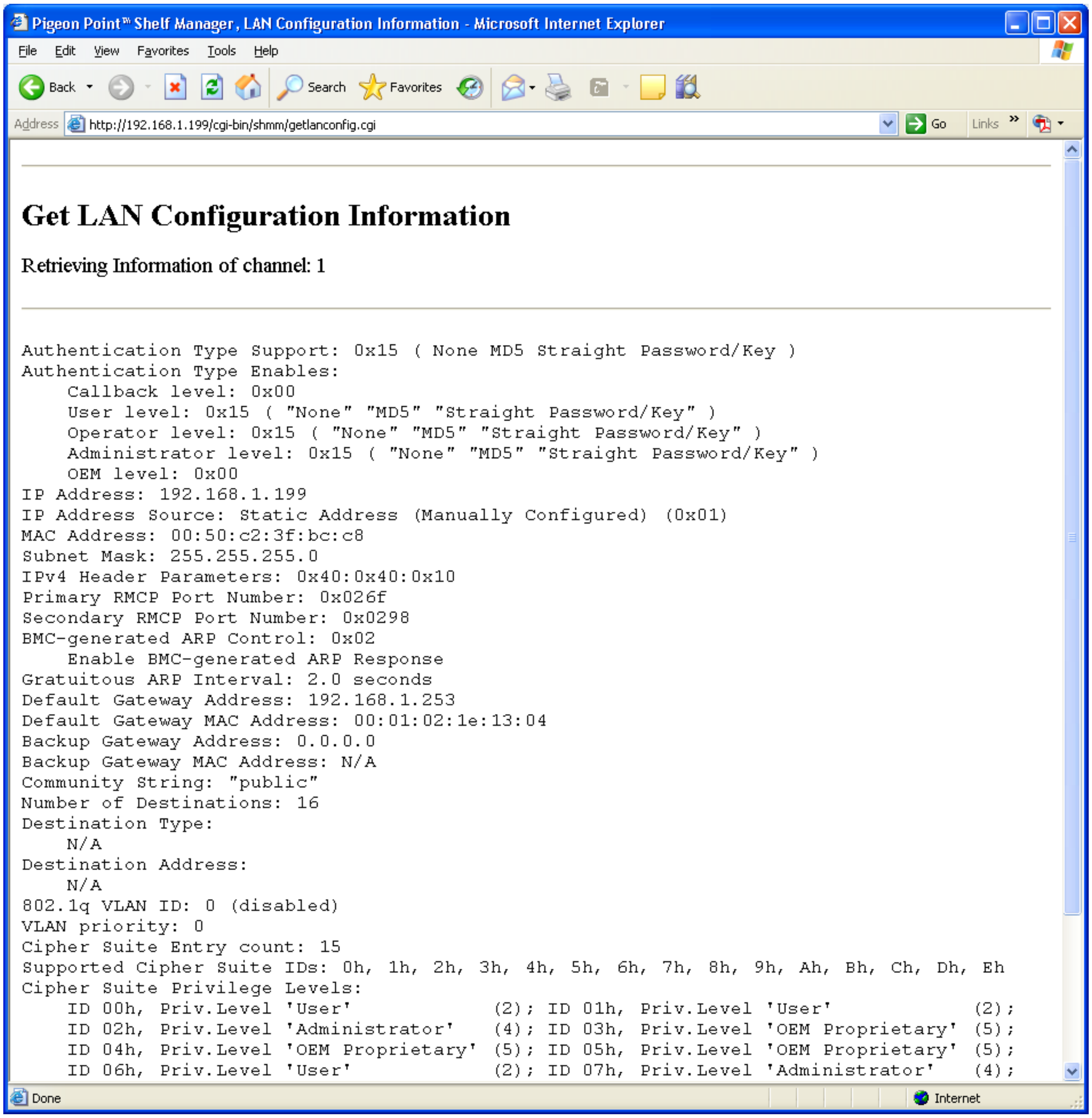


4.10 Get LAN Configuration Information

The page "Get LAN Configuration Information" allows the user to retrieve LAN configuration information for the specified channel. The user can request the value of one of the defined parameters or of all LAN parameters.



After the user fills in the request information and clicks the "Submit" button, the request is executed and the results page is generated, similar to the one below. The output produced by this command is essentially the same as the output produced by the CLI command **`getlanconfig`**.



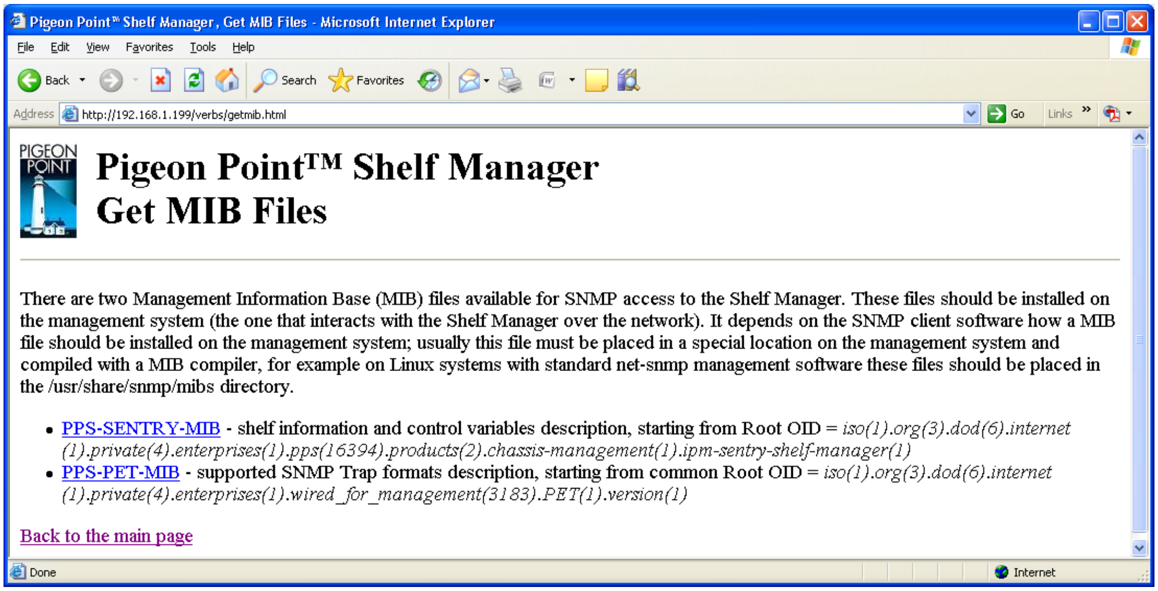
Get LAN Configuration Information

Retrieving Information of channel: 1

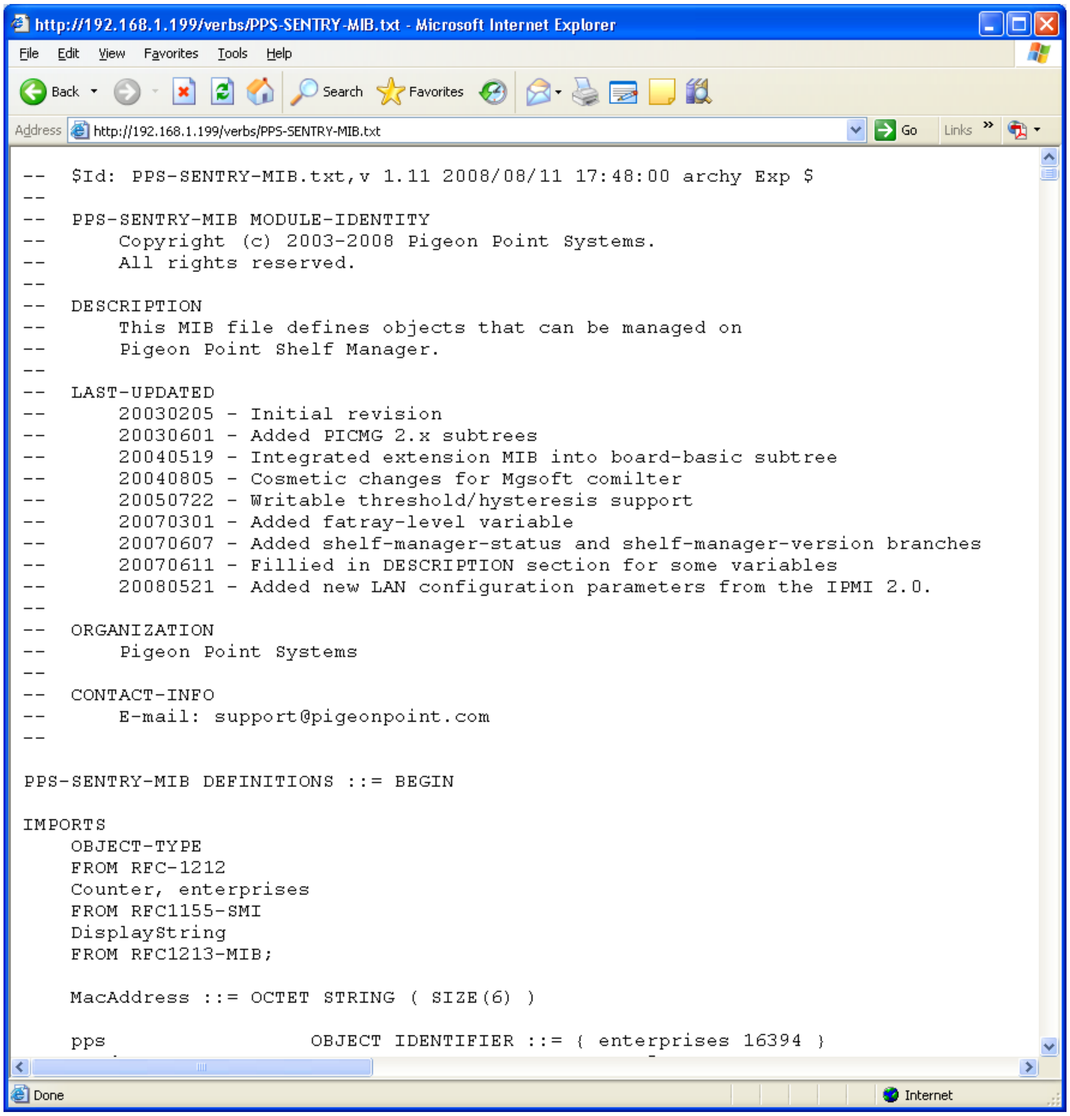
```
Authentication Type Support: 0x15 ( None MD5 Straight Password/Key )
Authentication Type Enables:
    Callback level: 0x00
    User level: 0x15 ( "None" "MD5" "Straight Password/Key" )
    Operator level: 0x15 ( "None" "MD5" "Straight Password/Key" )
    Administrator level: 0x15 ( "None" "MD5" "Straight Password/Key" )
    OEM level: 0x00
IP Address: 192.168.1.199
IP Address Source: Static Address (Manually Configured) (0x01)
MAC Address: 00:50:c2:3f:bc:c8
Subnet Mask: 255.255.255.0
IPv4 Header Parameters: 0x40:0x40:0x10
Primary RMCP Port Number: 0x026f
Secondary RMCP Port Number: 0x0298
BMC-generated ARP Control: 0x02
    Enable BMC-generated ARP Response
Gratuitous ARP Interval: 2.0 seconds
Default Gateway Address: 192.168.1.253
Default Gateway MAC Address: 00:01:02:1e:13:04
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
802.1q VLAN ID: 0 (disabled)
VLAN priority: 0
Cipher Suite Entry count: 15
Supported Cipher Suite IDs: 0h, 1h, 2h, 3h, 4h, 5h, 6h, 7h, 8h, 9h, Ah, Bh, Ch, Dh, Eh
Cipher Suite Privilege Levels:
    ID 00h, Priv.Level 'User'            (2); ID 01h, Priv.Level 'User'            (2);
    ID 02h, Priv.Level 'Administrator'   (4); ID 03h, Priv.Level 'OEM Proprietary' (5);
    ID 04h, Priv.Level 'OEM Proprietary' (5); ID 05h, Priv.Level 'OEM Proprietary' (5);
    ID 06h, Priv.Level 'User'            (2); ID 07h, Priv.Level 'Administrator'   (4);
```

4.11 Get Pigeon Point MIB Files

The “Get Pigeon Point MIB Files” page allows the user to obtain the contents of the Pigeon Point SNMP MIB files.



When the user clicks on a MIB file name, a page appears with the contents of the corresponding MIB file, similar to the one below.

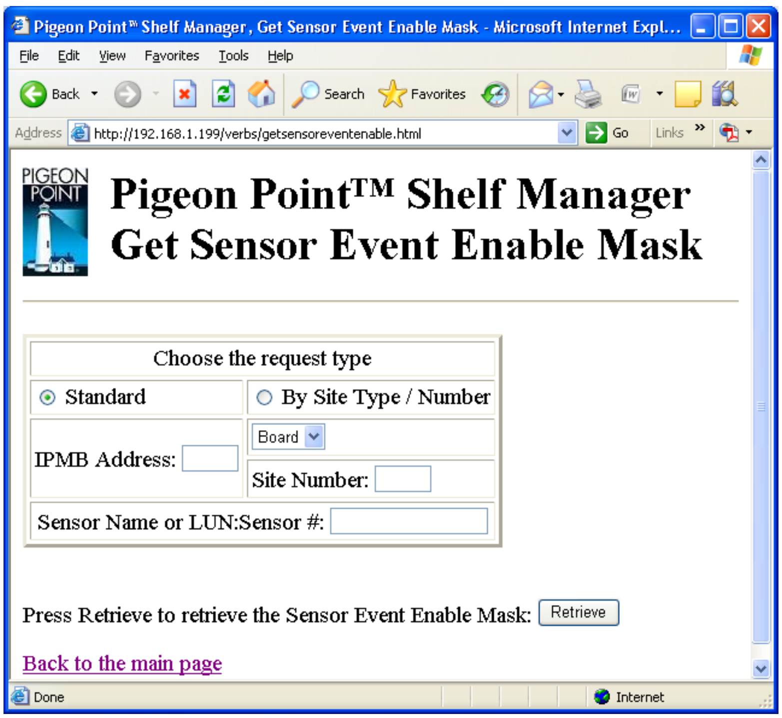



```
--  $Id: PPS-SENTRY-MIB.txt,v 1.11 2008/08/11 17:48:00 archy Exp $
--
--  PPS-SENTRY-MIB MODULE-IDENTITY
--      Copyright (c) 2003-2008 Pigeon Point Systems.
--      All rights reserved.
--
--  DESCRIPTION
--      This MIB file defines objects that can be managed on
--      Pigeon Point Shelf Manager.
--
--  LAST-UPDATED
--      20030205 - Initial revision
--      20030601 - Added PICMG 2.x subtrees
--      20040519 - Integrated extension MIB into board-basic subtree
--      20040805 - Cosmetic changes for Mgsoft comilter
--      20050722 - Writable threshold/hysteresis support
--      20070301 - Added fatray-level variable
--      20070607 - Added shelf-manager-status and shelf-manager-version branches
--      20070611 - Fillied in DESCRIPTION section for some variables
--      20080521 - Added new LAN configuration parameters from the IPMI 2.0.
--
--  ORGANIZATION
--      Pigeon Point Systems
--
--  CONTACT-INFO
--      E-mail: support@pigeonpoint.com
--

PPS-SENTRY-MIB DEFINITIONS ::= BEGIN

IMPORTS
    OBJECT-TYPE
    FROM RFC-1212
    Counter, enterprises
    FROM RFC1155-SMI
    DisplayString
    FROM RFC1213-MIB;

    MacAddress ::= OCTET STRING ( SIZE(6) )

    pps                 OBJECT IDENTIFIER ::= { enterprises 16394 }
```

4.12 Get Sensor Event Enable Mask

The page "Set Sensor Event Enable Mask" allows to the user to obtain the current event enable mask values of the specified sensor(s).

The user may 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. 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.



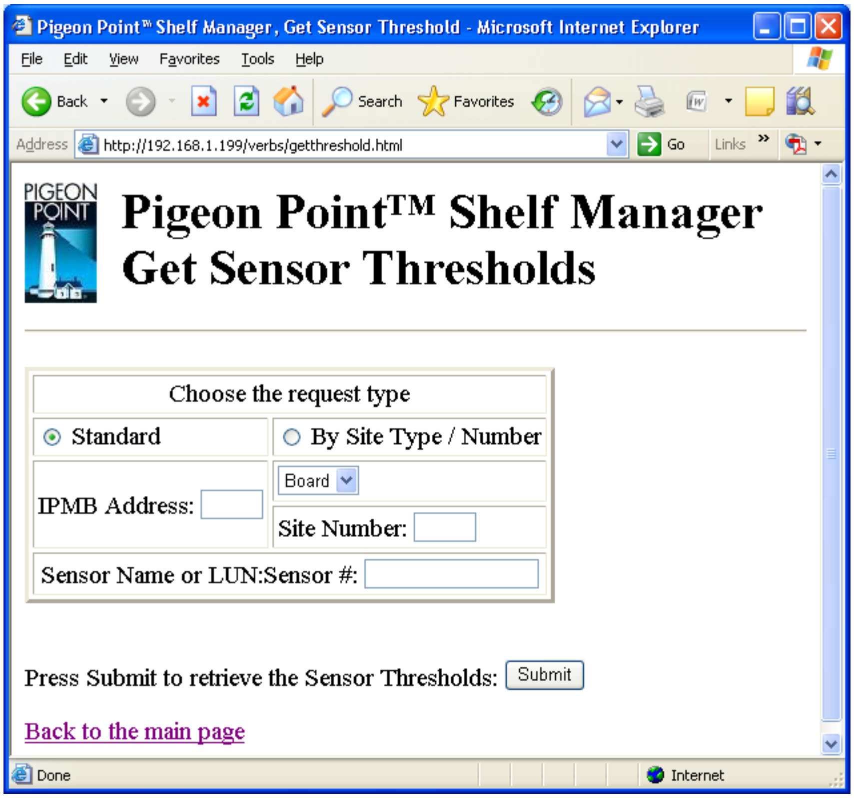
After the user fills in the request information and clicks the "Submit" button, the request is executed and the results page is generated, similar to the one below. The output produced by this command is essentially the same as the output produced by the CLI command **`getsensoreventenable`**.


Pigeon Point™ Shelf Manager, Get Sensor Event Enable Mask - Microsoft Int...
File Edit View Favorites Tools Help
Back Search Favorites
Address http://192.168.1.199/cgi-bin/shmm/getsensoreventenable.cgi Go Links
Get Sensor Event Enable Mask
20: LUN: 0, Sensor # 2 ("FRU 1 HOT_SWAP")
Type: Discrete (0x6f), "Hot Swap" (0xf0)
Assertion event mask: 0x00ff
Deassertion event mask: 0x0000
Back to the previous page
Done
Internet

4.13 Get Sensor Thresholds

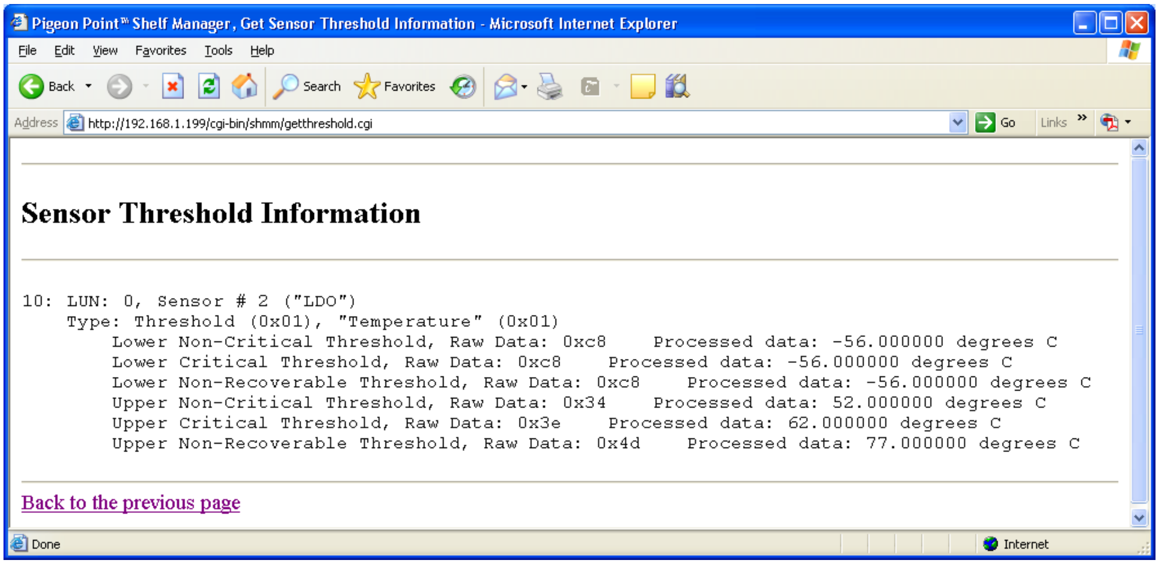
The page "Get Sensor Thresholds" allows the user to specify the IPM controller address and the sensor number or name for a threshold information request. Some of the fields may be left blank; in that case:

- if all of the fields are left blank, threshold information for all known sensors on all IPM controllers is provided.
- if only the IPM controller address is specified, threshold information for all sensors of the specified IPM controller is provided.



In the field "Sensor Name or LUN:Sensor #", the user can identify the target sensor by specifying the sensor name or specifying the sensor LUN and sensor number. In the last case, the LUN is optional; if specified, it is separated from the sensor number with a colon. If the user specifies only the sensor number, information is returned about known sensors with the specified sensor number on all LUNs. Valid values for the LUN are 0, 1 and 3. (LUN 2 is reserved.)

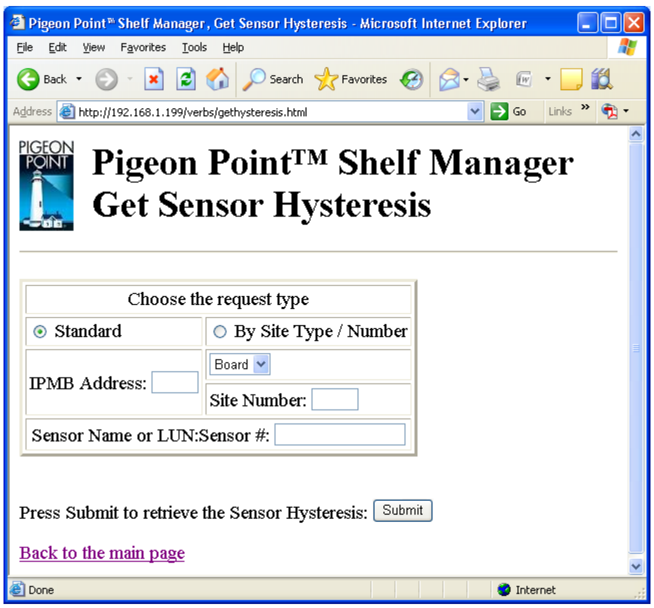
After the user fills in the desired fields and clicks the "Submit" button, the request is executed and the results page is shown, similar to the one below. The output is essentially equal to the output produced by the CLI command `getthreshold`.


Pigeon Point™ Shelf Manager, Get Sensor Threshold Information - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://192.168.1.199/cgi-bin/shmm/getthreshold.cgi
Go Links
Sensor Threshold Information
10: LUN: 0, Sensor # 2 ("LDO")
Type: Threshold (0x01), "Temperature" (0x01)
Lower Non-Critical Threshold, Raw Data: 0xc8 Processed data: -56.000000 degrees C
Lower Critical Threshold, Raw Data: 0xc8 Processed data: -56.000000 degrees C
Lower Non-Recoverable Threshold, Raw Data: 0xc8 Processed data: -56.000000 degrees C
Upper Non-Critical Threshold, Raw Data: 0x34 Processed data: 52.000000 degrees C
Upper Critical Threshold, Raw Data: 0x3e Processed data: 62.000000 degrees C
Upper Non-Recoverable Threshold, Raw Data: 0x4d Processed data: 77.000000 degrees C
Back to the previous page
Done
Internet

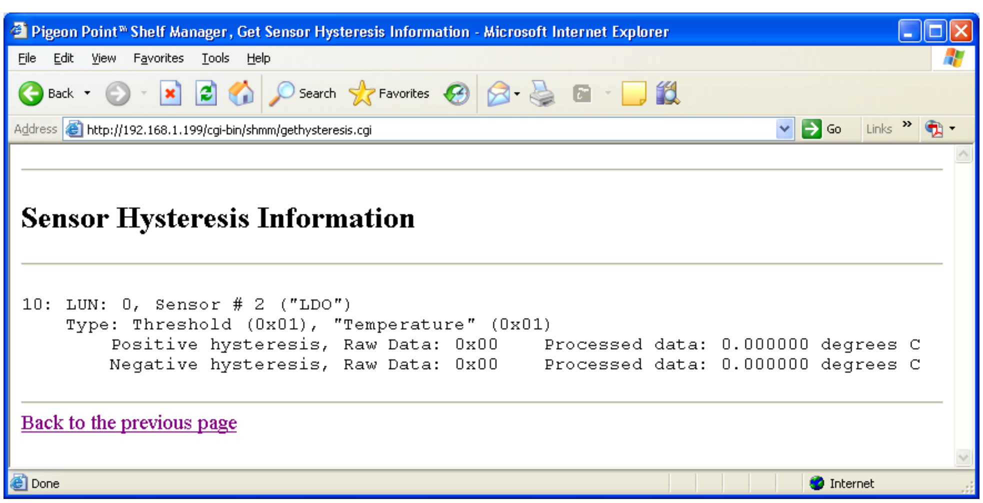
4.14 Get Sensor Hysteresis

The page "Get Sensor Hysteresis" allows the user to retrieve the positive-going and negative-going hysteresis for the specified sensor. In the field "Sensor Name or LUN:Sensor #", the user can identify the target sensor by specifying the sensor name or specifying the sensor LUN and sensor number.

In the last case, the LUN is optional; if specified, it is separated from the sensor number with a colon. If the user specifies only the sensor number, information is returned about known sensors with the specified sensor number on all LUNs. Valid values for the LUN are 0, 1 and 3. (LUN 2 is reserved.).



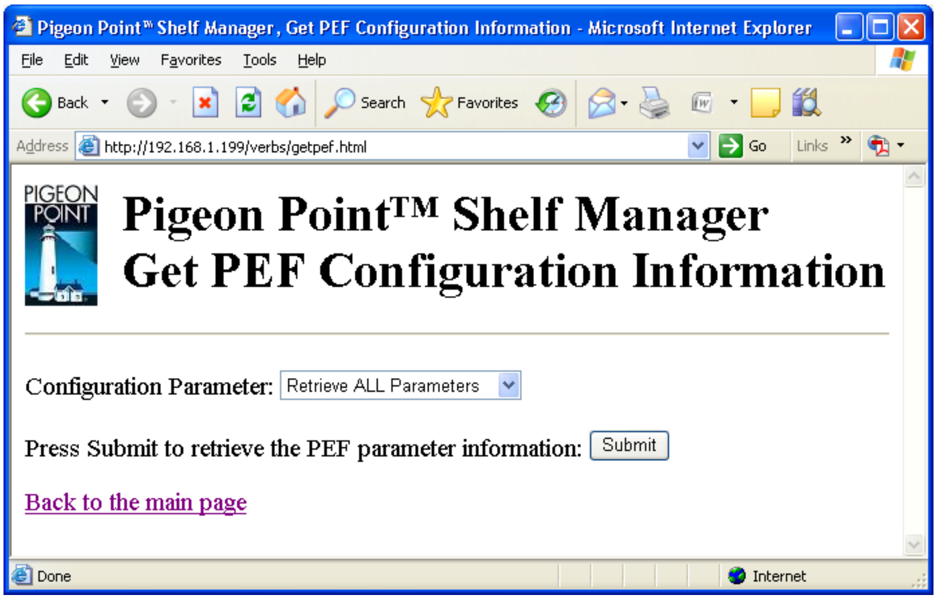
After the user fills in the desired fields and clicks the "Submit" button, the request is executed and the result page is shown, similar to the one below. The output is essentially equal to the output produced by the CLI command `gethysteresis`.



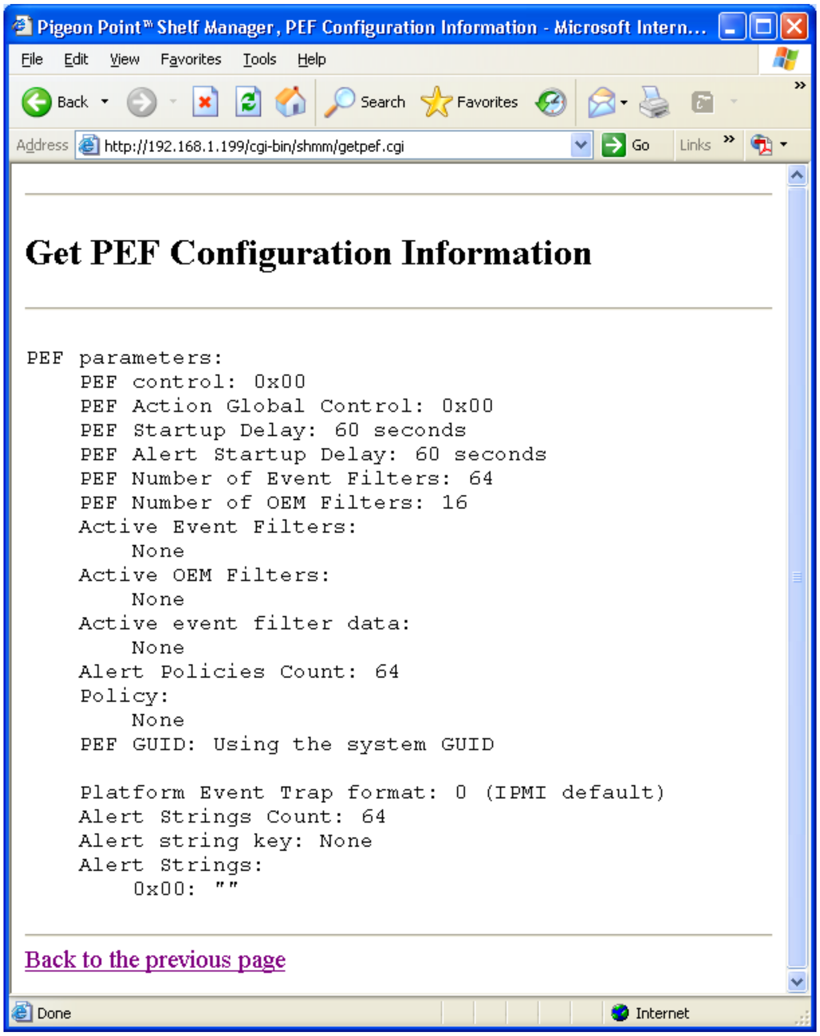
4.15 Get PEF Configuration Information

The page "Get PEF Configuration Information" allows the user to retrieve Platform Event Filter (PEF) configuration information.

The user can request the value of one of the defined parameters or of all PEF parameters.



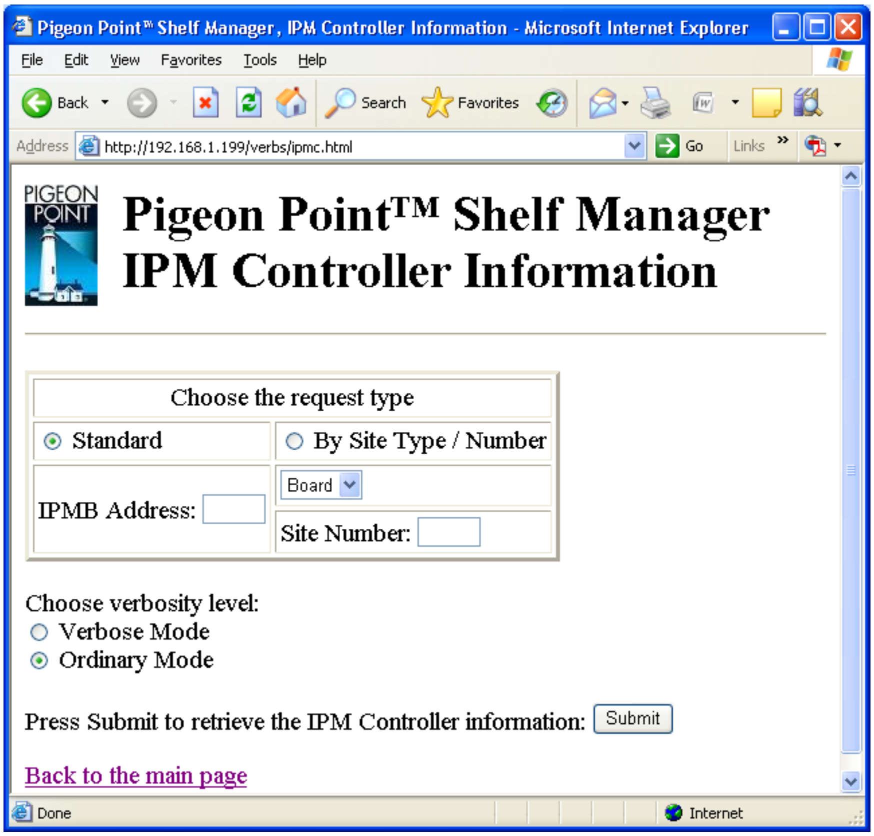
After the user fills in the request information and clicks the "Submit" button, the request is executed and the results page is generated. The results page will look similar to the one below. The output produced by this command is essentially the same as the output produced by the CLI command **`getpefconfig`**.


Pigeon Point™ Shelf Manager, PEF Configuration Information - Microsoft Intern...
File Edit View Favorites Tools Help
Back
Search
Favorites
Address http://192.168.1.199/cgi-bin/shmm/getpef.cgi
Go
Links
Get PEF Configuration Information
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 OEM Filters:
None
Active event filter data:
None
Alert Policies Count: 64
Policy:
None
PEF GUID: Using the system GUID
Platform Event Trap format: 0 (IPMI default)
Alert Strings Count: 64
Alert string key: None
Alert Strings:
0x00: ""
Back to the previous page
Done
Internet

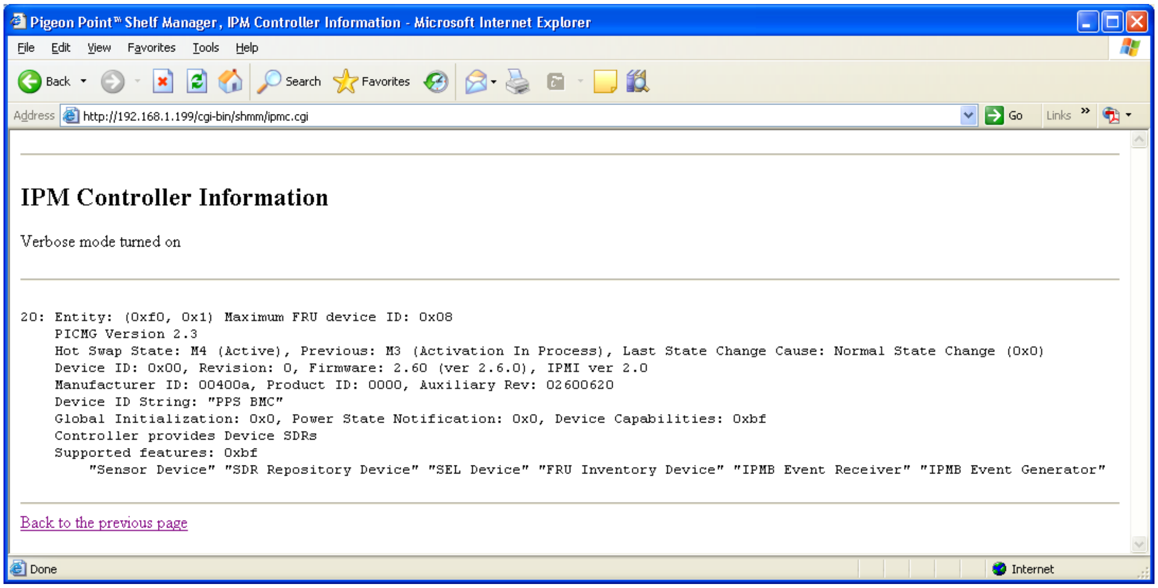
4.16 IPM Controller Information

The page "IPM Controller Information" allows the user to specify the IPM controller address and verbosity mode for the IPM controller information request.

The address field may be left blank; in that case, information about all known IPM controllers is provided.



After the user fills in the desired fields and clicks the "Submit" button, the request is executed and a results page is shown, similar to the one below. The output is essentially equal to the output produced by the CLI command **`ipmc`**.

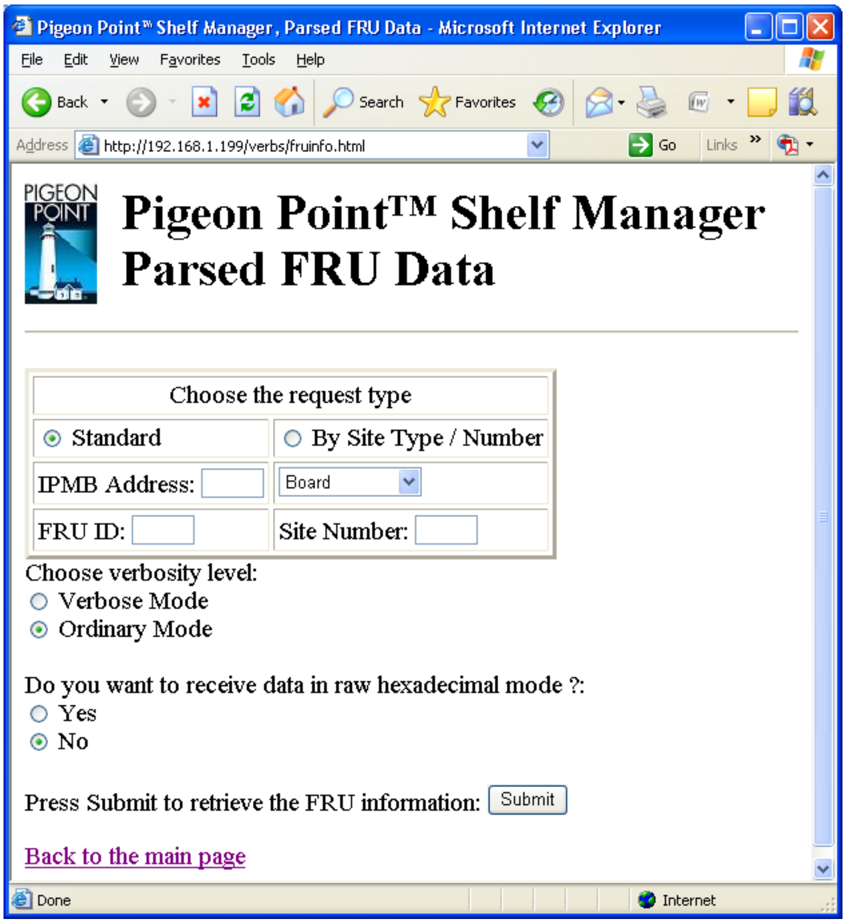


Note:

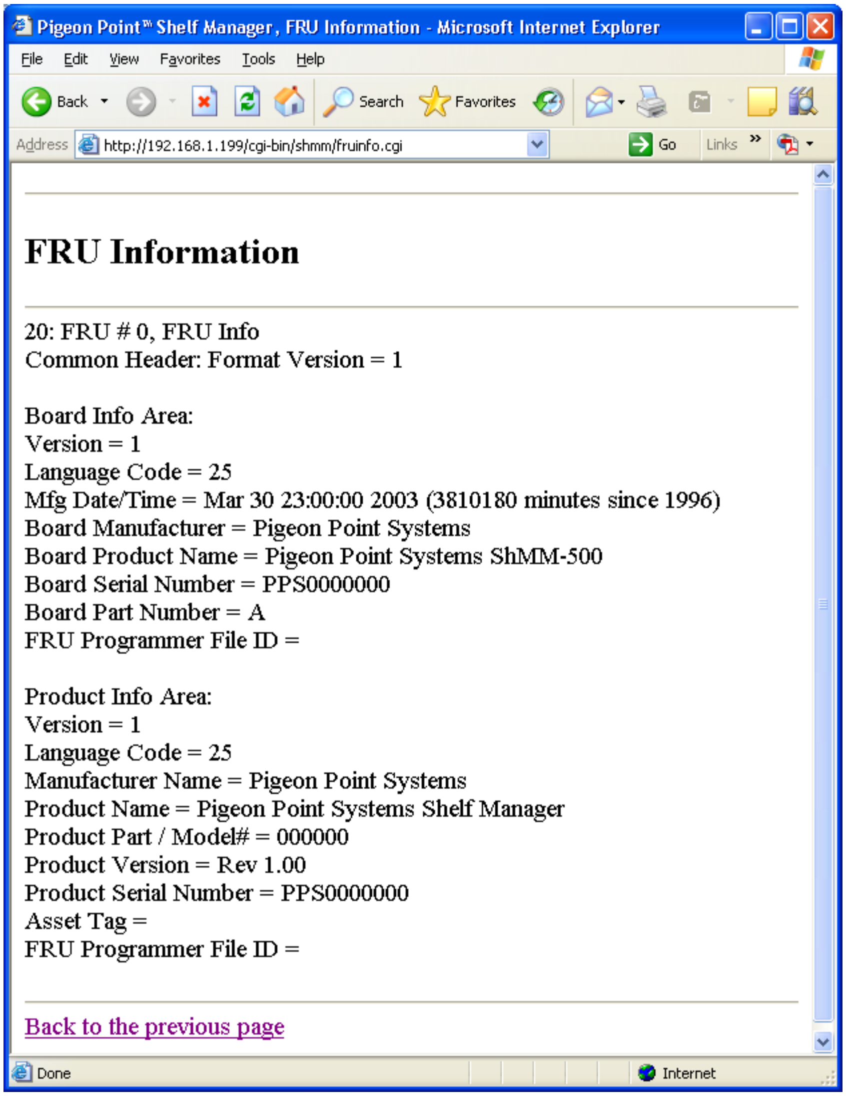
This and many subsequent pages offer an alternate style of request “By Site Type/Number”, which allows the user to specify the site type and site number as the address of the target shelf object. Currently all available site types are supported only in CompactPCI systems; in AdvancedTCA systems, only the site type “Board” is supported. The HTML user interface does not prohibit the user from choosing a site type that is unsupported on the target shelf; in that case, an error message will be returned by the service provider in the target shelf.

4.17 Parsed FRU Data

The page "Parsed FRU Data" allows the user to get a parsed version of the FRU data information. The user should specify the IPM controller address, FRU device ID or site type and the site number. Also, the user can change the verbosity level and request the data in raw format (as a hexadecimal dump).



After the user specifies all necessary information and clicks the "Submit" button, the request is executed and the results page is shown, similar to the one below. This command is essentially equal to the CLI command `fruinfo`.



FRU Information

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

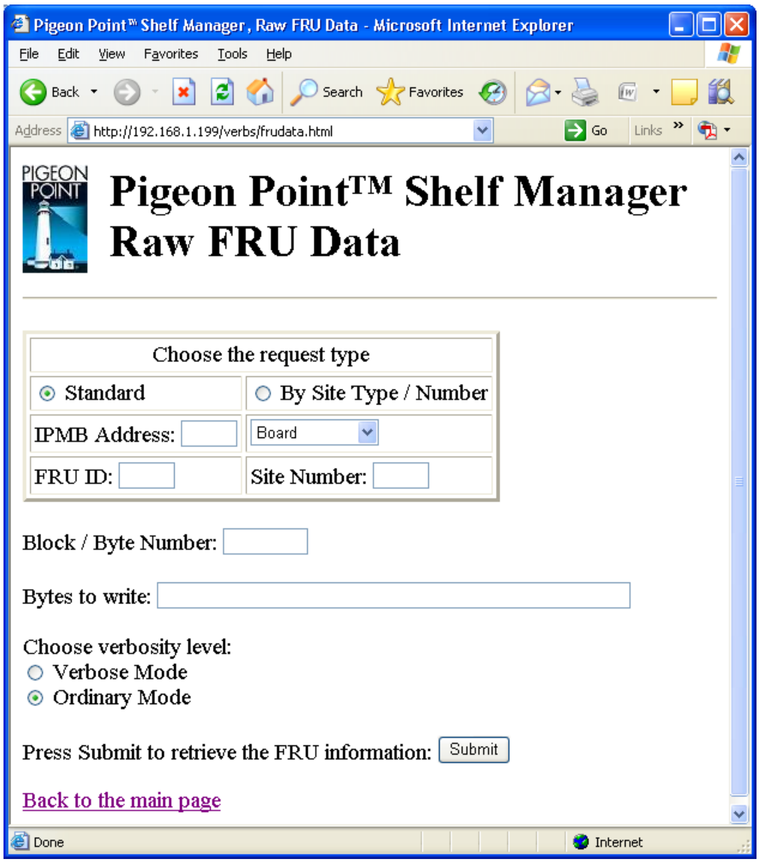
Board Info Area:
Version = 1
Language Code = 25
Mfg Date/Time = Mar 30 23:00:00 2003 (3810180 minutes since 1996)
Board Manufacturer = Pigeon Point Systems
Board Product Name = Pigeon Point Systems ShMM-500
Board Serial Number = PPS0000000
Board Part Number = A
FRU Programmer File ID =

Product Info Area:
Version = 1
Language Code = 25
Manufacturer Name = Pigeon Point Systems
Product Name = Pigeon Point Systems Shelf Manager
Product Part / Model# = 000000
Product Version = Rev 1.00
Product Serial Number = PPS0000000
Asset Tag =
FRU Programmer File ID =

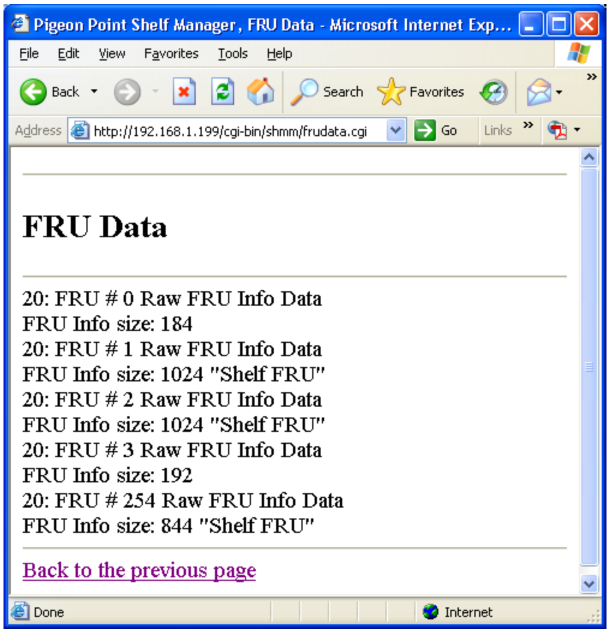
Back to the previous page

4.18 Raw FRU Data

The page "Raw FRU Data" allows the user to get the FRU data information in raw form. The user can specify the IPM controller address, FRU device ID or site type and the facility level. If none of the parameters or only the IPM controller address is specified, this page shows the FRU Inventory Ares Info on each FRU in the shelf or associated with that IPM controller.

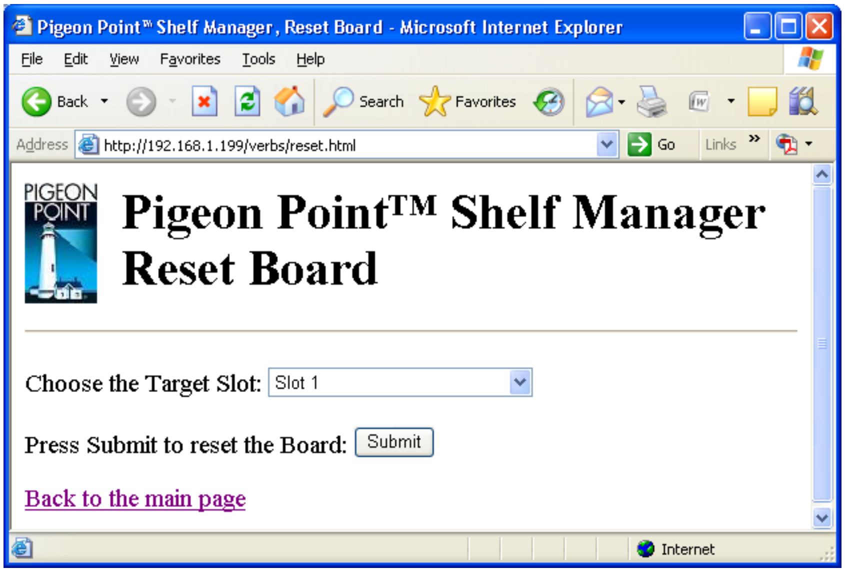


After the user specifies all necessary information and clicks the "Submit" button, the request is executed and the results page is shown, similar to the one below. This command is essentially equal to the CLI command **`frudata`**.

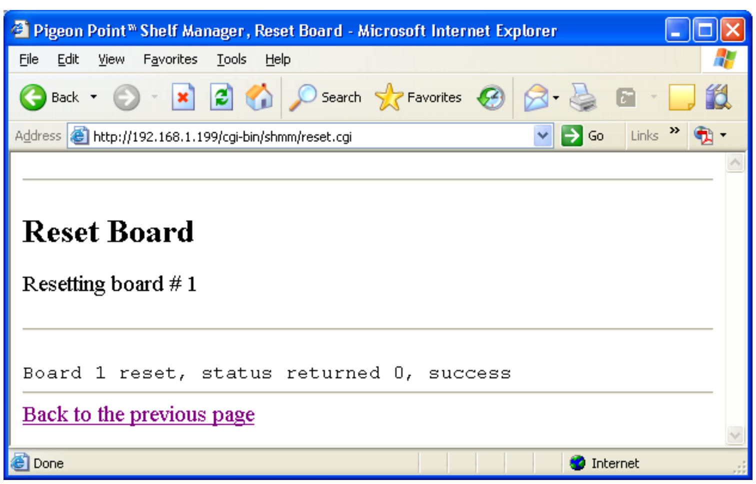

Pigeon Point Shelf Manager, FRU Data - Microsoft Internet Exp...
File Edit View Favorites Tools Help
Back Search Favorites
Address http://192.168.1.199/cgi-bin/shmm/frudata.cgi Go Links
FRU Data
20: FRU # 0 Raw FRU Info Data
FRU Info size: 184
20: FRU # 1 Raw FRU Info Data
FRU Info size: 1024 "Shelf FRU"
20: FRU # 2 Raw FRU Info Data
FRU Info size: 1024 "Shelf FRU"
20: FRU # 3 Raw FRU Info Data
FRU Info size: 192
20: FRU # 254 Raw FRU Info Data
FRU Info size: 844 "Shelf FRU"
Back to the previous page
Done
Internet

4.19 Reset Board

The page “Reset Board” allows the user to request a reset action for a board in a specific physical slot. The target slot number must be chosen from the list.



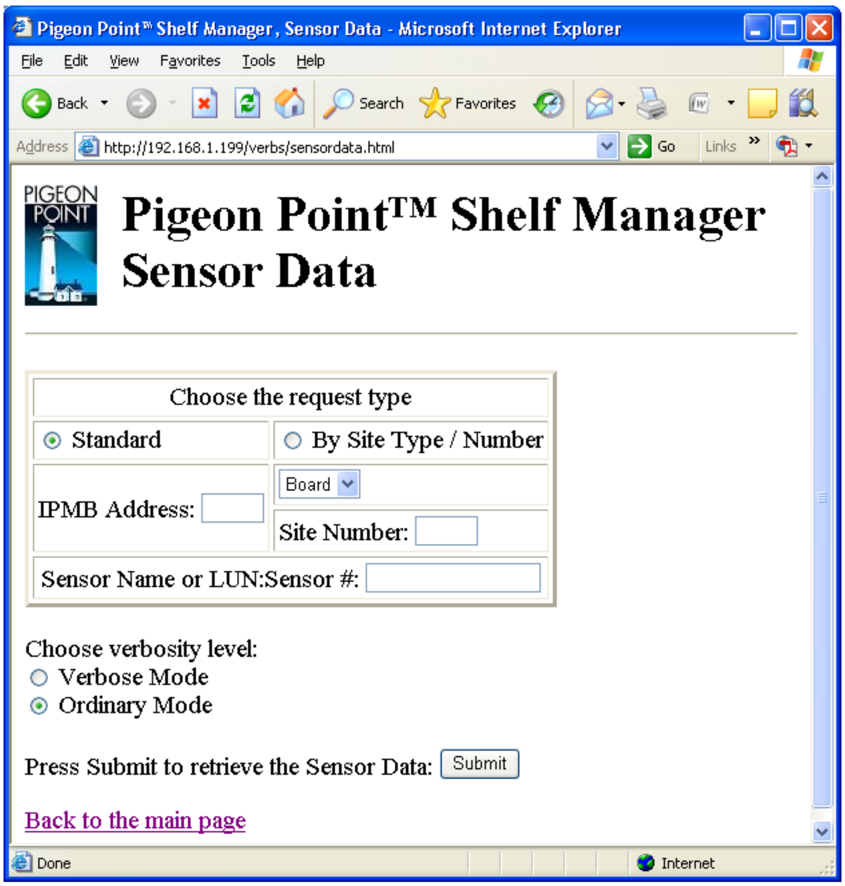
After the user chooses the target slot number and clicks the “Submit” button, the request is executed and the results page is shown, similar to the one below. This command is essentially equal to the CLI command `boardreset`.



4.20 Sensor Data

The page "Sensor Data" allows the user to specify the IPM controller address and the sensor number or name for the sensor data request. Some of the fields may be left blank; in that case:

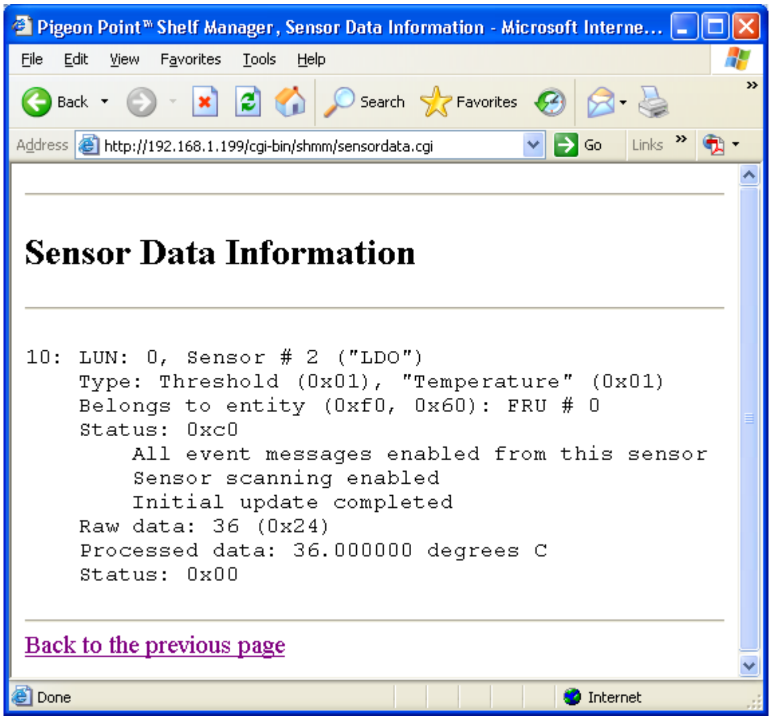
- if all of the fields are left blank, data from all known sensors on all IPM controllers is provided.
- if only the IPM controller address is specified, data from all sensors of the specified IPM controller is provided.



In the field "Sensor Name or LUN:Sensor #", the user can identify the target sensor by specifying the sensor name or specifying the sensor LUN and sensor number.

In the last case, the LUN is optional; if specified, it is separated from the sensor number with a colon. If the user specifies only the sensor number, information is returned about known sensors with the specified sensor number on all LUNs. Valid values for the LUN are 0, 1 and 3. (LUN 2 is reserved.)

After the user fills in the desired fields and clicks the "Submit" button, the request is executed and the results page is shown, similar to the one below. The output is essentially equal to the output produced by the CLI command **`sensordata`**.

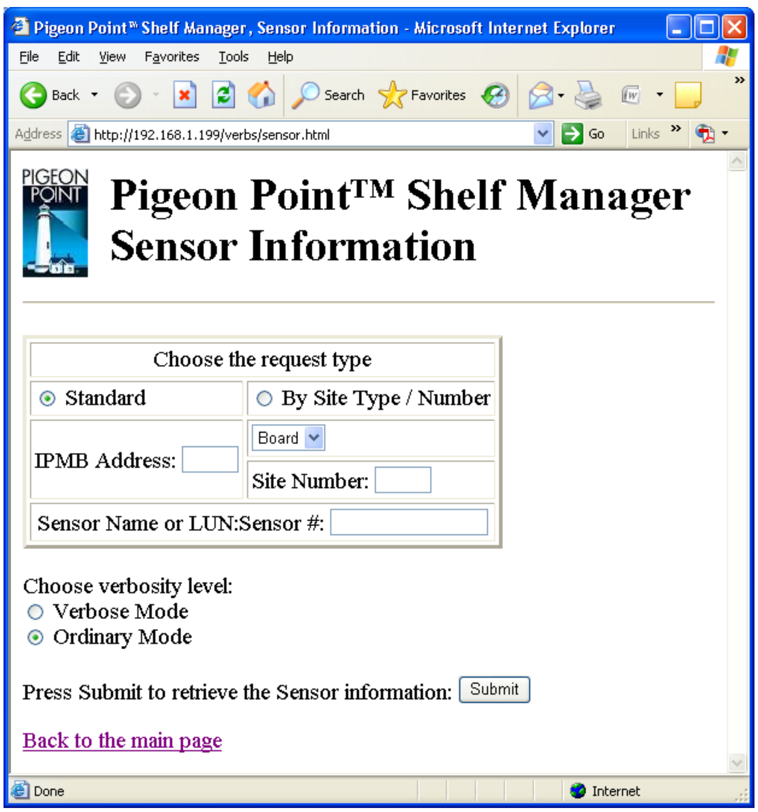

Pigeon Point™ Shelf Manager, Sensor Data Information - Microsoft Interne...
File Edit View Favorites Tools Help
Back
Search
Favorites
Address http://192.168.1.199/cgi-bin/shmm/sensordata.cgi
Go
Links
Sensor Data Information
10: LUN: 0, Sensor # 2 ("LDO")
Type: Threshold (0x01), "Temperature" (0x01)
Belongs to entity (0xf0, 0x60): FRU # 0
Status: 0xc0
All event messages enabled from this sensor
Sensor scanning enabled
Initial update completed
Raw data: 36 (0x24)
Processed data: 36.000000 degrees C
Status: 0x00
Back to the previous page
Done
Internet

4.21 Sensor Information

The page "Sensor Information" allows the user to specify the IPM controller address, sensor number or name, and verbosity mode for the sensor information request. Some of the fields may be left blank; in that case:

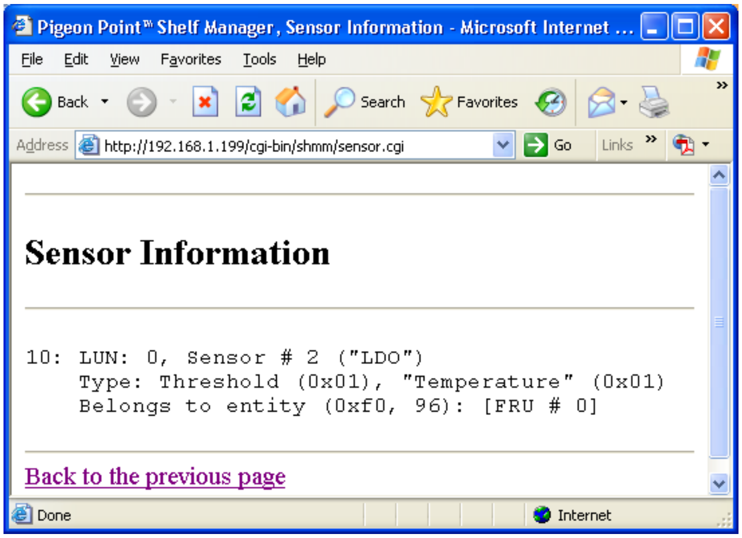
- if all of the fields are left blank, information about all known sensors on all IPM controllers is provided
- if only the IPM controller address is specified, information about all sensors of the specified IPM controller is provided.

In the field "Sensor Name or LUN:Sensor #", the user can identify the target sensor by specifying the sensor name or specifying the sensor LUN and sensor number.



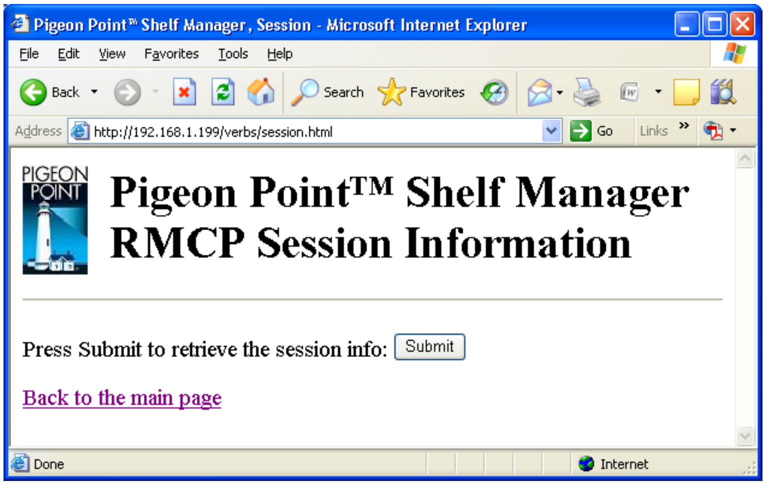
In the last case, the LUN is optional; if specified, it is separated from the sensor number with a colon. If the user specifies only the sensor number, information is returned about known sensors with the specified sensor number on all LUNs. Valid values for the LUN are 0, 1 and 3. (LUN 2 is reserved.)

After the user fills in the desired fields and clicks the "Submit" button, the request is executed and the results page is shown, similar to the one below. The output is essentially equal to the output produced by the CLI command **`sensor`**.

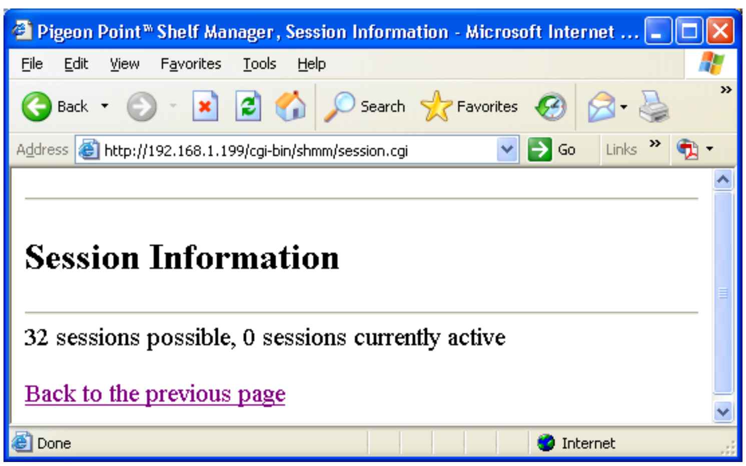

Pigeon Point™ Shelf Manager, Sensor Information - Microsoft Internet ...
File Edit View Favorites Tools Help
Back Search Favorites
Address http://192.168.1.199/cgi-bin/shmm/sensor.cgi Go Links
Sensor Information
10: LUN: 0, Sensor # 2 ("LDO")
Type: Threshold (0x01), "Temperature" (0x01)
Belongs to entity (0xf0, 96): [FRU # 0]
Back to the previous page
Done
Internet

4.22 Session Information

The page “Session Information” allows the user to obtain information about active RMCP sessions.

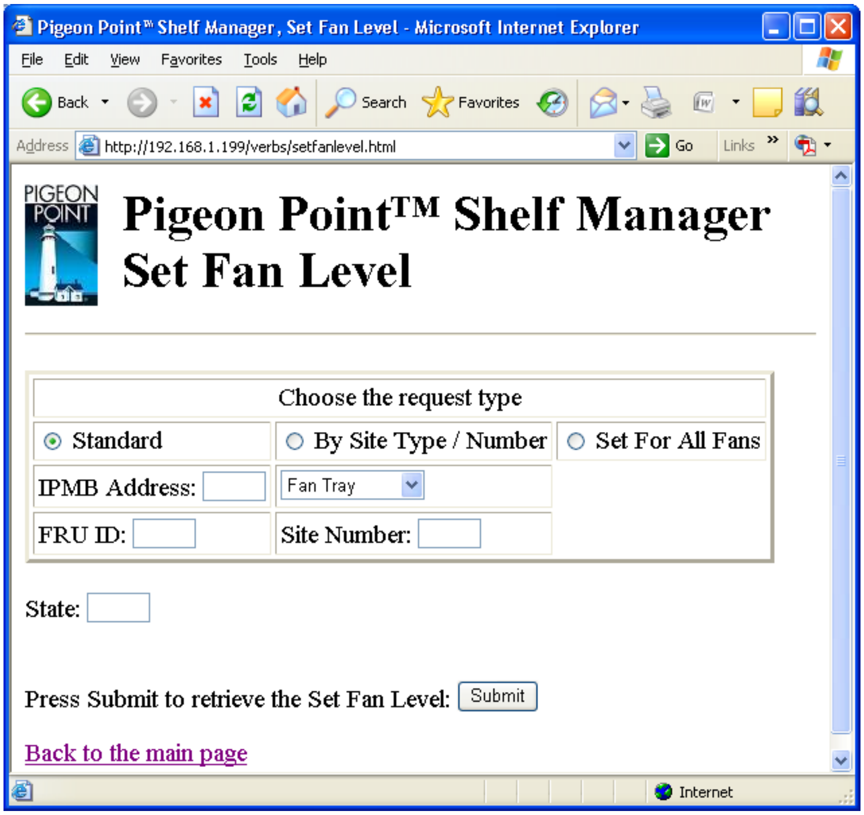


After the user clicks the “Submit” button, the request is executed and the results page is shown, similar to the one below. The output is essentially equal to the output produced by the CLI command **`session`**.

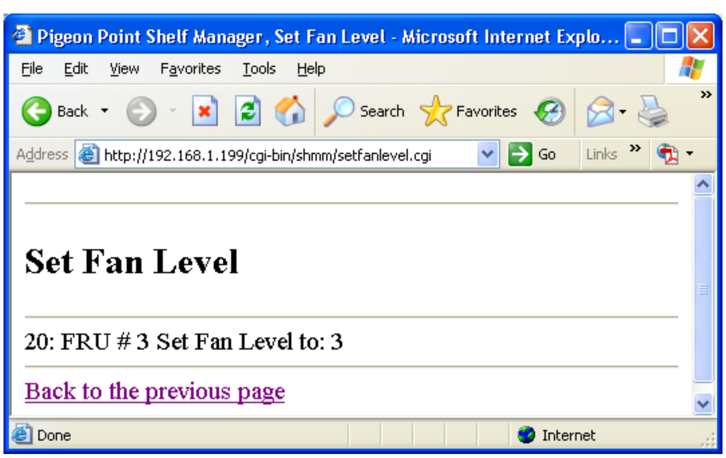


4.23 Set Fan Level

The page "Set Fan Level" allows the user to specify the IPM controller address and the FRU ID to set the fan level of the specified fan. Alternatively, using the radio button "Set For All Fans", the user can set the requested fan level for all known fans in the shelf.



After the user fills in all fields and clicks the "Submit" button, the request is executed and the results page is shown, similar to the one below. This command is essentially equal to the CLI command **`setfanlevel`**.

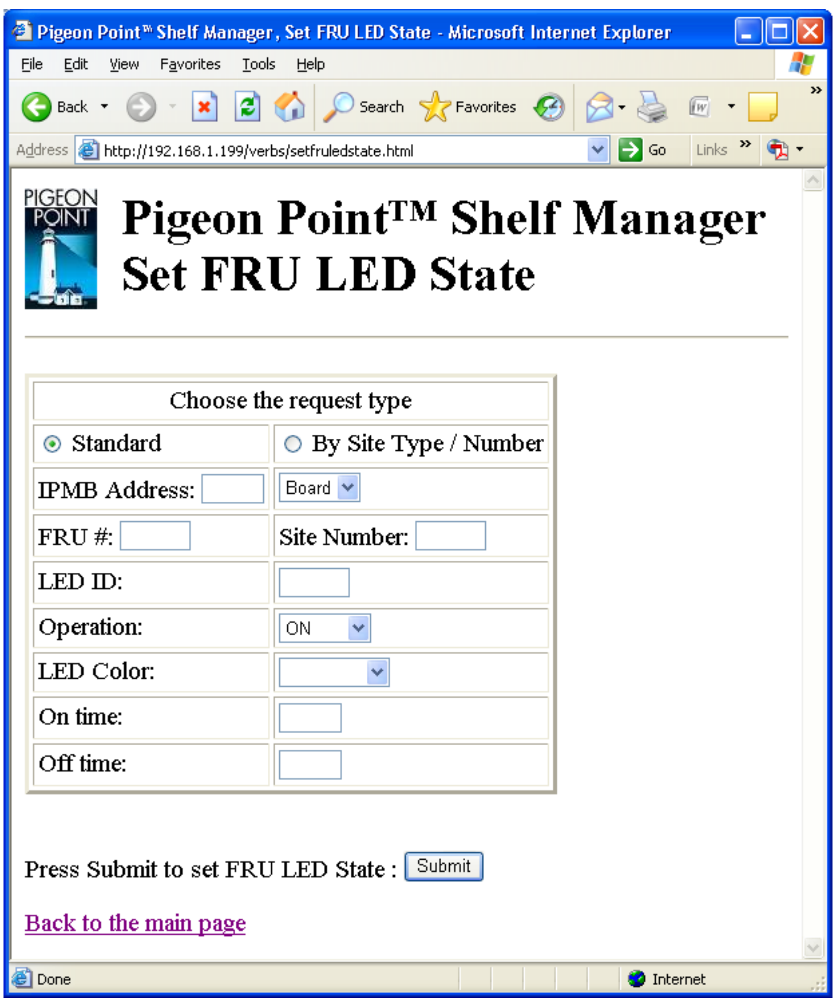


4.24 Set FRU LED State

The page "Set FRU LED State" allows the user to set the state of a specific LED or all LEDs for the given FRU. Either the IPMB address of an IPM controller and FRU device ID or a Site Type and Site Number can be specified. The "LED ID" field must be filled with either an LED ID (a numerical value) or **`ALL`**. In the latter case, the specified operation applies to all LEDs.

One of the following operations must be chosen:

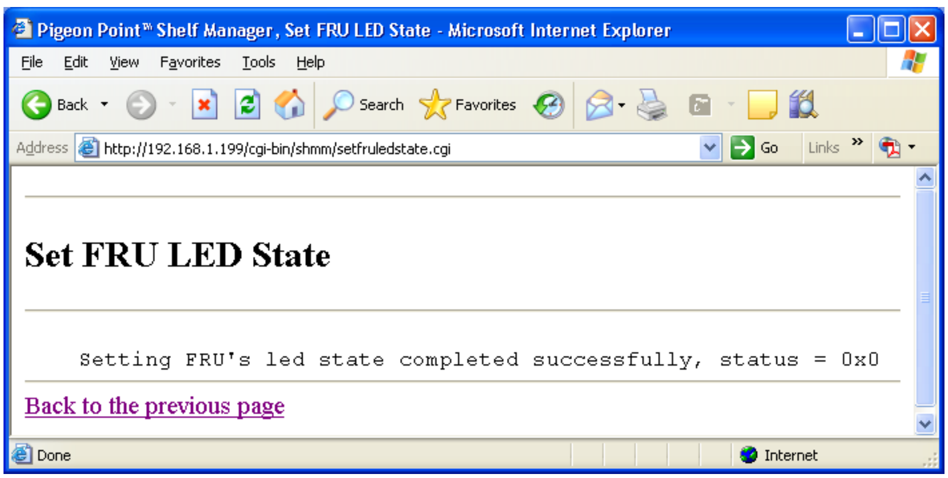
- ON – turn on the LED
- OFF – turn off the LED
- LOCAL – revert to local control of the LED
- BLINK – cause the LED to blink, repeatedly turning it on for a period of time specified in "On Time" field (in milliseconds) and then turning it off for a period of time specified in "Off Time" field (in milliseconds)
- TEST – run a lamp test for a period of time specified in "On Time" field (in milliseconds).
- For the TEST operation a value in the "On Time" field must be less then 12800 ms (12.8 sec); for the BLINK operation, values in both the "On Time" and "Off Time" fields must be within 10 – 2500 ms range.



The optional parameter "LED Color" specifies a color, via a symbolic name. If the parameter is not specified, the default LED color is used. The possible values of "LED Color" are below:

- BLUE
- RED
- GREEN
- AMBER
- ORANGE
- WHITE
- NONE

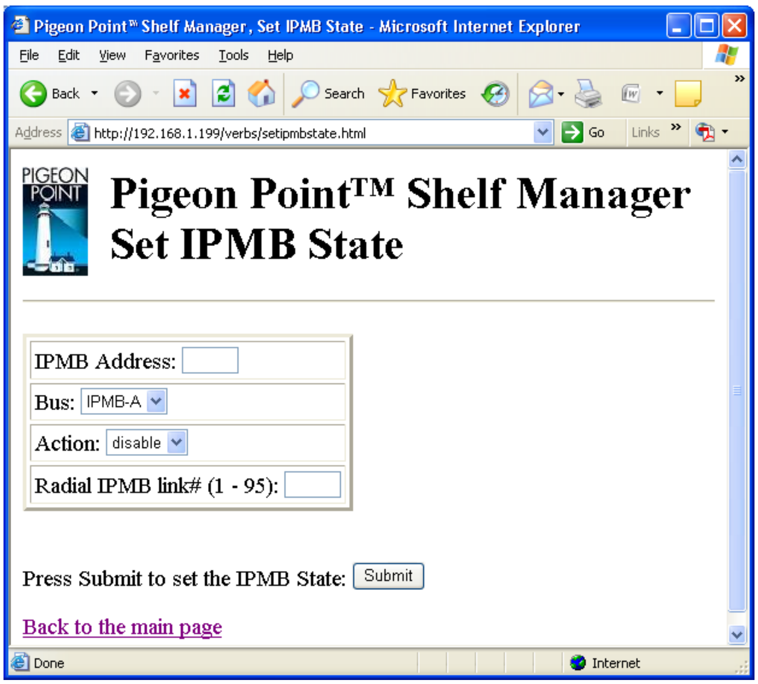
After the user fills in fields and clicks the "Submit" button, the request is executed and the results page is shown, similar to the one below. This command is essentially equal to the CLI command **`setfruledstate`**.



4.25 Set IPMB State

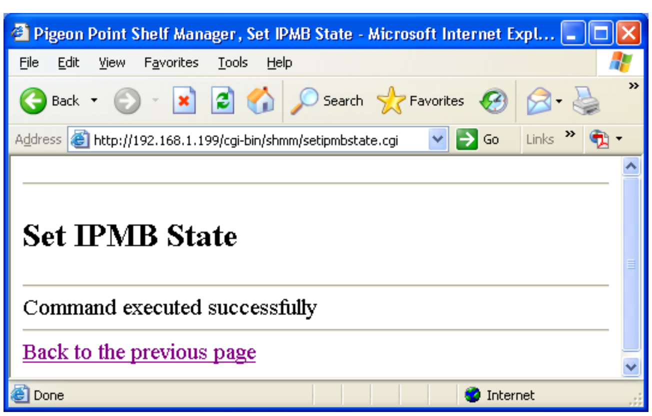
The page "Set IPMB State" allows the user to enable/disable an IPMB link on the target IPM controller. The field "IPMB Address" must contain an IPMB address of the target IPM controller.

The field "Bus" defines the bus (either IPMB-A or IPMB-B) to be enabled/disabled. The field "Action" defines the operation to be performed: "enable" – to enable link, "disable" – to disable link. In a bused environment, or in a radial environment if the target IPM controller is not an IPMB hub, the field "Radial IPMB link# (1-95)" must be left empty.



After the user fills in the request information and clicks the "Submit" button, the request is executed and the results page is generated similar to the one below.

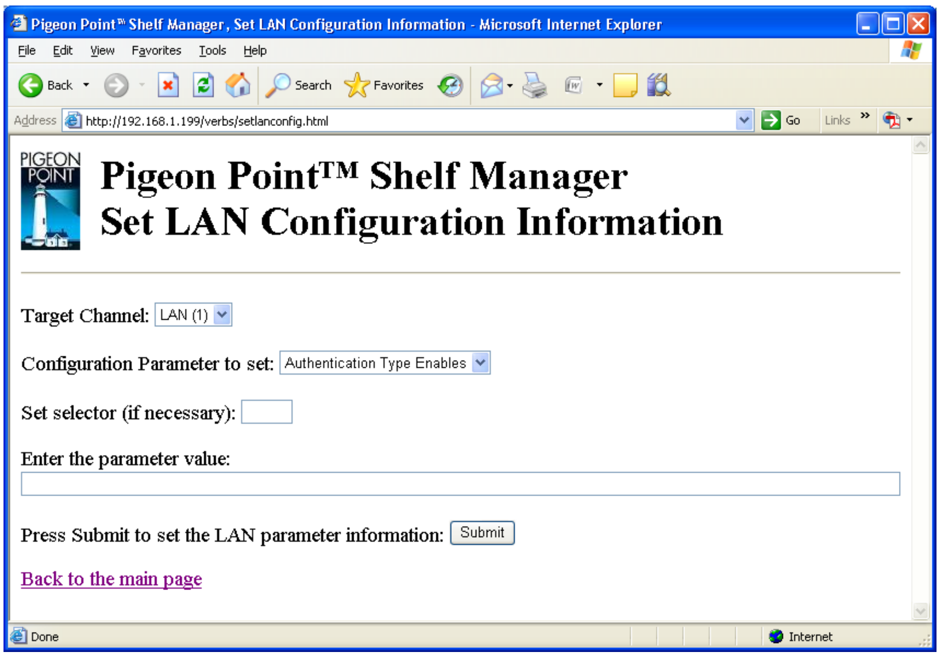
The output produced by this command is essentially the same as the output produced by the CLI command **`setipmbstate`**.



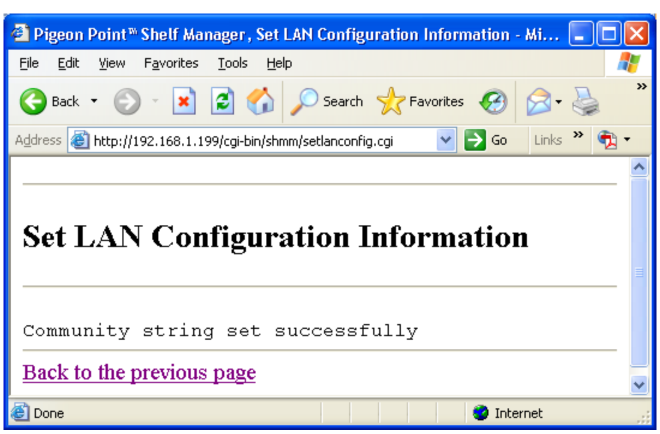
4.26 Set LAN Configuration Information

The page "Set LAN Configuration Information" allows the user to set a value of one of the LAN configuration parameters for the specified channel. The user should identify the field to be modified and the new value, via the following actions:

- choose the target channel
- choose one of the settable parameters from the drop-down configuration parameter list
- choose the set selector (item number) if applicable
- specify the parameter value according to the format for the selected parameter.



After the user specifies all necessary information and clicks the "Submit" button, the request is executed and the results page is shown, similar to the one below. This command is essentially equal to the CLI command **`setlanconfig`**.

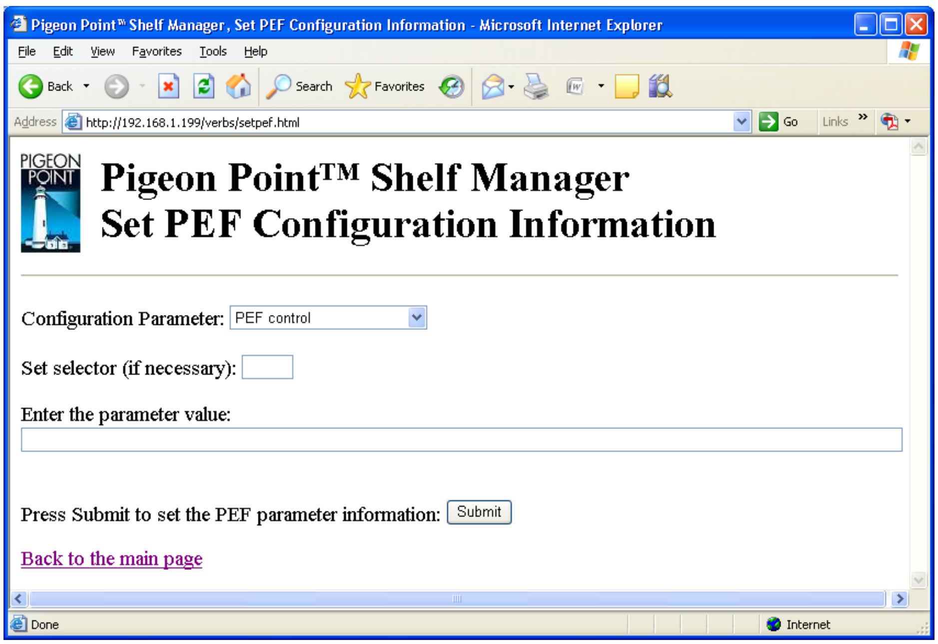


4.27 Set PEF Configuration Information

The page "Set PEF Configuration Information" allows the user to set a value of one of the PEF (Platform Event Filter) configuration parameters.

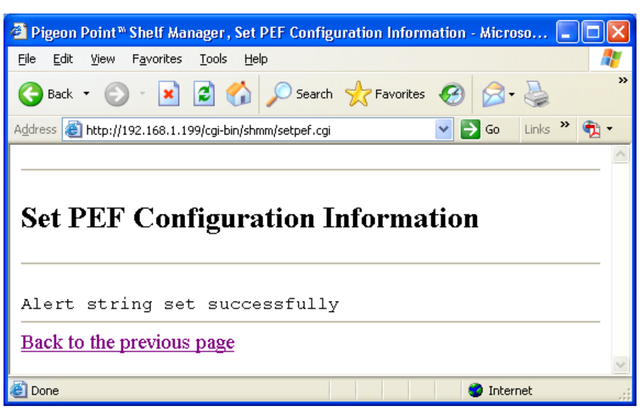
The user should identify the field to be modified and the new value, via the following actions:

- choose one of the settable parameters from the drop-down list
- choose the set selector (item number) if applicable
- specify the parameter value according to the format for the selected configuration parameter. The formats are described in the CLI command section for **`setpefconfig`**.



After the user specifies all necessary information and clicks the "Submit" button, the request is executed and the results page is shown, similar to the one below.

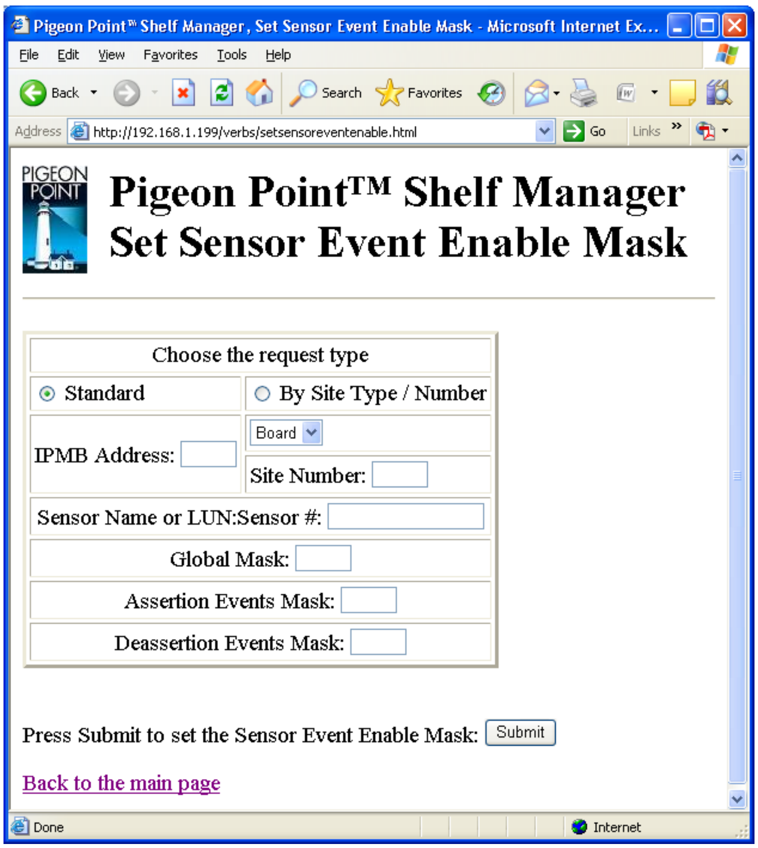
This command is essentially equal to the CLI command **`setpefconfig`**.



4.28 Set Sensor Event Enable

The page “Set Sensor Event Enable” allows the user to change 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.

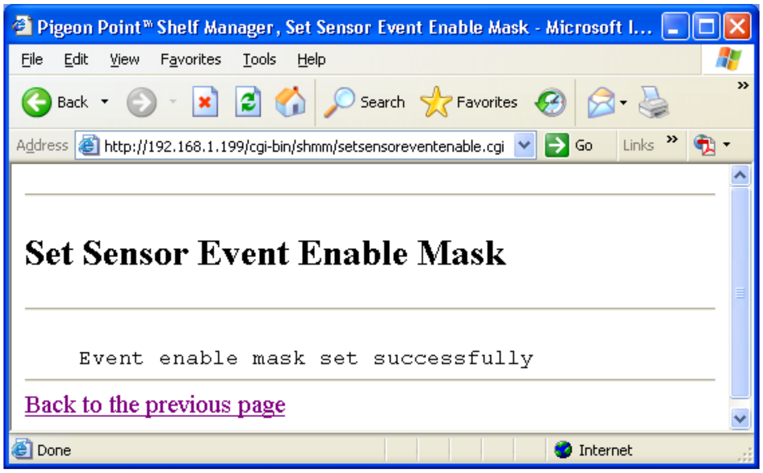
The user is allowed to qualify the sensor number with the Logical Unit Number (LUN) if the target controller supports sensors on multiple LUNs. If the LUN is omitted, the request is applied to the sensor with the specified sensor number on the lowest LUN.



For example, if the request 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 action is applied to the sensor 3 on LUN 1.

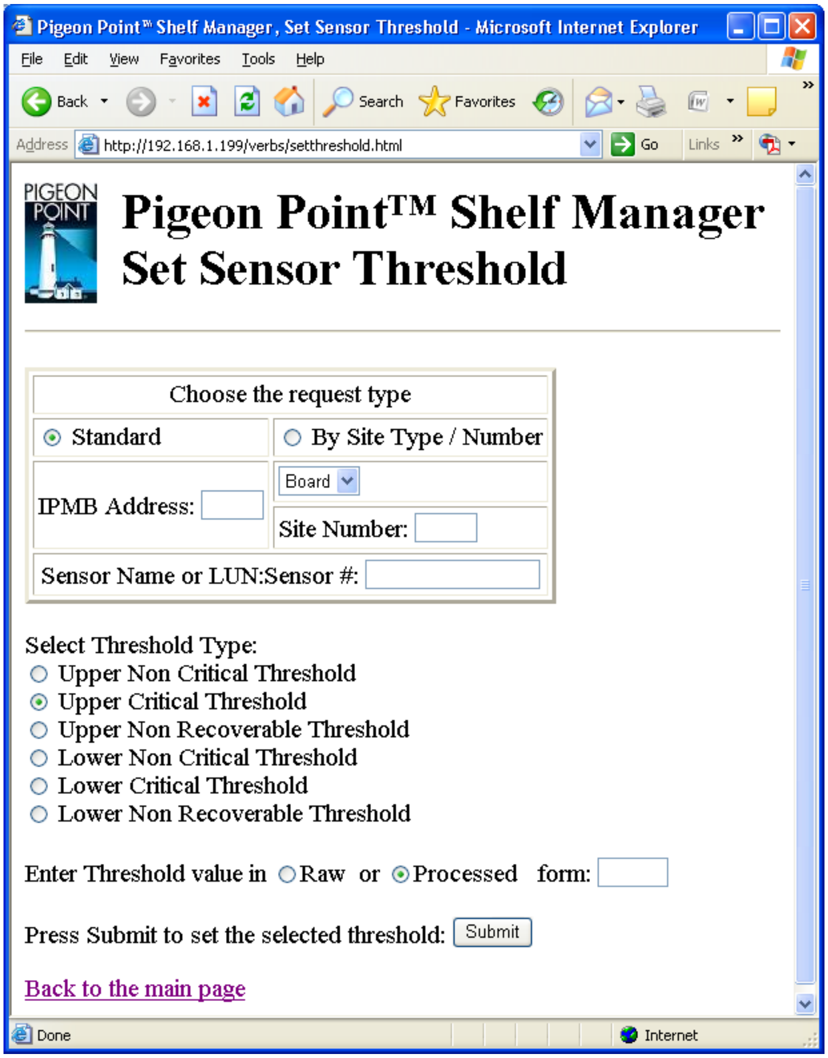
Sensor names are not qualified with LUN numbers, since it is assumed that sensor names will normally be unique within the controller. To qualify a sensor number with the LUN the user should concatenate the LUN, ‘:’ and the sensor number.

After the user fills in the request information and clicks the “Submit” button, the request is executed and the results page is generated, similar to the one below. The output produced by this command is essentially the same as the output produced by the CLI command **`setsensoreventenable`**.



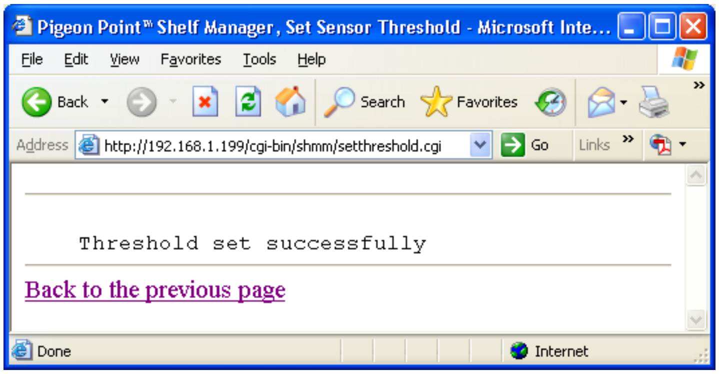
4.29 Set Sensor Thresholds

The page “Set Sensor Thresholds” allows the user to specify the IPM controller address and the sensor number or name for the threshold information request. All fields must be filled in. The new threshold value supplied by the user should be the raw byte value.



In the field “Sensor Name or LUN:Sensor #”, the user can identify the target sensor by specifying the sensor name or specifying the sensor LUN and sensor number. In the last case, the LUN is optional; if specified, it is separated from the sensor number with a colon. Valid values for the LUN are 0, 1 and 3. (LUN 2 is reserved.)

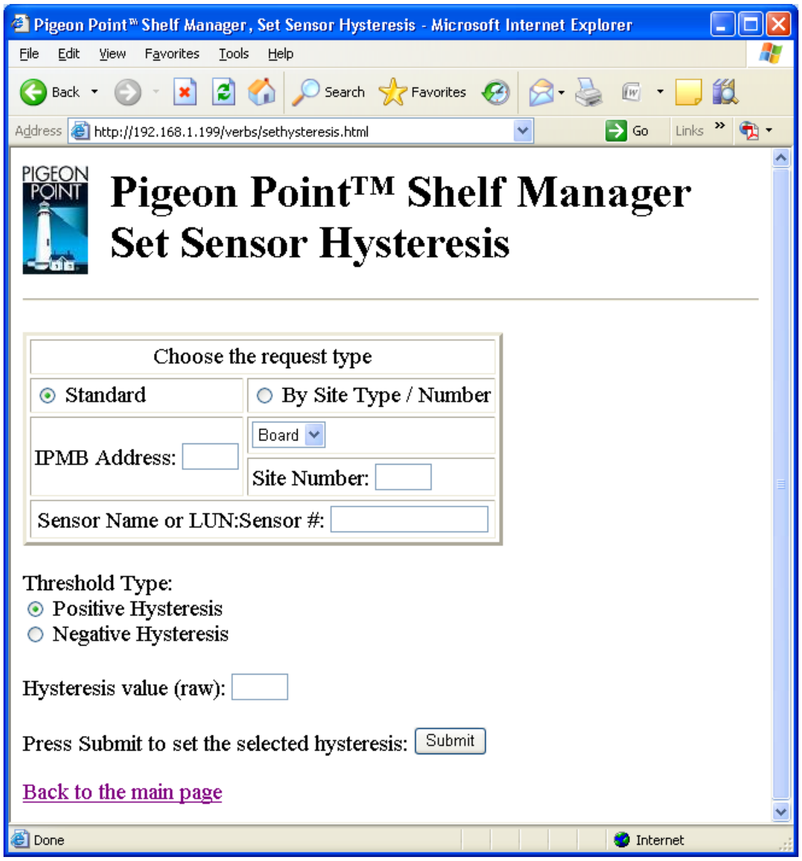
If the user specifies only the sensor number, the target sensor will be the sensor with the specified sensor number on the lowest LUN. For example, if the target IPM controller exposes sensors with the number 3 on LUNs 1 and 3, specifying sensor number 3 causes the command to affect sensor 3 on LUN 1.



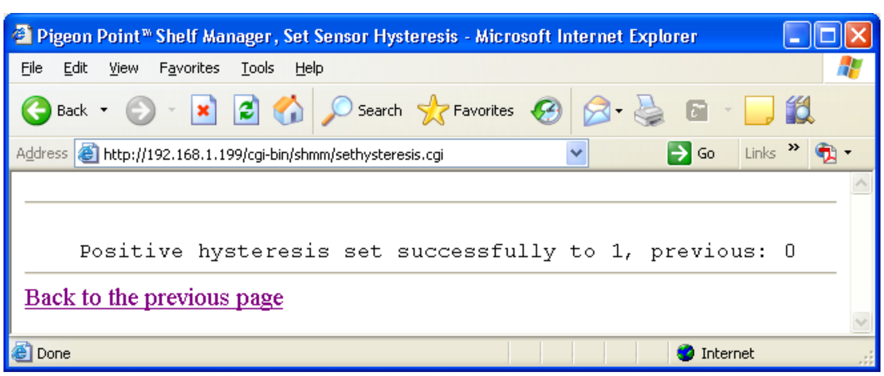
After the user fills in all fields and clicks the “Submit” button, the request is executed and the results page is shown, similar to the one below. This command is essentially equal to the CLI command **`setthreshold`**.

4.30 Set Sensor Hysteresis

The page "Set Sensor Hysteresis" allows the user to set value for the positive-going and negative-going hysteresis of the specified sensor. The user should identify the IPM controller address and the sensor number or name and the hysteresis to be set. All fields must be filled in. The new hysteresis value supplied by the user should be a raw byte value.



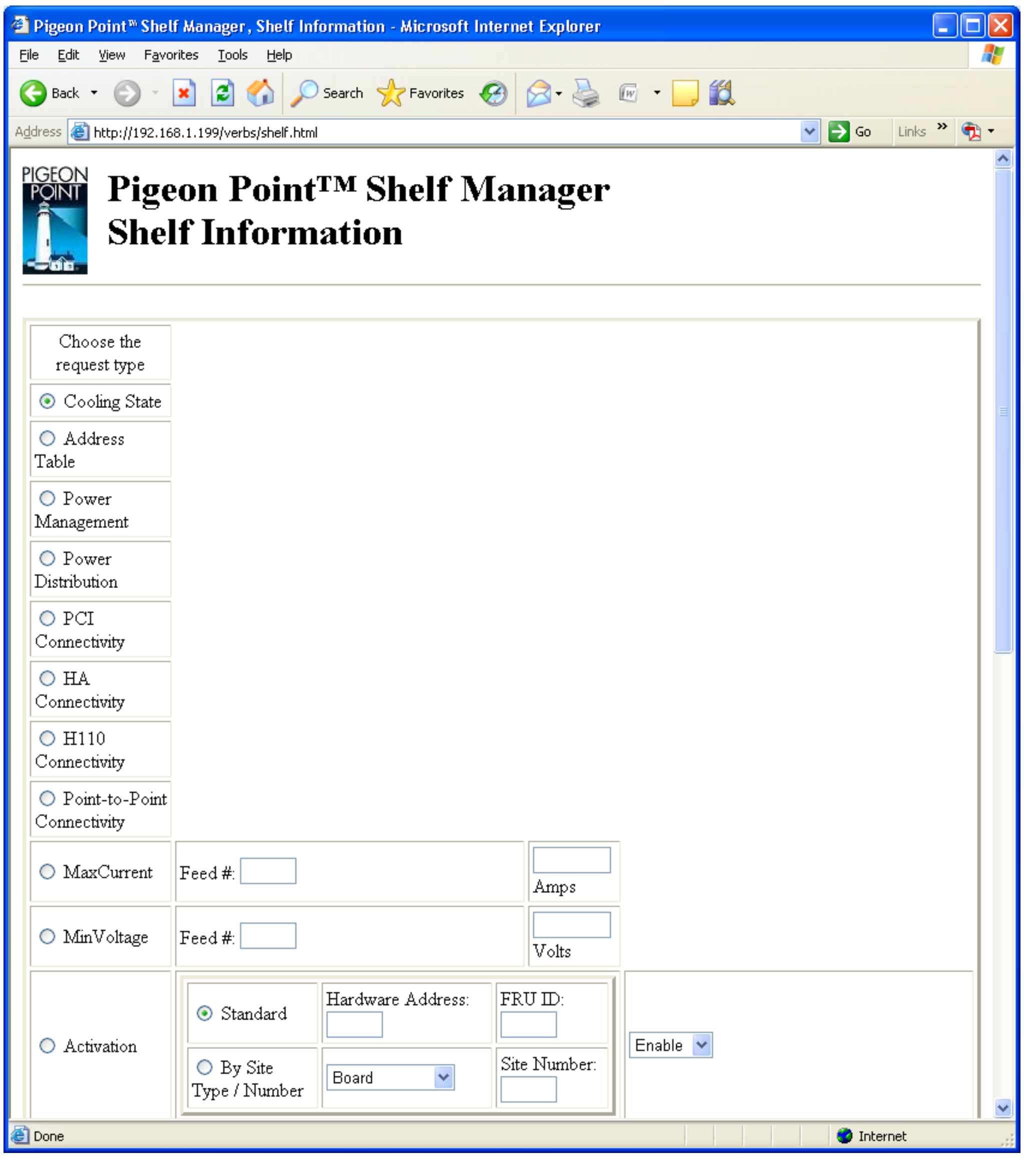
After the user fills in all fields and clicks the "Submit" button, the request is executed and the result page is shown, similar to the one below. This command is essentially equal to the CLI command **`sethysteresis`**.



4.31 Shelf Information

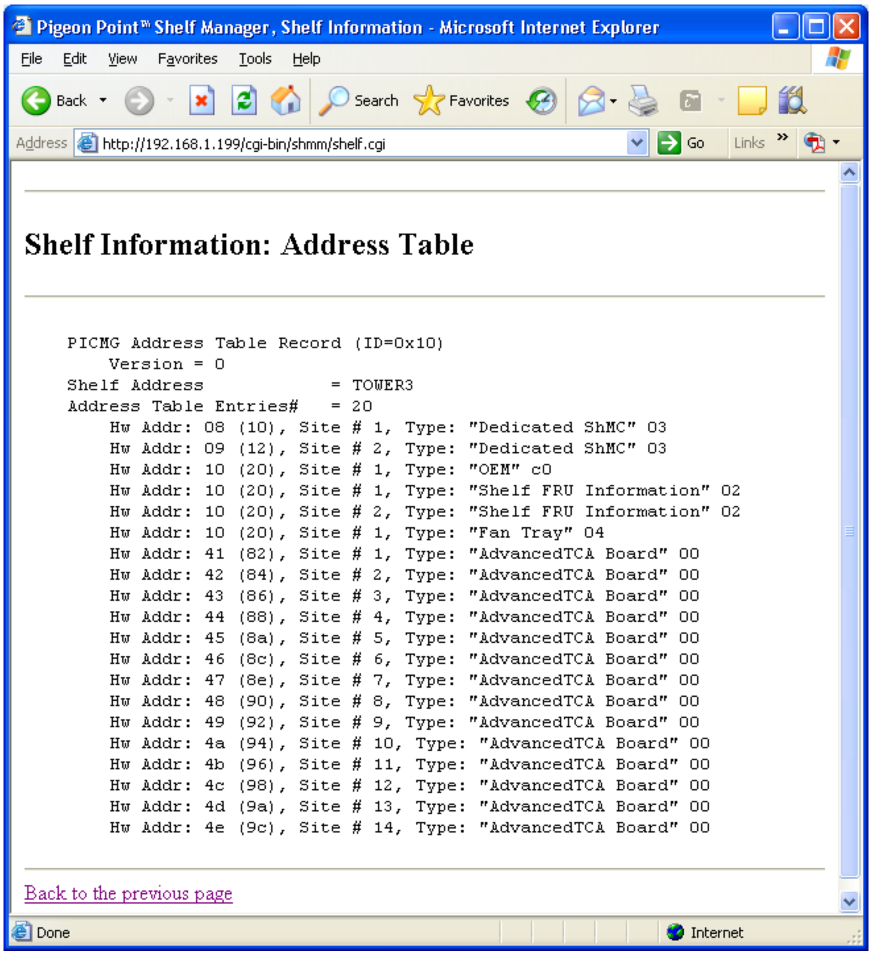
The page "Shelf Information" allows the user to request information items from the Shelf FRU Information, plus some current operating parameters of the shelf. Currently, four information types are provided:

- Cooling State
- Address Table
- Power Distribution
- Power Management.



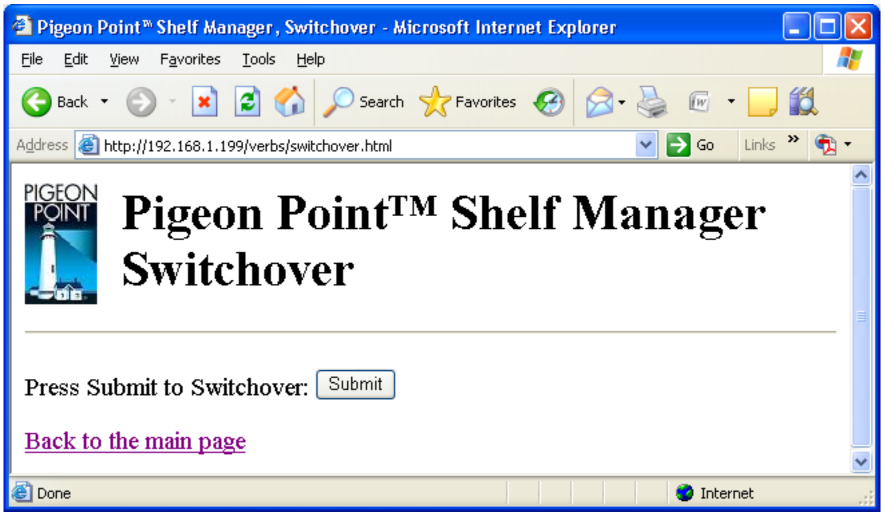
After the user chooses the information type and clicks the "Submit" button, the request is executed and the results page is shown.

The output is essentially equal to the output produced by the CLI command **`shelf`** command with a corresponding subcommand. The output page for the information type “Address Table” is shown below.

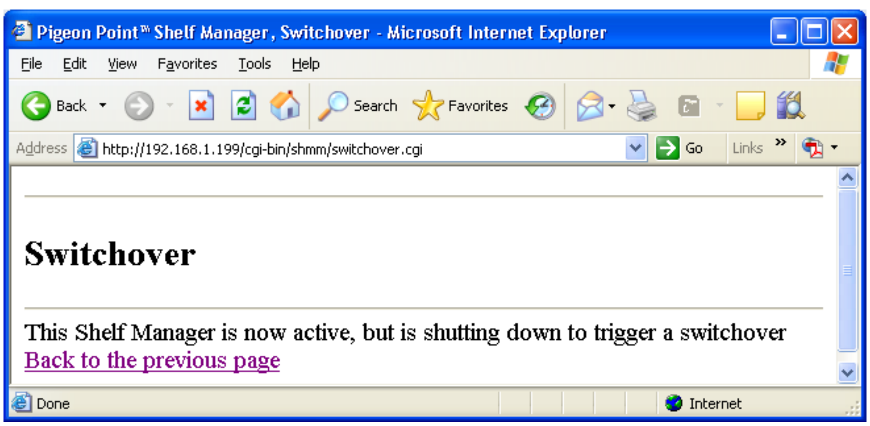


4.32 Switchover

The page “Switchover” allows the user to initiate a switchover from Active to Backup Shelf Manager.



After the user clicks the “Submit” button, the request is executed and the following results page is shown. The output is essentially equal to the output produced by the CLI command **`switchover`**. Alternatively, the browser may report on an error because the Shelf Manager shuts down before the boa demon produces the results page.



4.33 System Event Log

The page “System Event Log” allows the user to specify parameters for the System Event Log (SEL) information request or clear the event log.

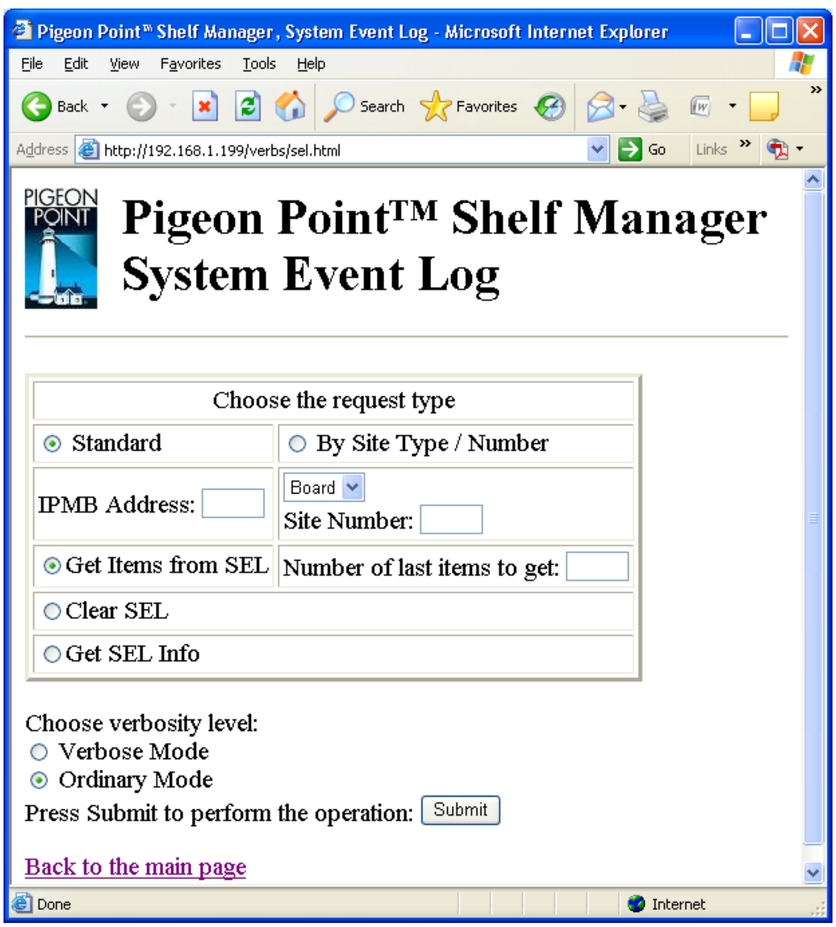
To retrieve system event log information, choose the upper radio option “Get Items from SEL”.

Some of the fields may be left blank; in that case:

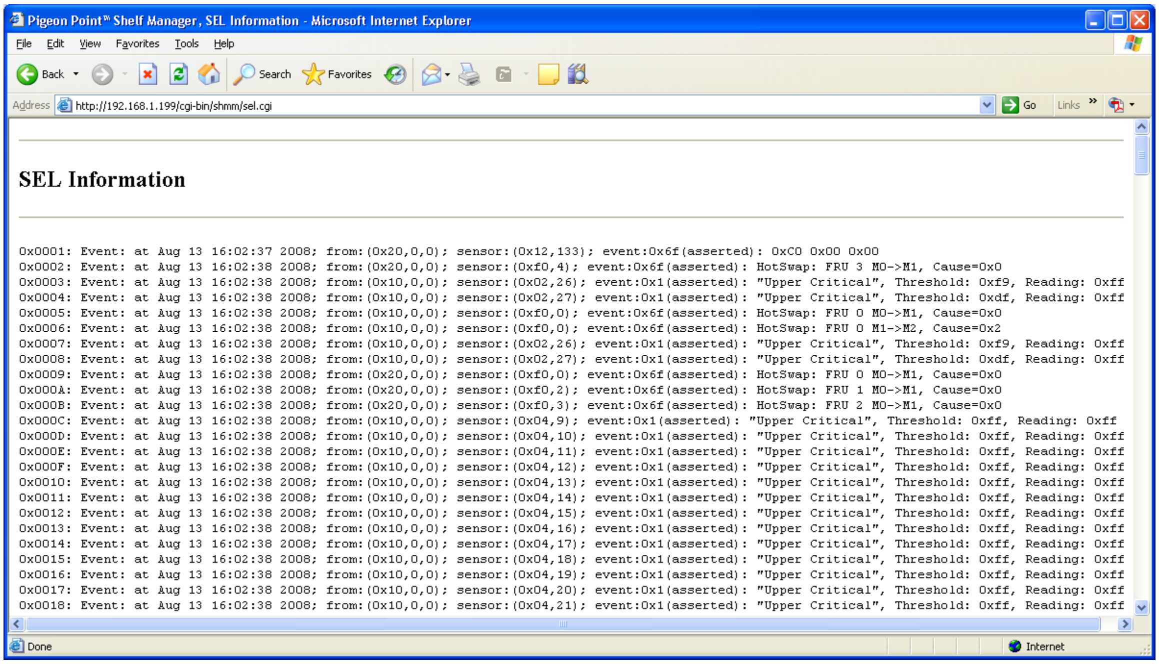
- if the IPM controller address is left blank, the SEL is accessed on the Shelf Manager (IPMB address 20h).
- if the parameter “Number of last items to get” is omitted, the entire SEL is retrieved.

To clear the system event log, choose the middle radio option. The IPM controller address may be left blank; in that case, the SEL on the Shelf Manager (IPMB address 20h) is cleared.

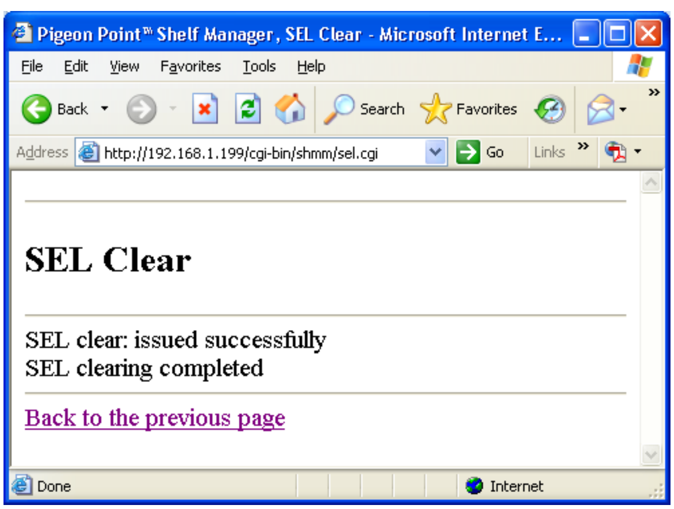
To retrieve information about the system event log, choose the lower radio option. The IPM controller address may be left blank; in that case, the information about the SEL on the ShMC (IPMB address 20h) is provided.



After the user chooses the upper radio option and fills in desired fields and clicks the “Submit” button, the request is executed and the results page is shown, similar to the one below. The output is essentially equal to the output produced by the CLI command **sel**.

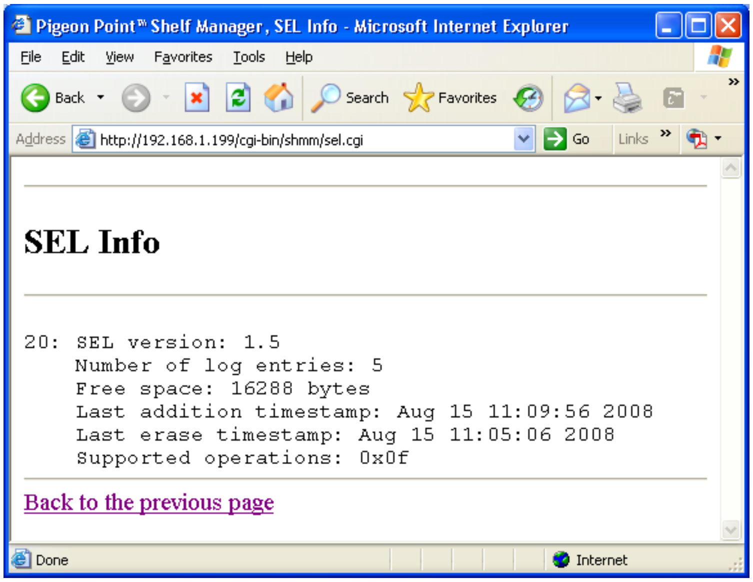


After the user chooses the middle radio option and clicks the “Submit” button, the request to clear SEL is executed and the results page is shown, similar to the one below. This command is essentially equal to the CLI command **sel clear**.



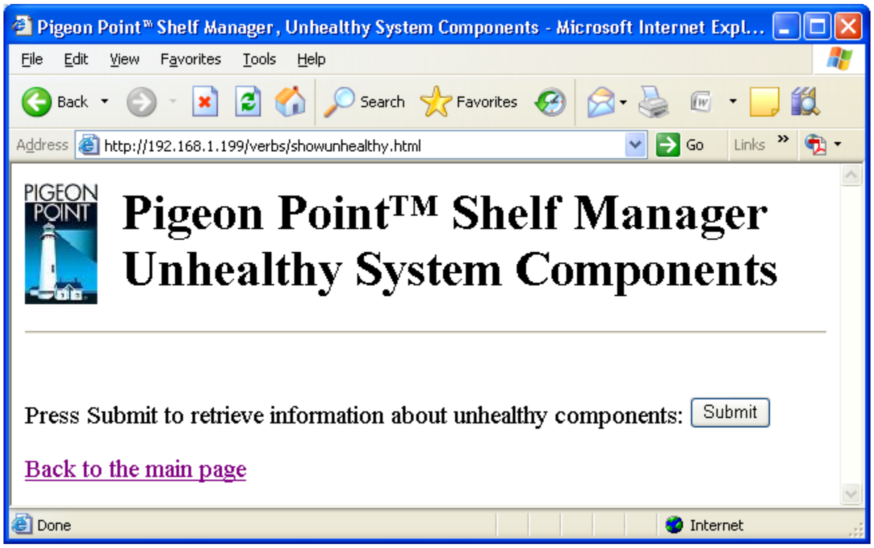
After the user chooses the lower radio option and clicks the “Submit” button, the request to get information about SEL is executed and the results page is shown, similar to the one below.

This command is essentially equal to the CLI command **sel info**. The fields in the lower form may be left blank.


Pigeon Point™ Shelf Manager, SEL Info - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address http://192.168.1.199/cgi-bin/shmm/sel.cgi
Go
Links
SEL Info
20: SEL version: 1.5
Number of log entries: 5
Free space: 16288 bytes
Last addition timestamp: Aug 15 11:09:56 2008
Last erase timestamp: Aug 15 11:05:06 2008
Supported operations: 0x0f
Back to the previous page
Done
Internet

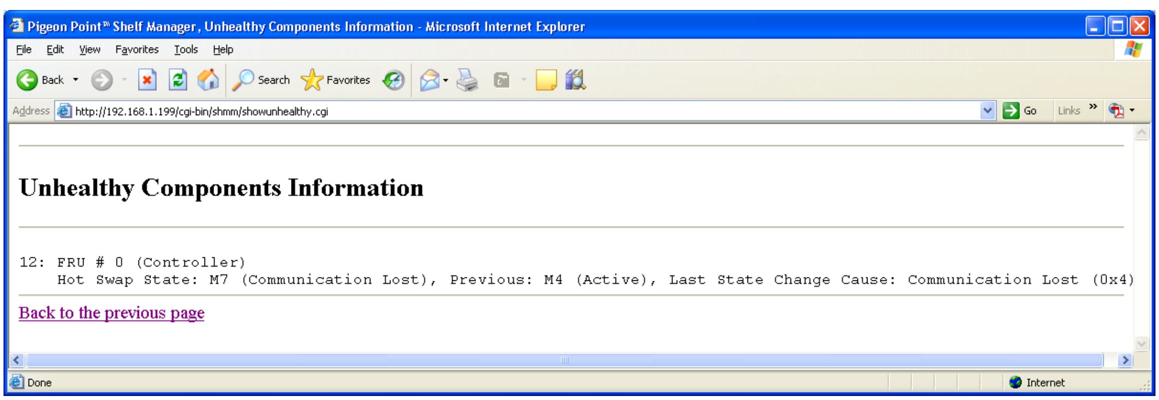
4.34 Unhealthy System Components

The page "Unhealthy System Components" allows the user to request information about unhealthy system components. To request information, the user should press the "Submit" button.



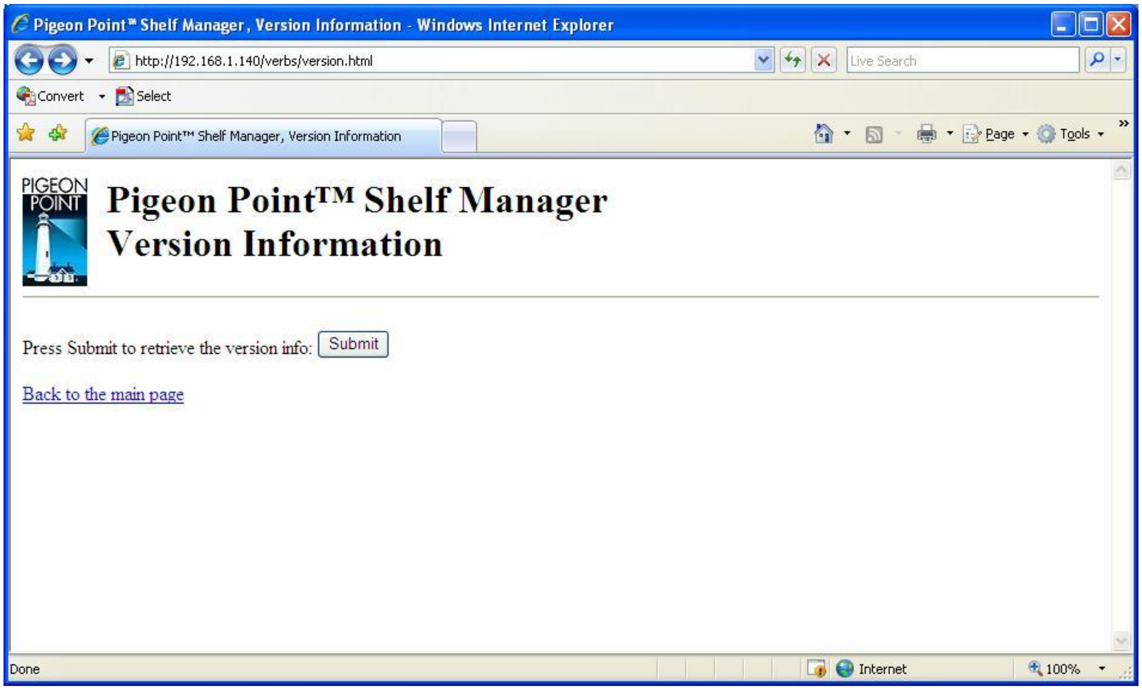
After the user clicks the "Submit" button, the request is executed and the results page is shown.

Usually this page will be empty, but may show some unhealthy components as in the example below. The output is essentially equal to the output produced by the CLI command **`showunhealthy`**.

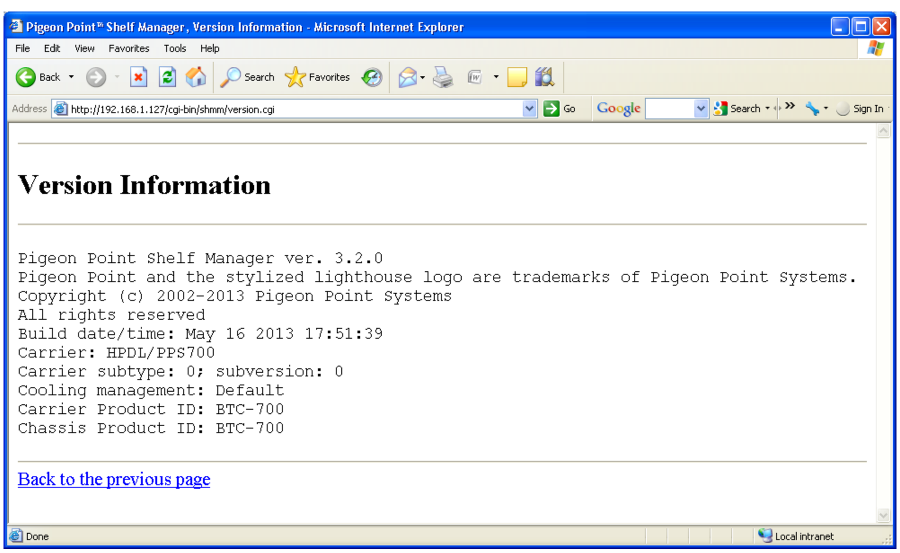


4.35 Version

The page "Version" allows the user to request information about the software version of the Pigeon Point Shelf Manager.



After the user clicks the "Submit" button, the request is executed and a results page similar to the following one is shown. The output is essentially equal to the output produced by the CLI command **`version`**.



5 Simple Network Management Protocol (SNMP)

The Pigeon Point Shelf Manager supports a Simple Network Management Protocol (SNMP) interface to the shelf configuration and control variables. The following groups of variables are supported by this interface:

- IPM Controllers
- FRU Devices
- Sensors
- Boards
- Shelf/shelves
- System Event Log
- LAN Configuration Parameters
- PEF Configuration Parameters

According to SNMP rules, the variables from these groups are represented via a hierarchical data model, each variable identified via an object identifier (OID). These object identifiers have a common root OID:

```
iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).pps(16394).p
roducts(2).chassis-management(1).ipm-sentry-shelf-manager(1)
```

16394 is a unique private Pigeon Point Systems enterprise number obtained from IANA. The root OID in the remainder of this section is denoted as **`<ROOT_OID>`**.

The structure of the branches of the SNMP variables tree is described in the following subsections. The definition of SNMP variables provided by the Shelf Manager is contained in a Management Information Base (MIB) file. This file should be installed on the management system (the one that interacts with the Shelf Manager over the network). It depends on the SNMP client software how the MIB file should be installed on the management system; usually this file must be placed in a special location on the management system and compiled with a MIB compiler.

The Shelf Manager SNMP interface provides two groups of MIB variables: Basic and Advanced. The Basic MIB variables provide user-friendly access to the information that can be retrieved from the Pigeon Point Shelf Manager. It ensures that all objects are indexed naturally for the user. Also, information in the Basic MIB variables is processed to be more readable and easily understandable for a user who is inexperienced with the details of IPMI.

The Advanced MIB variables assume that the user is experienced enough to use the indexing by IPMB address and FRU ID which are natural for the objects described in the IPMI 1.5.1 and PICMG 3.0 specifications. Accessing the variables described in the Advanced MIB variables is more simple and robust than accessing the Basic MIB variables, but the variables are delivered to the user in non-processed format.

It is worth mentioning that in certain cases, using the Advanced MIB variables, information can be retrieved even though an “object unavailable” error is reported when using Basic MIB variables.

This happens because the information is internally handled differently: in most cases the Basic MIB variables access Shelf FRU Information that can be unavailable, corrupted or contain incomplete information. Access to Advanced MIB variables in most cases does not require retrieving data from the Shelf FRU: information that is cached internally in the Shelf Manager is used instead.
The user can use both Basic and Advanced MIB variables simultaneously.

In redundant configurations, the external IP address is always maintained by the active Shelf Manager and is switched over to the backup Shelf Manager when the general switchover takes place. Therefore, if the client uses the SNMP interface with the external IP address of the Shelf Manager in redundant configurations, it always communicates to the active Shelf Manager.

The backup Shelf Manager can however be accessed via SNMP, if it exposes a private IP address. In that case, Basic MIB variables are not supported; in the Advanced MIB variable tree, only the Shelf variables (see 5.2.7) are supported.

Using the existing U-Boot variable **`ipaddr`**, each ShMM (both active and backup) can be assigned its own IP address for the Ethernet adapter "eth0", which will be available immediately after Monterey Linux starts on a given ShMM. On the active ShMM, this IP addresses will coexist with the RMCP address on Ethernet adapter 0.

On the backup Shelf Manager, this ShMM-specific IP address will be preserved across switchovers. That is, both active and backup ShMMs are always accessible via these ShMM-specific addresses, but the RMCP address is always served by the active Shelf Manager. Please see the Pigeon Point Shelf Manager User Guide for additional background on this topic.

It should be mentioned that access to some SNMP variables may require FRU data read or write operations to be invoked. In some cases an entire FRU Info section (Board Info for example) is retrieved as part of this process, and access to such variables may take a rather long time. However, once retrieved, FRU information is cached and any further access to this data will use the cache, and will be faster.

5.1 Basic MIB Variables

5.1.1 Board Variables

The variables defined in this section contain information about the CompactPCI boards in 2.x systems or ATCA boards in ATCA systems. This information is provided in the form of an SNMP table. Each entry in this table provides information about a single board. Entries are indexed by a Physical Slot number, which is equal to the site number. This group of variables is uses the prefix **`board-basic`** to distinguish them from the board variables in described in the Advanced MIB Variables section.

Board basic information variables have the following OID:
<ROOT_OID>.32.1.<var>.<boardnum>

Here **<var>** is the index of a particular variable in the table entry describing a particular board slot. The variable indices are defined in the table below. **<boardnum>** is the Physical Board number.

Table 15 Board Variable Indices

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
board-basic-slot-number	1	INTEGER	Read-only	Table entry index, equal to **<boardnum>**.
board-basic-present	2	INTEGER	Read-only	1 – if board is present in the slot, 0 – otherwise.
board-basic-healthy	3	INTEGER	Read-only	1 – if board is present and healthy, 0 – otherwise.
board-basic-reset	4	INTEGER	Read-write	When reading: 1 – if board is in the reset state, 0 – otherwise. Writing 1 to this variable triggers a reset of the specified board.
board-basic-powered	5	INTEGER	Read-write	When reading: 1 – if board is the powered state, 0 – otherwise, -1 – if information is unavailable. Writing to this variable powers the specified board ON (if value=1) or OFF (if value=0). The variable currently returns an accurate value only on CompactPCI shelves where a radial BD_SEL# signal directly corresponds to the state of backend power for the CompactPCI board.

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
`board-basic-slave-address`	6	INTEGER	Read-only	8-bit Slave address of the IPM controller representing this board on IPMB. If the unit in question is not currently installed, then return -1. If the unit in question is installed, but does not have an active/working IPM controller, return 32 (20h). If the unit is installed and has an IPM controller, return the IPM controller's slave address.
`board-basic-fru-device-id`	7	INTEGER	Read-only	The FRU Device ID of the board. If the unit in question is not currently installed, then return -1. If the unit in question is installed, but does not have an active/working IPM controller, return the control FRU ID. The control FRU ID is used in conjunction with the BMC IPMI address (20h), and represents the board to the BMC so that it can be managed via the IPMI interface. This is applicable to CompactPCI systems only. If the unit is installed and has an IPM controller, returns 0.
`board-basic-fruinfo-product-area-present`	8	INTEGER	Read-only	1 – if the product area is present within the board FRU Information, 0 – otherwise.
`board-basic-fruinfo-product-manufacturer`	9	DisplayString	Read-only	Returns the product manufacturer from the board FRU Information or “N/A”.

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
`board-basic-fruinfo-product-name`	10	DisplayString	Read-only	Returns the product name from the board FRU Information or “N/A”.
`board-basic-fruinfo-product-part-model-number`	11	DisplayString	Read-only	Returns the product part model number from the board FRU Information or “N/A”.
`board-basic-fruinfo-product-version-number`	12	DisplayString	Read-only	Returns the product version from the board FRU Information or “N/A”.
`board-basic-fruinfo-product-serial-number`	13	DisplayString	Read-only	Returns the product serial number from the board FRU Information or “N/A”.
`board-basic-fruinfo-board-area-present`	14	INTEGER	Read-only	1 – if the board area is present within the board FRU Information, 0 – otherwise.
`board-basic-fruinfo-board-manufacturer`	15	DisplayString	Read-only	Returns the board manufacturer from the board FRU Information or “N/A”.
`board-basic-fruinfo-board-product-name`	16	DisplayString	Read-only	Returns the board product name from the board FRU Information or “N/A”.
`board-basic-fruinfo-board-serial-number`	17	DisplayString	Read-only	Returns the board serial number from the board FRU Information or “N/A”.
`board-basic-fruinfo-board-part-number`	18	DisplayString	Read-only	Returns the board part number from the board FRU Information or “N/A”.
`board-basic-fruinfo-board-manufacture-time`	19	DisplayString	Read-only	Returns the board manufacturing time: the number of seconds since 00:00:00, January 1, 1970, Coordinated Universal Time (UTC); -1 if the corresponding field is not present in the board FRU information.
`board-basic-fruinfo-product-asset-tag`	20	DisplayString	Read-only	Returns the product asset tag from the board FRU Information or “N/A”.

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
board-basic-fruinfo-product-fru-file-id	21	DisplayString	Read-only	Returns the product FRU File ID from the board FRU Information or “N/A”.
board-basic-fruinfo-board-fru-file-id	22	DisplayString	Read-only	Returns the board FRU File ID from the board FRU Information or “N/A”.

For example, to check the powered state of the board in slot 8, use the following OID:
<ROOT_OID>.32.1.5.8

5.1.2 Fan Tray Variables

The variables defined in this section contain information about the Fan Trays in the system. This information is provided in the form of an SNMP table. Each entry in this table provides information about a single Fan Tray. Entries are indexed by a physical Fan Tray number which is equal to Fan Tray site number.

Fan Tray information variables have the following OID:
<ROOT_OID>.33.1.<var>.<fantraynum>

Here **<var>** is the index of a particular variable in the table entry describing a particular Fan Tray slot. The variable indices are defined in the table below. **<fantraynum>** is the Physical Fan Tray number.

Table 16 Basic Fan Tray Variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
fantray-slot-number	1	INTEGER	Read-only	Table entry index, equal to **<fantraynum>**.
fantray-present	2	INTEGER	Read-only	1 – if fan tray is present in the slot, 0 – otherwise.
fantray-healthy	3	INTEGER	Read-only	1 – if fan tray is present and healthy, 0 – otherwise.
fantray-health-led	4	INTEGER	Read-write	When reading: the led state is returned (0 = off, 1 = on). Writing to this variable turns the led on (value=1) or off (value=0). This variable is available in 2.x systems only. In ATCA systems it always is equal to -1.

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
`fantray-slave-address`	5	INTEGER	Read-only	The 8-bit Slave address of the IPM controller representing this Fan Tray on IPMB. If the unit in question is not currently installed, then return -1. If the unit in question is installed, but does not have an active/working IPM controller, return 32 (20h). If the unit is installed and has an IPM controller, return the IPM controller's slave address.
`fantray-fru-device-id`	6	INTEGER	Read-only	The FRU Device ID of the fan tray. If the unit in question is not currently installed, then return -1. If the unit in question is installed, but does not have an active/working IPM controller, return the control FRU ID. The control FRU ID is used in conjunction with the BMC IPMI address (20h), and represents the fan to BMC so that the fan tray can be managed via the IPMI interface. This is applicable to CompactPCI systems only.
`fantray-fruinfo-product-area-present`	7	INTEGER	Read-only	1 – if the product area is present within the fan tray FRU Information, 0 – otherwise.
`fantray-fruinfo-product-manufacturer`	8	DisplayString	Read-only	Returns the product manufacturer from the fan tray FRU Information, or “N/A”.
`fantray-fruinfo-product-name`	9	DisplayString	Read-only	Returns the product name from the fan tray FRU Information, or “N/A”.

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
`fantray-fruinfo-product-part-model-number`	10	DisplayString	Read-only	Returns the product part model number from the fan tray FRU Information, or “N/A”.
`fantray-fruinfo-product-version-number`	11	DisplayString	Read-only	Returns the product version from the fan tray FRU Information, or “N/A”.
`fantray-fruinfo-product-serial-number`	12	DisplayString	Read-only	Returns the product serial number from the fan tray FRU Information, or “N/A”.
`fantray-fruinfo-board-area-present`	13	INTEGER	Read-only	1 – if the board area is present within the fan tray FRU Information, 0 – otherwise.
`fantray-fruinfo-board-manufacturer`	14	DisplayString	Read-only	Returns the board manufacturer from the fan tray FRU Information, or “N/A”.
`fantray-fruinfo-board-product-name`	15	DisplayString	Read-only	Returns the board product name from the fan tray FRU Information, or “N/A”.
`fantray-fruinfo-board-serial-number`	16	DisplayString	Read-only	Returns the board serial number from the fan tray FRU Information, or “N/A”.
`fantray-fruinfo-board-part-number`	17	DisplayString	Read-only	Returns the board part number from the fan tray FRU Information, or “N/A”.
`fantray-fruinfo-board-manufacture-time`	18	INTEGER	Read-only	Returns the board manufacturing time: the number of seconds since 00:00:00, January 1, 1970, Coordinated Universal Time (UTC); -1 if the corresponding field is not present in the fan tray FRU information.

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
`fantray-fan-level`	19	OCTET STRING (SIZE(1..3)	Read-write	Returns Override Fan Level, Local Control Fan Level, Local Control Enable State as reported by the controller in a "Get Fan Level" response on reading. Local Control Fan Level and Local Control Enable State are optional fields. When writing, byte values for Fan Level and optional Local Control Enable State should be supplied as in a "Set Fan Level" command request.
`fantray-fruinfo-product-asset-tag`	20	DisplayString	Read-only	Returns the product asset tag from the fan tray FRU Information, or "N/A".
`fantray-fruinfo-product-fru-file-id`	21	DisplayString	Read-only	Returns the product FRU File ID from the fan tray FRU Information, or "N/A".
`fantray-fruinfo-board-fru-file-id`	22	DisplayString	Read-only	Returns the board FRU File ID from the fan tray FRU Information, or "N/A".

For example, to check the led state of the Fan Tray # 8, use the following OID:
`<ROOT_OID>.33.1.4.8`

5.1.3 Power Supply Variables

The variables defined in this section contain information about the Power Supplies in a CompactPCI chassis. This information is provided in the form of an SNMP table. Each entry in this table provides information about a single Power Supply. Entries are indexed by a physical Power Supply number which is equal to site number.

Power Supply information variables have the following OID:
`<ROOT_OID>.34.1.<var>.<powersupplynum>`

Here **`<var>`** is the index of a particular variable in the table entry describing a particular Power Supply slot. The variable indices are defined in the table below. **`<powersupplynum>`** is the Physical Power Supply number.

Note: Since both PEMs and Power Supplies can now be used in ATCA shelves, this table is now obsolete for non-CompactPCI systems. Instead, two separate tables are available: the table "xTCA

PEM Variables" (see section 5.1.12) describes ATCA Power Entry Modules and the table "xTCA Power Supply Variables" (see section 5.1.13) describes ATCA Power Supplies.

The following variables are defined for each power supply slot:

Table 17 Basic Power Supply Variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
`powersupply-slot-number`	1	INTEGER	Read-only	Table entry index, equal to **`<powersupplynum>`**.
`powersupply-degrade`	2	INTEGER	Read-only	1 – if power supply is in the Degraded state, 0 – otherwise. This variable is available in 2.x systems only.
`powersupply-fail`	3	INTEGER	Read-only	1 – if power supply is in the Failed state, 0 – otherwise. This variable is available in 2.x systems only.
`powersupply-inhibit`	4	INTEGER	Read-write	1 – if power supply is in the Inhibited state, 0 – otherwise. Writing a value to this field inhibits the power supply (if value=1) or re-enables it (if value=0). This variable is available in 2.x systems only.
`powersupply-healthy`	5	INTEGER	Read-only	1 – if power supply is healthy, 0 – otherwise.
`powersupply-slave-address`	6	INTEGER	Read-only	The 8-bit Slave address of the IPM controller representing this Power supply on IPMB. If the unit in question is not currently installed, then return -1. If the unit in question is installed, but does not have an active/working IPM controller, return 32 (20h). If the unit is installed and has an IPM controller, return the IPM controller's slave

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
				address.
`powersupply-fru-device-id`	7	INTEGER	Read-only	The FRU Device ID of the power supply. If the unit in question is not currently installed, then return -1. If the unit in question is installed, but does not have an active/working IPM controller, returns the control FRU ID. The control FRU ID is used in conjunction with the BMC IPMI address (20h), and represents the power supply to the BMC so that the power supply can be managed via the IPMI interface. This is applicable to CompactPCI systems only.
`powersupply-fruinfo-product-area-present`	8	INTEGER	Read-only	1 – if the product area is present within the power supply FRU Information, 0 – otherwise.
`powersupply-fruinfo-product-manufacturer`	9	DisplayString	Read-only	Returns the product manufacturer from the power supply FRU Information, or “N/A”.
`powersupply-fruinfo-product-name`	10	DisplayString	Read-only	Returns the product name from the power supply FRU Information, or “N/A”.
`powersupply-fruinfo-product-part-model-number`	11	DisplayString	Read-only	Returns the product part model number from the power supply FRU Information, or “N/A”.
`powersupply-fruinfo-product-version-number`	12	DisplayString	Read-only	Returns the product version from the power supply FRU Information, or “N/A”.
`powersupply-fruinfo-product-serial-number`	13	DisplayString	Read-only	Returns the product serial number from the power supply FRU Information, or “N/A”.
`powersupply-fruinfo-board-`	14	INTEGER	Read-only	1 – if the board area is

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
area-present				present within the power supply FRU Information, 0 – otherwise.
powersupply-fruinfo-board-manufacturer	15	DisplayString	Read-only	Returns the board manufacturer from the power supply FRU Information, or “N/A”.
powersupply-fruinfo-board-product-name	16	DisplayString	Read-only	Returns the board product name from the power supply FRU Information, or “N/A”.
powersupply-fruinfo-board-serial-number	17	DisplayString	Read-only	Returns the board serial number from the power supply FRU Information, or “N/A”.
powersupply-fruinfo-board-part-number	18	DisplayString	Read-only	Returns the board part number from the power supply FRU Information, or “N/A”.
powersupply-fruinfo-board-manufacture-time	19	INTEGER	Read-only	Returns the board manufacturing time: the number of seconds since 00:00:00, January 1, 1970, Coordinated Universal Time (UTC); -1 if the corresponding field is not present in the power supply FRU information.
powersupply-fruinfo-product-asset-tag	20	DisplayString	Read-only	Returns the product asset tag from the power supply FRU Information, or “N/A”.
powersupply-fruinfo-product-fru-file-id	21	DisplayString	Read-only	Returns the product FRU File ID from the power supply FRU Information, or “N/A”.
powersupply-fruinfo-board-fru-file-id	22	DisplayString	Read-only	Returns the board FRU File ID from the power supply FRU Information, or “N/A”.

For example, to check if the product area information is present for the Power Supply # 3, use the following OID:

<ROOT_OID>.34.1.2.8

5.1.4 Shelf Manager Variables

The variables defined in this section contain information about the Shelf Managers in the system. This information is provided in the form of an SNMP table. Each entry in this table provides information about a single Shelf Manager. Entries are indexed by a physical Shelf Manager number, which is equal to the site number.

Shelf Manager information variables have the following OID:
`<ROOT_OID>.35.1.<var>.<shelfmanagernum>`

Here **`<var>`** is the index of a particular variable in the table entry describing a particular Shelf manager slot. The variable indices are defined in the table below. **`<shelfmanagernum>`** is the Physical Shelf Manager number.

Table 18 Basic Shelf Manager Variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
`shelf-manager-slot-number`	1	INTEGER	Read-only	Table entry index, equal to **`<shelfmanagernum>`**.
`shelf-manager-ipmc-slave-address`	2	INTEGER	Read-only	The 8-bit Slave address of the IPM controller representing this Shelf Manager on IPMB.
`shelf-manager-present`	3	INTEGER	Read-only	1 – if Shelf Manager is present in the slot, 0 – otherwise.
`shelf-manager-healthy`	4	INTEGER	Read-only	1 – if Shelf Manager is healthy, 0 – otherwise.
`shelf-manager-active`	5	INTEGER	Read-write	1 – if Shelf Manager is active, 0 – otherwise. Writing 0 to this field triggers a reboot of the Shelf Manager, causing a switchover to the other Shelf Manager .
`shelf-manager-reset`	6	INTEGER	Read-write	1 – if Shelf Manager is in the reset state, 0 – otherwise. Writing 1 to this field triggers a reset of the target Shelf Manager if the other Shelf Manager is present (works similar to the IPMI “Cold Reset” command).

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
shelf-manager-fruinfo-product-area-present	7	INTEGER	Read-only	1 – if the product area is present within the Shelf Manager FRU Information, 0 – otherwise.
shelf-manager-fruinfo-product-manufacturer	8	DisplayString	Read-only	Returns the product manufacturer from the Shelf Manager FRU Information, or "N/A".
shelf-manager-fruinfo-product-name	9	DisplayString	Read-only	Returns the product name from the Shelf Manager FRU Information, or "N/A".
shelf-manager-fruinfo-product-part-model-number	10	DisplayString	Read-only	Returns the product part model number from the Shelf Manager FRU Information, or "N/A".
shelf-manager-fruinfo-product-version-number	11	DisplayString	Read-only	Returns the product version from the Shelf Manager FRU Information, or "N/A".
shelf-manager-fruinfo-product-serial-number	12	DisplayString	Read-only	Returns the product serial number from the Shelf Manager FRU Information, or "N/A".
shelf-manager-fruinfo-board-area-present	13	INTEGER	Read-only	1 – if the board area is present within the Shelf Manager FRU Information, 0 – otherwise.
shelf-manager-fruinfo-board-manufacturer	14	DisplayString	Read-only	Returns the board manufacturer from the Shelf Manager FRU Information, or "N/A".
shelf-manager-fruinfo-board-product-name	15	DisplayString	Read-only	Returns the board product name from the Shelf Manager FRU Information, or "N/A".
shelf-manager-fruinfo-board-serial-number	16	DisplayString	Read-only	Returns the board serial number from the Shelf Manager FRU Information, or "N/A".
shelf-manager-fruinfo-board-part-number	17	DisplayString	Read-only	Returns the board part number from the Shelf Manager FRU Information, or "N/A".

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
`shelf-manager-fruinfo-board-manufacture-time`	18	INTEGER	Read-only	Returns the board manufacturing time: the number of seconds since 00:00:00, January 1, 1970, Coordinated Universal Time (UTC); -1 if the corresponding field is not present in the Shelf Manager FRU information.
`shelf-manager-fruinfo-product-asset-tag`	19	DisplayString	Read-only	Returns the product asset tag from the Shelf Manager FRU Information, or "N/A".
`shelf-manager-fruinfo-product-fru-file-id`	20	DisplayString	Read-only	Returns the product FRU File ID from the Shelf Manager FRU Information, or "N/A".
`shelf-manager-fruinfo-board-fru-file-id`	21	DisplayString	Read-only	Returns the board FRU File ID from the Shelf Manager FRU Information, or "N/A".

For example, to check the slave address of the Shelf Manager # 2, use the following OID:
`<ROOT_OID>.35.1.2.2`

5.1.5 Chassis Variables

The variables defined in this section contain information about the Chassis (Shelf). This information is provided in the form of an SNMP branch. Each entry in this table provides information about a single Chassis. Entries are indexed by a physical Chassis number.

Chassis information variables have the following OID:
`<ROOT_OID>.36.<var>`

Here **`<var>`** is the index of a particular variable in the table entry.

Table 19 Basic Chassis Variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
`chassis-id`	1	DisplayString	Read-write	Read/ Write Shelf Address.
`chassis-type`	2	INTEGER	Read-only	The 8-bit Chassis Type from the Shelf FRU Information.
`chassis-part-number`	3	DisplayString	Read-only	Chassis Part Number from the Shelf FRU Information.
`chassis-serial-number`	4	DisplayString	Read-only	Chassis Serial Number from the Shelf FRU Information.

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
chassis-product-area-present	5	INTEGER	Read-only	1 – if the product area is present within the Shelf FRU Information, 0 – otherwise.
chassis-product-manufacturer	6	DisplayString	Read-only	Returns the product manufacturer from the Shelf FRU Information or "N/A".
chassis-product-name	7	DisplayString	Read-only	Returns the product name from the Shelf FRU Information or "N/A".
chassis-product-part-model-number	8	DisplayString	Read-only	Returns the product part model number from the Shelf FRU Information or "N/A".
chassis-product-version-number	9	DisplayString	Read-only	Returns the product version from the Shelf FRU Information or "N/A".
chassis-product-serial-number	10	DisplayString	Read-only	Returns the product serial number from the Shelf FRU Information or "N/A".
chassis-board-area-present	11	INTEGER	Read-only	1 – if the board area is present within the Shelf FRU Information, 0 – otherwise.
chassis-board-manufacturer	12	DisplayString	Read-only	Returns the board manufacturer from the Shelf FRU Information or "N/A".
chassis-board-product-name	13	DisplayString	Read-only	Returns the board product name from the Shelf FRU Information or "N/A".
chassis-board-serial-number	14	DisplayString	Read-only	Returns the board serial number from the Shelf FRU Information or "N/A".
chassis-board-part-number	15	DisplayString	Read-only	Returns the board part number from the Shelf FRU Information or "N/A".
chassis-board-manufacture-time	16	INTEGER	Read-only	Returns the board manufacturing time: the number of seconds since 00:00:00, January 1, 1970, Coordinated Universal Time (UTC); -1 if the corresponding field is not present in the Shelf FRU information.

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
chassis-product-asset-tag	17	DisplayString	Read-only	Returns the product asset tag from the Shelf FRU Information, or "N/A".
chassis-product-fru-file-id	18	DisplayString	Read-only	Returns the product FRU File ID from the Shelf FRU Information, or "N/A".
chassis-board-fru-file-id	19	DisplayString	Read-only	Returns the board FRU File ID from the Shelf FRU Information, or "N/A".

For example, to check the chassis type use the following OID:
<ROOT_OID>.36.2.0

5.1.6 Event Variables

The variables defined in this section contain information about the SEL entries in the system. This information is provided in the form of an SNMP table. Each entry in this table provides information about a single SEL entry.

SEL entry information variables have the following OID:
<ROOT_OID>.37.1.<var>.<selentrynum>

Here **<var>** is the index of a particular variable in the table entry describing a particular SEL entry. The variable indices are defined in the table below. **<selentrynum>** is the sel entry number.

Table 20 Basic Event Variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
event-index	1	INTEGER	Read-only	Table entry index, equal to **<selentrynum>**.
event-delete	2	INTEGER	Read-write	Returns 0 on reading, Writing 1 causes the current SEL entry to be deleted.
event-timestamp	3	INTEGER	Read-only	Timestamp of the SEL entry
event-class	4	INTEGER	Read-only	Returns sensor type value of the event. The following sensor types are recognized in the MIB: other (0), temperature (1), voltage (2), current (3), fan (4), HotSwap ('F0'H), PowerState ('E1'H)

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
`event-type`	5	INTEGER	Read-only	Event type other (0), aboveUpperNonRecoverable(1), aboveUpperCritical (2), aboveUpperNonCritical (3), belowLowerNonRecoverable(4), belowLowerCritical (5), belowLowerNonCritical (6), inserted (7), activated (8), communicationLost (9), communicationRestored (10), deactivated (11), extracted (12), powerDegrade (13), powerFail (14), powerInhibit (15).
`event-asserted`	6	INTEGER	Read-only	Event assertion state: deasserted (0), asserted (1).
`event-origin-site-type`	7	INTEGER	Read-only	Origin site type.
`event-origin-site-number`	8	INTEGER	Read-only	Origin site number.
`event-origins-slave-address`	9	INTEGER	Read-only	Origin IPMB address.
`event-origin-fru-id`	10	INTEGER	Read-only	Origin FRU Device ID.
`event-origin-sensor-number`	11	INTEGER	Read-only	Origin sensor number.

For example, to check the timestamp of the Record ID 10 in the SEL (which may or may not exist on an actual shelf at a given point in time), use the following OID:
`<ROOT_OID>.37.1.3.10`

The order in which the SEL entries are returned corresponds to the Record ID order in the SEL. This ensures that the index of each of the arrays (represented by **`event-index`**) increases monotonically for each “Get Next” operation. The resulting order of SEL entries may not correspond to the order in which corresponding events have been placed into the SEL; the **`event-timestamp`** variable can be used to reconstruct the order in which the events have been placed into the SEL.

5.1.7 Shelf Manager Status Variables

The variables defined in this section contain information about the Shelf Manager status.

Shelf Manager Status variables have the following OID:

<ROOT_OID>.38.<var>

Here **<var>** is the index of a particular variable in the table entry.

Table 21 Basic Shelf Manager Status Variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
`rmcp-interface-status`	1	INTEGER	Read-only	0 – RMCP interface is down, 1 – RMCP interface is up.
`shelf-fru-found-status`	2	INTEGER	Read-only	0 – Shelf FRU is not found, 1- Shelf FRU is found.
`active-status`	3	INTEGER	Read-only	0 - the current ShMM is Backup, 1 – the current ShMM is Active.

For example, to check the status of RMCP interface, use the following OID:
<ROOT_OID>.38.1.0

5.1.8 Shelf Manager Version Variables

The variables defined in this section contain information about the Shelf Manager version.

Shelf Manager Version variables have the following OID:
<ROOT_OID>.39.<var>

Here **<var>** is the index of a particular variable in the table entry.

Table 22 Basic Shelf Manager Version Variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
`major-version`	1	INTEGER	Read-only	Returns Shelf Manager major version.
`minor-version`	2	INTEGER	Read-only	Returns Shelf Manager minor version.
`carrier-type`	3	DisplayString	Read-only	Returns carrier type.
`carrier-subtype`	4	INTEGER	Read-only	Returns carrier subtype.
`carrier-subversion`	5	INTEGER	Read-only	Returns carrier subversion.
`functional-level`	6	INTEGER	Read-only	Returns 0 for the standard Shelf Manager; 1 for the entry level Shelf Manager.

For example, to get the Shelf Manager Major version, use the following OID:
<ROOT_OID>.39.1.0

5.1.9 TELCO Alarm Variables

The variables defined in this section contain information about the state of the TELCO alarms.

TELCO alarm variables have the following OID:
<ROOT_OID>.40.<var>

Here **<var>** is the index of a particular variable in the table entry.

Table 23 Basic TELCO Alarm Variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
minor-alarm	1	INTEGER	Read-write	Reflects the state of the Minor Alarm: 1 means "set", 0 means "cleared". Writing 1 to this variable sets the Minor Alarm; writing 0 to this variable clears the Minor Alarm.
major-alarm	2	INTEGER	Read-write	Reflects the state of the Major Alarm: 1 means "set", 0 means "cleared". Writing 1 to this variable sets the Major Alarm; writing 0 to this variable clears the Major Alarm.
critical-alarm	3	INTEGER	Read-write	Reflects the state of the Critical Alarm: 1 means "set", 0 means "cleared". Writing 1 to this variable sets the Critical Alarm; writing 0 to this variable clears the Critical Alarm.
alarm-cutoff	4	INTEGER	Read-only	Reflects the state of the Alarm Cutoff: 1 means "set", 0 means "cleared".

For example, to get the state of the Minor Alarm, use the following OID:
<ROOT_OID>.40.1.0

5.1.10 SEL Variables

The variables defined in this section contain information about the SEL state.

SEL variables have the following OID:
<ROOT_OID>.41.<var>

Here **<var>** is the index of a particular variable in the table entry.

Table 24 Basic SEL Variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
sel-version	1	INTEGER	Read-only	Returns System Event Log version byte; for example, 51h

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
				for IPMI 1.5.
`sel-entry-count`	2	INTEGER	Read-only	Returns the current number of SEL entries.
`sel-capacity`	3	INTEGER	Read-only	Returns capacity of the SEL in event records.
`sel-add-timestamp`	4	INTEGER	Read-only	Returns the Timestamp for the most recent SEL entry addition in seconds since 1/1/1970.
`sel-del-timestamp`	5	INTEGER	Read-only	Returns Timestamp for the most recent SEL entry deletion in seconds since 1/1/1970.
`sel-overflow-state`	6	INTEGER	Read-only	Returns the SEL Overflow status flag. 1 means "set", 0 means "not set".

For example, to get the current number of SEL entries, use the following OID:
`<ROOT_OID>.41.2.0`

5.1.11 Carrier-specific Variables

Some carriers may have carrier/vendor specific SNMP extensions with the following root OID:
`<ROOT_OID>.42.<var>`,
where **`<var>`** depends on a carrier type.

5.1.12 xTCA PEM Variables

The variables defined in this section contain information about power entry modules in xTCA shelves (as differentiated from CompactPCI or PICMG 2.x chassis).

xTCA PEM variables have the following OID:
`<ROOT_OID>.43.<var>`

Here **`<var>`** is the index of a particular variable in the table entry.

Table 25 Basic xTCA PEM Variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
`xtca-pem-slot-number`	1	INTEGER	Read-only	Table entry index, that is equal to the PEM site number.
`xtca-pem-present`	2	INTEGER	Read-only	Returns 1 if the PEM is present, 0 otherwise.
`xtca-pem-healthy`	3	INTEGER	Read-only	Returns 0 if the PEM is unhealthy (i.e. it is in M1, M7 or the latest State Change Cause

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
				for this PEM was unexpected), 1 otherwise.
xtca-pem-slave-address	4	INTEGER	Read-only	Returns the 8-bit slave address of the IPM controller representing this PEM on IPMB.
xtca-pem-fru-device-id	5	INTEGER	Read-only	Returns the FRU Device ID of the PEM.
xtca-pem-fruinfo-product-area-present	6	INTEGER	Read-only	Returns 1 if the product area is present within the PEM FRU Information, 0 otherwise
xtca-pem-fruinfo-product-manufacturer	7	DisplayString	Read-only	Returns the product manufacturer string from the PEM FRU Information, or “N/A” if it’s not present.
xtca-pem-fruinfo-product-name	8	DisplayString	Read-only	Returns the product name from the PEM FRU Information, or “N/A” if it’s not present.
xtca-pem-fruinfo-product-part-model-number	9	DisplayString	Read-only	Returns the product part model number from the PEM FRU Information, or “N/A” if it’s not present.
xtca-pem-fruinfo-product-version-number	10	DisplayString	Read-only	Returns the product version number from the PEM FRU Information, as a text string, or “N/A” if it’s not present.
xtca-pem-fruinfo-product-serial-number	11	DisplayString	Read-only	Returns the product serial number from the PEM FRU Information, as a text string, or “N/A” if it’s not present.
xtca-pem-fruinfo-board-area-present	12	INTEGER	Read-only	Returns 1 if the board area is present within the PEM FRU Information, 0 otherwise
xtca-pem-fruinfo-board-manufacturer	13	DisplayString	Read-only	Returns the board manufacturer string from the PEM FRU Information, or “N/A” if it’s not present.
xtca-pem-fruinfo-board-product-name	14	DisplayString	Read-only	Returns the board product name from the PEM FRU Information, or “N/A” if it’s not present.
xtca-pem-fruinfo-board-serial-number	15	DisplayString	Read-only	Returns the board serial number from the PEM FRU Information, as a text string, or “N/A” if it’s not present.
xtca-pem-	16	DisplayString	Read-	Returns the board part number

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
fruinfo-board-part-number			only	from the PEM FRU Information, or “N/A” if it’s not present.
xtca-pem-fruinfo-board-manufacture-time	17	INTEGER	Read-only	Returns the board manufacturing time: the number of seconds since 00:00:00, January 1, 1970, (UTC); -1 if the corresponding field is not present in the PEM FRU information.
xtca-pem-fruinfo-product-asset-tag	18	DisplayString	Read-only	Returns the product asset tag from the PEM FRU Information, or “N/A”.
xtca-pem-fruinfo-product-fru-file-id	19	DisplayString	Read-only	Returns the product FRU File ID from the PEM FRU Information, or “N/A”.
xtca-pem-fruinfo-board-fru-file-id	20	DisplayString	Read-only	Returns the board FRU File ID from the PEM FRU Information, or “N/A”.

For example, to check if the product area information is present for the PEM # 2, use the following OID:
<ROOT_OID>.43.1.6.2

5.1.13 xTCA Power Supply Variables

The variables defined in this section contain information about AC power supplies in in xTCA shelves (as differentiated from CompactPCI or PICMG 2.x chassis).

xTCA power supply variables have the following OID:
<ROOT_OID>.44.<var>

Here **<var>** is the index of a particular variable in the table entry.

Table 26 Basic xTCA Power Supply Variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
xtca-ps-slot-number	1	INTEGER	Read-only	Table entry index, that is equal to the power supply site number.
xtca-ps-present	2	INTEGER	Read-only	Returns 1 if the power supply is present, 0 otherwise.
xtca-ps-healthy	3	INTEGER	Read-only	Returns 0 if the power supply is unhealthy (i.e. it is in M1, M7 or the latest State Change Cause for this power supply was

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
				unexpected), 1 otherwise.
`xtca-ps-slave-address`	4	INTEGER	Read-only	Returns the 8-bit slave address of the IPM controller representing this power supply on IPMB.
`xtca-ps-fru-device-id`	5	INTEGER	Read-only	Returns the FRU Device ID of the power supply.
`xtca-ps-fruinfo-product-area-present`	6	INTEGER	Read-only	Returns 1 if the product area is present within the power supply FRU Information, 0 otherwise
`xtca-ps-fruinfo-product-manufacturer`	7	DisplayString	Read-only	Returns the product manufacturer string from the power supply FRU Information, or “N/A” if it’s not present.
`xtca-ps-fruinfo-product-name`	8	DisplayString	Read-only	Returns the product name from the power supply FRU Information, or “N/A” if it’s not present.
`xtca-ps-fruinfo-product-part-model-number`	9	DisplayString	Read-only	Returns the product part model number from the power supply FRU Information, or “N/A” if it’s not present.
`xtca-ps-fruinfo-product-version-number`	10	DisplayString	Read-only	Returns the product version number from the power supply FRU Information, as a text string, or “N/A” if it’s not present.
`xtca-ps-fruinfo-product-serial-number`	11	DisplayString	Read-only	Returns the product serial number from the power supply FRU Information, as a text string, or “N/A” if it’s not present.
`xtca-ps-fruinfo-board-area-present`	12	INTEGER	Read-only	Returns 1 if the board area is present within the power supply FRU Information, 0 otherwise
`xtca-ps-fruinfo-board-manufacturer`	13	DisplayString	Read-only	Returns the board manufacturer string from the power supply FRU Information, or “N/A” if it’s not present.
`xtca-ps-fruinfo-board-product-name`	14	DisplayString	Read-only	Returns the board product name from the power supply FRU Information, or “N/A” if it’s not present.
`xtca-ps-fruinfo-board-serial-number`	15	DisplayString	Read-only	Returns the board serial number from the power supply FRU Information, as a text string, or

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
				"N/A" if it's not present.
`xtca-ps-fruinfo-board-part-number`	16	DisplayString	Read-only	Returns the board part number from the power supply FRU Information, or "N/A" if it's not present.
`xtca-ps-fruinfo-board-manufacture-time`	17	INTEGER	Read-only	Returns the board manufacturing time: the number of seconds since 00:00:00, January 1, 1970, (UTC); -1 if the corresponding field is not present in the power supply FRU information.
`xtca-ps-fruinfo-product-asset-tag`	18	DisplayString	Read-only	Returns the product asset tag from the power supply FRU Information, or "N/A".
`xtca-ps-fruinfo-product-fru-file-id`	19	DisplayString	Read-only	Returns the product FRU File ID from the power supply FRU Information, or "N/A".
`xtca-ps-fruinfo-board-fru-file-id`	20	DisplayString	Read-only	Returns the board FRU File ID from the power supply FRU Information, or "N/A".

For example, to check if the board area information is present for the power supply # 3, use the following OID:

```
<ROOT_OID>.44.1.12.3
```

5.2 Advanced MIB Variables

5.2.1 IPM Controller Variables

The variables defined in this section contain information about the IPM controllers in the shelf. This information is provided in the form of an SNMP table. Each entry in this table provides information about a single IPM controller. Entries are indexed by an 8-bit address of the IPM Controller on the IPMB.

IPM controller information variables have the following OID:
<ROOT_OID>.1.1.<var>.<addr>

Here **`<var>`** is the index of a particular variable in the table entry describing a particular IPM controller. The variable indices are defined in the table below. **`<addr>`** is the 8-bit IPMB address of the IPM controller.

Table 27 Advanced IPM Controller Variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
`ipm-controller-index`	1	INTEGER	Read-only	Table entry index, equal to **`<addr>`**.
`ipm-controller-sdr-version`	2	INTEGER	Read-only	SDR Version of the Management Controller Device Locator Record for this controller.
`ipm-controller-picmg-version`	3	INTEGER	Read-only	PICMG Extension Version as reported by the controller in a “Get PICMG Properties” reply.
`ipm-controller-slave-address`	4	INTEGER	Read-only	Device Slave Address as defined in the Management Controller Device Locator Record for this controller.
`ipm-controller-channel-number`	5	INTEGER	Read-only	Channel Number as defined in the Management Controller Device Locator Record for this controller
`ipm-controller-power-state-notification`	6	INTEGER	Read-only	Power State Notification as defined in the Management Controller Device Locator Record for this controller.
`ipm-controller-global-initialization`	7	INTEGER	Read-only	Global Initialization as defined in the Management Controller Device Locator Record for this controller.

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
ipm-controller-capabilities	8	INTEGER	Read-only	Device Capabilities as defined in the Management Controller Device Locator Record for this controller.
ipm-controller-id-string	9	DisplayString (SIZE(0..255))	Read-only	Device ID String as defined in the Management Controller Device Locator Record for this controller.
ipm-controller-maximum-fru	10	INTEGER	Read-only	Max FRU Device ID as reported by the controller in "Get PICMG Properties" reply.
ipm-controller-own-fru-id	11	INTEGER	Read-only	FRU Device ID for IPM Controller as reported by the controller in "Get PICMG Properties" reply.

For example, to get the Device ID String of the IPM Controller at IPMB address 20h = 32_{10} (that is, the Shelf Manager itself), use the following OID:
<ROOT_OID>.1.1.9.32

5.2.2 FRU Device Variables

The variables defined in this section contain information about the FRU devices in the shelf. This information is provided in the form of an SNMP table. Each entry in this table provides information about a single FRU. The table lists all FRUs for which FRU Device Locator Records (SDR Type 11h) or Management Controller Device Locator Records (SDR Type 12h) are present in the SDR Repository.

FRU device information variables have the following OID:
<ROOT_OID>.2.1.<var>.<ipmb_addr>.<fru_id>

Here **<var>** is the index of a particular variable in the table entry describing a particular FRU device. The variable indices are defined in the table below. **<ipmb_addr>** is the IPMB address of IPM controller and **<fru_id>** is the number of the FRU device on this IPM controller.

Table 28 Advanced FRU Variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
fru-device-index	1	INTEGER	Read-only	Table entry index, equal to ((**<ipmb_addr>** << 16) |**<fru_id>**).

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
`fru-device-sdr-version`	2	INTEGER	Read-only	SDR Version of the FRU Device or Management Controller Device Locator Record for this FRU. If the record is absent, this field is read as -1.
`fru-device-slave-address`	3	INTEGER	Read-only	Device Slave Address as defined in the FRU Device or Management Controller Device Locator Record for this FRU
`fru-device-fru-device-id`	4	INTEGER	Read-only	FRU Device ID as defined in the FRU Device Locator Record for this FRU, or 0 for Management Controller devices
`fru-device-channel-number`	5	INTEGER	Read-only	Channel Number as defined in the FRU Device or Management Controller Device Locator Record for this FRU or -1 if the record is absent.
`fru-device-device-type`	6	INTEGER	Read-only	For FRUs with FRU Device ID different from zero: Device Type as defined in the FRU Device Locator Record for this FRU. Since the Management Controller Device Locator Record doesn't provide the Device Type information, for Management Controller devices (FRU #0), this field is set to FRU Inventory Device (10h). If the record is absent, this field is read as -1.
`fru-device-device-type-modifier`	7	INTEGER	Read-only	For FRUs with FRU Device ID different from zero: Device Type Modifier as defined in the FRU Device Locator Record for this FRU. Since the Management Controller Device Locator Record doesn't provide the Device Type information, for Management Controller devices (FRU #0), this field is set to Unspecified (FFh). If the record is absent, this field is read as -1.

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
`fru-device-fru-entity-id`	8	INTEGER	Read-only	(FRU) Entity ID as defined in the FRU Device or Management Controller Device Locator Record for this FRU. If the record is absent, this field is read as -1.
`fru-device-fru-entity-instance`	9	INTEGER	Read-only	(FRU) Entity Instance as defined in the FRU Device or Management Controller Device Locator Record for this FRU. If the record is absent, this field is read as -1.
`fru-device-id-string`	10	DisplayString (SIZE(0..255))	Read-only	Device ID String as defined in the FRU Device or Management Controller Device Locator Record for this FRU. If the record is absent, this field is read as “N/A”.
`fru-device-hot-swap-state`	11	INTEGER	Read-only	Current PICMG 3.0 FRU state (M1...M7) for this FRU. If this variable is equal to n, that means that the FRU is in state Mn.
`fru-device-activated`	12	INTEGER	Read-write	When reading: 1 means that the FRU device is active (that is, in state M4), 0 is returned otherwise. Writing 1 to this variable triggers sending the “Set FRU Activation (Activate FRU)” command to this FRU, if the FRU is in state M2 or M5, and sending “Set FRU Activation Policy (Clear Locked)” command if the FRU is in state M1. Writing 0 to this variable triggers sending the “Set FRU Activation (Deactivate FRU)” command to this FRU, if the FRU is in state M2, M3, M4, or M5, and sending “Set FRU Activation Policy (Set Locked)” command if the FRU is in state M1 or M6.

For example, to get the Device ID String of the FRU 0 of IPM controller at IPMB address 20h = 32_{10} (Shelf Manager), use the following OID:
<ROOT_OID>.2.1.10.32.0

5.2.3 Sensor Variables

The variables defined in this section contain information about the sensors in the shelf. This information is provided in the form of an SNMP table. Each entry in this table provides information about a single sensor. The table lists all sensors for which Full Sensor Records (SDR Type 01h), Compact Sensor Records (SDR Type 02h) or Event-Only Sensor Records (SDR Type 03h) exist in the shelf.

Sensor variables have the following OID:
<ROOT_OID>.3.1.<var>.<ipmb_addr>.<seqnum>

Here **<var>** is the index of a particular variable in the table entry describing a particular sensor. The variable indices are defined in the table below. **<ipmb_addr>**.**<seqnum>** is a compound index where **<ipmb_addr>** is the IPMB address of an IPM controller and **<seqnum>** is the sequential number of the sensor on this IPM controller. This sequential number is not necessarily equal to the sensor number, if the target IPM controller defines sensors on multiple LUNs.

Table 29 Advanced Sensor Variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
sensor-index	1	INTEGER	Read-only	Table entry index, equal to ((**<ipmb_addr>** << 16) |**<seqnum>**).
sensor-sdr-version	2	INTEGER	Read-only	SDR Version of the Sensor Record.
sensor-record-type	3	INTEGER	Read-only	Record Type of the Sensor Record: 01h – for Full Sensor Records, 02h – for Compact Sensor Records, 03h – for Event-Only Sensor Records.
sensor-owner-id	4	INTEGER	Read-only	Sensor Owner ID as defined in the Sensor Record.
sensor-owner-lun	5	INTEGER	Read-only	Sensor Owner LUN as defined in the Sensor Record.
sensor-number	6	INTEGER	Read-only	Sensor Number as defined in the Sensor Record.
sensor-entity-instance	7	INTEGER	Read-only	Entity Instance as defined in the Sensor Record.
sensor-entity-id	8	INTEGER	Read-only	Entity ID as defined in the Sensor Record.

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
sensor-initialization	9	INTEGER	Read-only	Sensor Initialization as defined in the Sensor Record. Read as 0 for Event-Only Sensor Records.
sensor-capabilities	10	INTEGER	Read-only	Sensor Capabilities as defined in the Sensor Record. Read as 0 for Event-Only Sensor Records.
sensor-type	11	INTEGER	Read-only	Sensor Type as defined in the Sensor Record.
sensor-event	12	INTEGER	Read-only	Event/Reading Type Code as defined in the Sensor Record.
sensor-assertion-event-mask	13	INTEGER	Read-only	Assertion Event Mask / Lower Threshold Reading Mask as defined in the Sensor Record. Read as 0 for Event-Only Sensor Records
sensor-deassertion-event-mask	14	INTEGER	Read-only	Deassertion Event Mask / Upper Threshold Reading Mask as defined in the Sensor Record. Read as 0 for Event-Only Sensor Records
sensor-mask	15	INTEGER	Read-only	Discrete Reading Mask / Settable Threshold Mask, Readable Threshold Mask as defined in the Sensor Record. Read as 0 for Event-Only Sensor Records
sensor-unit1	16	INTEGER	Read-only	Sensor Units 1 as defined in the Sensor Record. Read as 0 for Event-Only Sensor Records
sensor-unit2	17	INTEGER	Read-only	Sensor Units 2 – Base Unit as defined in the .Sensor Record. Read as 0 for Event-Only Sensor Records
sensor-unit3	18	INTEGER	Read-only	Sensor Units 3 – Modifier Unit as defined in the Sensor Record. Read as 0 for Event-Only Sensor Records

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
`sensor-linearization`	19	INTEGER	Read-only	Linearization as defined in the Sensor Record. Valid only for Full Sensor Records. Read as 0 for Compact Sensor Records and Event-Only Sensor Records.
`sensor-M`	20	INTEGER	Read-only	M sensor reading conversion parameter as defined in the Sensor Record. Valid only for Full Sensor Records. Read as 1 for Compact Sensor Records. Read as 0 for Event-Only Sensor Records.
`sensor-tolerance`	21	INTEGER	Read-only	Tolerance sensor reading conversion parameter as defined in the Sensor Record. Valid only for Full Sensor Records. Read as 0 for Compact Sensor Records and Event-Only Sensor Records.
`sensor-B`	22	INTEGER	Read-only	B sensor reading conversion parameter as defined in the Sensor Record. Valid only for Full Sensor Records. Read as 0 for Compact Sensor Records and Event-Only Sensor Records.
`sensor-accuracy`	23	INTEGER	Read-only	Accuracy sensor reading conversion parameter as defined in the Sensor Record. Valid only for Full Sensor Records. Read as 0 for Compact Sensor Records and Event-Only Sensor Records.
`sensor-accuracy-exp`	24	INTEGER	Read-only	Accuracy exp sensor reading conversion parameter as defined in the Sensor Record. Valid only for Full Sensor Records. Read as 0 for Compact Sensor Records and Event-Only Sensor Records.

Variable	Index	Type	Access Mode	Description
`sensor-R-exp`	25	INTEGER	Read-only	R exp sensor reading conversion parameter as defined in the Sensor Record. Valid only for Full Sensor Records. Read as 0 for Compact Sensor Records and Event-Only Sensor Records.
`sensor-B-exp`	26	INTEGER	Read-only	B exp sensor reading conversion parameter as defined in the Sensor Record. Valid only for Full Sensor Records. Read as 0 for Compact Sensor Records and Event-Only Sensor Records.
`sensor-characteristic-flags`	27	INTEGER	Read-only	Analog characteristic flags as defined in the Sensor Record. Valid only for Full Sensor Records. Read as 0 for Compact Sensor Records and Event-Only Sensor Records.
`sensor-reading`	28	INTEGER	Read-only	Current sensor reading in raw form. Read as 0 for Event-Only Sensor Records.
`sensor-processed-reading`	29	DisplayString (SIZE(0..255))	Read-only	Current sensor reading processed according to reading conversion formula for this sensor. For discrete sensors, the current state mask is returned. Read as a string “N/A” for Event-Only Sensor Records.
`sensor-nominal-reading`	30	INTEGER	Read-only	Nominal Reading as defined in the Sensor Record. Valid only for Full Sensor Records. Read as 0 for Compact Sensor Records and Event-Only Sensor Records.
`sensor-nominal-maximum`	31	INTEGER	Read-only	Normal Maximum as defined in the Sensor Record. Valid only for Full Sensor Records. Read as 0 for Compact Sensor Records and Event-Only Sensor Records.

Variable	Index	Type	Access Mode	Description
`sensor-nominal-minimum`	32	INTEGER	Read-only	Normal Minimum as defined in the Sensor Record. Valid only for Full Sensor Records. Read as 0 for Compact Sensor Records and Event-Only Sensor Records.
`sensor-maximum-reading`	33	INTEGER	Read-only	Sensor Maximum Reading as defined in the Sensor Record. Valid only for Full Sensor Records. Read as 0 for Compact Sensor Records and Event-Only Sensor Records.
`sensor-minimum-reading`	34	INTEGER	Read-only	Sensor Minimum Reading as defined in the Sensor Record. Valid only for Full Sensor Records. Read as 0 for Compact Sensor Records and Event-Only Sensor Records.
`sensor-upper-non-recoverable-threshold`	35	INTEGER	Read-write	Current value of the Upper non-recoverable Threshold for the specified sensor. For Event-only sensors, read as 0, writes are ignored.
`sensor-upper-critical-threshold`	36	INTEGER	Read-write	Current value of the Upper critical Threshold for the specified sensor. For Event-only sensors, read as 0, writes are ignored.
`sensor-upper-non-critical-threshold`	37	INTEGER	Read-write	Current value of the Upper non-critical Threshold for the specified sensor. For Event-only sensors, read as 0, writes are ignored.
`sensor-lower-non-recoverable-threshold`	38	INTEGER	Read-write	Current value of the Lower non-recoverable Threshold for the specified sensor. For Event-only sensors, read as 0, writes are ignored.
`sensor-lower-critical-threshold`	39	INTEGER	Read-write	Current value of the Lower critical Threshold for the specified sensor. For Event-only sensors, read as 0, writes are ignored.

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
`sensor-lower-non-critical-threshold`	40	INTEGER	Read-write	Current value of the Lower non-critical Threshold for the specified sensor. For Event-only sensors, read as 0, writes are ignored.
`sensor-positive-going-threshold-hysteresis`	41	INTEGER	Read-write	Current value of the Positive-going Threshold Hysteresis for the specified sensor. For Event-only sensors, read as 0, writes are ignored.
`sensor-negative-going-threshold-hysteresis`	42	INTEGER	Read-write	Current value of the Negative-going Threshold Hysteresis for the specified sensor. For Event-only sensors, read as 0, writes are ignored.
`sensor-id-string`	43	DisplayString (SIZE(0..255))	Read-only	ID String as defined in the Sensor Record.
`sensor-entire-sensor-data`	44	OCTET STRING (SIZE(0..128))	Read-only	Entire contents of the SDR: 48..64 bytes for Full Sensor Record, 32..48 bytes for Compact Sensor Record, 17..33 bytes for Event-Only Sensor Record.
`sensor-processed-unr-threshold`	45	DisplayString (SIZE(0..255))	Read-only	Processed current value of the Upper non-recoverable Threshold for the specified sensor. Read “N/A” when the sensor is not threshold-based. Read “N/A” when the sensor is threshold-based and has no such threshold.
`sensor-processed-uc-threshold`	46	DisplayString (SIZE(0..255))	Read-only	Processed current value of the Upper critical Threshold for the specified sensor. Read “N/A” when the sensor is not threshold-based. Read “N/A” when the sensor is threshold-based and has no such threshold.

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
`sensor-processed-unc-threshold`	47	DisplayString (SIZE(0..255))	Read-only	Processed current value of the Upper non-critical Threshold for the specified sensor. Read “N/A” when the sensor is not threshold-based. Read “N/A” when the sensor is threshold-based and has no such threshold.
`sensor-processed-lnr-threshold`	48	DisplayString (SIZE(0..255))	Read-only	Processed current value of the Lower non-recoverable Threshold for the specified sensor. Read “N/A” when the sensor is not threshold-based. Read “N/A” when the sensor is threshold-based and has no such threshold.
`sensor-processed-lc-threshold`	49	DisplayString (SIZE(0..255))	Read-only	Processed current value of the Lower critical Threshold for the specified sensor. Read “N/A” when the sensor is not threshold-based. Read “N/A” when the sensor is threshold-based and has no such threshold.
`sensor-processed-lnc-threshold`	50	DisplayString (SIZE(0..255))	Read-only	Processed current value of the Lower non-critical Threshold for the specified sensor. Read “N/A” when the sensor is not threshold-based. Read “N/A” when the sensor is threshold-based and has no such threshold.

For example, to get the ID String of the second sensor on the IPM controller at IPMB address 20h = 32_{10} (Shelf Manager), use the following OID:
`<ROOT_OID>.3.1.43.32.2`

5.2.4 Board Variables

The variables defined in this section contain information about the AdvancedTCA Board slots in the system. This information is provided in the form of an SNMP table. Each entry in this table provides information about a single board slot. Entries are indexed by a Physical Slot number.

The semantics of the board variables below are different between the ATCA context and the PICMG 2.x (CompactPCI) context. Therefore, the description for each of these variables essentially contains of the two parts, prefixed by “AdvancedTCA:” and “CompactPCI:” respectively.

Board information variables have the following OID:
`<ROOT_OID>.4.1.<var>.<slotnum>`

Here **`<var>`** is the index of a particular variable in the table entry describing a particular board slot. The variable indices are defined in the table below. **`<slotnum>`** is the Physical Slot number.

Table 30 Advanced Board Slot Variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
`board-index`	1	INTEGER	Read-only	Table entry index, equal to **`<slotnum>`**.
`board-present`	2	INTEGER	Read-only	Both ATCA and CompactPCI: 1 – if a board is present in the slot, 0 – otherwise.
`board-healthy`	3	INTEGER	Read-only	AdvancedTCA: 1 – if a board is present and healthy, 0 – if the board is either not present, or not healthy. Unhealthy board is a board in state M1 or M7. CompactPCI: This variable reflects the state of the HEALTHY# signal for the slot: 1 – the board is healthy, 0 – the board is not healthy.

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
board-reset	4	INTEGER	Read-write	AdvancedTCA: When reading: 1 – if the board is not present (in state M0), 0 – otherwise. Writing 1 to this variable triggers sending the "FRU Control (Cold Reset)" command to the IPM Controller of this board. Writing 0 to this variable is ignored. CompactPCI: When reading, reflects the reset state of the board: 1 – in reset, 0 – not in reset. Writing 1 to this variable triggers a reset of the board by pulsing the BD_SEL# signal for the slot; writing 0 to this variable is ignored.
board-slave-address	5	INTEGER	Read-only	Both ATCA and CompactPCI: The 8-bit Slave address of the IPM Controller representing this board on IPMB, according to the address table in the Shelf FRU Information.
board-fru-device-id	6	INTEGER	Read-only	Both ATCA and CompactPCI: The FRU Device ID for the board, according to the address table in the Shelf FRU Information.

For example, to check the presence of the board in slot 8, use the following OID:
<ROOT_OID>.4.1.2.8

5.2.5 FRU LED Variables

The variables defined in this section contain information about the FRU LEDs. This information is provided in the form of an SNMP table. Each entry in this table provides information about a single LED. The table lists all LEDs for all FRUs for which FRU Device Locator Records (SDR Type 11h) or Management Controller Device Locator Records (SDR Type 12h) are present in the SDR Repository.

FRU LED information variables have the following OID:
<ROOT_OID>.22.1.<var>.<ipmb_addr>.<fru_id>.<led_number>

Here **<var>** is the index of a particular variable in the table entry describing a particular FRU LED. The variable indices are defined in the table below. **<ipmb_addr>** is the IPMB address of IPM controller and **<fru_id>** is the number of the FRU device on this IPM controller. The **<led_number>** is the number of the LED as defined in the ATCA specification (0 = blue LED, 1 = out-of-service LED and so on).

Table 31 FRU LED Variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
led-index	1	INTEGER	Read-only	Table entry index, defined as follows: Bits 0-15: LED number Bits 16-23: FRU device ID Bits 24-31: IPMB address of the IPM controller
led-color-capabilities	2	INTEGER	Read-only	The bit mask of colors supported by the LED, defined as follows: [0] – reserved, set to 0 [1] – LED supports BLUE [2] – LED supports RED [3] – LED supports GREEN [4] – LED supports AMBER [5] – LED supports ORANGE [6] – LED supports WHITE
led-state-capabilities	3	INTEGER	Read-only	The bit mask of special LED flags, defined as follows: [0] – LED is powered from Payload power [1] – LED has another hardware restriction
led-default-local-color	4	INTEGER	Read-only	The default LED color in local control state, in the range 0 to 6, defined as follows: 0 – local control not supported 1 – BLUE 2 – RED 3 – GREEN 4 – AMBER 5 – ORANGE 6 – WHITE

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
`led-default-override-color`	5	INTEGER	Read-only	The default LED color in override state, in the range 1 to 6, defined as follows: 1 – BLUE 2 – RED 3 – GREEN 4 – AMBER 5 – ORANGE 6 – WHITE
`led-current-state-flags`	6	INTEGER	Read-only	The bit mask of current LED state flags, defined as follows: [0] – the LED has local control state [1] – the override state has been enabled [2] – the lamp test has been enabled [3] – LED has a hardware restriction that is not currently met
`led-local-state`	7	INTEGER	Read-only	The current LED local control state and color. Reported as 0 if the LED does not support local control state, otherwise defined as follows: Bits 0-7: local control LED function (byte 4 of the “Get FRU LED State” command response) Bits 8-15: local control on-duration (byte 5 of the “Get FRU LED State” command response) Bits 16-24: local control color (byte 6 of the “Get FRU LED State” command response)

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
led-override-state	8	INTEGER	Read-write	The current LED local control state and color. On read, reported as 0 if the command response to "Get FRU LED State" does not include bytes 7 to 9. Otherwise, and for write access, the value is defined as follows: Bits 0-7: override state LED function (byte 7 of the "Get FRU LED State" command response, byte 4 of the "Set FRU LED State" command request) Bits 8-15: override state on-duration (byte 8 of the "Get FRU LED State" command response, byte 5 of the "Set FRU LED State" command request) Bits 16-24: override state color (byte 9 of the "Get FRU LED State" command response, byte 6 of the "Set FRU LED State" command request). On write, return to local control state can be requested by placing FCh in bits 0-7; lamp test can be requested by placing FBh in bits 0-7 and lamp test duration, in hundreds of milliseconds, in bits 8-15.
led-lamp-test-duration	9	INTEGER	Read-only	Current lamp test duration, in hundreds of milliseconds. Reported as 0 if the LED is not in a lamp test state.

For example, to get the current state flags of the LED 1 on FRU 0 of IPM controller at IPMB address 82h = 130_{10} (the first hub board), use the following OID:
<ROOT_OID>.22.1.5.130.0.1

5.2.6 System Event Log Variables

The variables defined in this section contain information about the System Event Log (SEL). This information is provided in the form of an SNMP table. Each entry in this table provides information about a single System Event Log record. Table entries are indexed by a SEL Record ID.

SEL information variables have the following OID:
<ROOT_OID>.5.1.<var>.<recid>

Here **<var>** is the index of a particular variable in the table entry describing a particular SEL record. The variable indices are defined in the table below. **<recid>** is the 16-bit SEL Record ID: 1...FFFEh.

Table 32 Advanced System Log Variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
sel-index	1	INTEGER	Read-only	Table entry index, equal to **<recid>**.
sel-contents	2	OCTET STRING (SIZE(0..128))	Read-only	Contents of the SEL entry.

For example, to get the contents of the SEL entry with Record ID 3001, use the following OID:
<ROOT_OID>.5.1.2.3001

5.2.7 Shelf Variables

The variables defined in this section contain information about the shelf in general. This information is provided in the form of an SNMP table. Each entry in this table provides information about a single shelf. Table entries are indexed by shelf numbers. The current release of the Shelf Manager software supports only one shelf per Shelf Manager. The table index is intended to allow for future extensions, but for this revision it must be set to 1.

Shelf information variables have the following OID:
<ROOT_OID>.6.1.<var>.<shelfid>

Here **<var>** is the index of a particular variable in the table entry describing a particular shelf. The variable indices are defined in the table below. **<shelfid>** is the shelf number. For this release **<shelfid>** must be set to 1.

Table 33 Advanced Shelf Variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
shelf-index	1	INTEGER	Read-only	Table entry index, equal to **<shelfid>**
shelf-healthy	2	INTEGER	Read-only	1 – if the shelf doesn't have unhealthy components, 0 – if there exist unhealthy components in the shelf.

For example, to get the health status of the entire shelf, use the following OID:
<ROOT_OID>.6.1.2.1

5.2.8 LAN Configuration Variables

The variables defined in this section contain LAN configuration information. This information is provided in the form of an SNMP table. Each entry in this table provides information about a single configuration variable. Table entries are further indexed by IPMI channel numbers. The current release of the Shelf Manager supports two LAN channel numbers – IPMI channel #1 and IPMI channel #2. Channel #2 data are available only if two LAN channels are used by the Shelf Manager for RMCP access, and they are used in Active-Active mode.

Shelf information variables have the following OID:
<ROOT_OID>.7.1.<var>.<channel>

Here **`<var>`** is the index of a particular variable in the table entry describing a particular LAN channel configuration. The variable indices are defined in the table below. **`<channel>`** is the IPMI channel number. For this release, LAN channel number 1 is always supported and LAN channel number 2 is optionally supported. The current release also has a fixed number of supported destinations – 16. Thus the SNMP variables for the Destination Type and Destination Addresses parameters are implemented as fixed-sized arrays.

Table 34 Advanced LAN Channel Variables

Variable	Index	Type	Access Mode	Description
`lan-configuration-index`	1	INTEGER	Read-only	Table entry index, equal to **`<channel>`**.
`lan-configuration-set-in-progress`	2	INTEGER	Read-only	Set In Progress parameter for the LAN channel.
`lan-configuration-authentication-type-support`	3	INTEGER	Read-only	Authentication Type Support parameter for the LAN channel.
`lan-configuration-authentication-type-enable`	4	OCTET STRING (SIZE(5))	Read-write	Authentication Type Enables parameter for the LAN channel.
`lan-configuration-ip-address`	5	IpAddress	Read-write	IP Address parameter for the LAN channel.
`lan-configuration-ip-address-source`	6	INTEGER	Read-only	IP Address Source parameter for the LAN channel.
`lan-configuration-mac-address`	7	OCTET STRING (SIZE(6))	Read-write	MAC Address parameter for the LAN channel.
`lan-configuration-subnet-mask`	8	IpAddress	Read-write	Subnet Mask parameter for the LAN channel

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
`lan-configuration-ipv4-header-parameters`	9	OCTET STRING (SIZE(3))	Read-write	IPv4 Header Parameters parameter for the LAN channel.
`lan-configuration-primary-rmcp-port-number`	10	INTEGER	Read-write	Primary RMCP Port Number parameter for the LAN channel.
`lan-configuration-secondary-rmcp-port-number`	11	INTEGER	Read-write	Secondary RMCP Port Number parameter for the LAN channel.
`lan-configuration-bmc-generated-arp-control`	12	INTEGER	Read-write	BMC-generated ARP control parameter for the LAN channel.
`lan-configuration-gratuitous-arp-interval`	13	INTEGER	Read-write	Gratuitous ARP interval parameter for the LAN channel.
`lan-configuration-default-gateway-address`	14	IpAddress	Read-write	Default Gateway Address parameter for the LAN channel.
`lan-configuration-default-gateway-mac-address`	15	OCTET STRING (SIZE(6))	Read-write	Default Gateway MAC Address parameter for the LAN channel
`lan-configuration-backup-gateway-address`	16	IpAddress	Read-write	Backup Gateway Address parameter for the LAN channel.
`lan-configuration-backup-gateway-mac-address`	17	OCTET STRING (SIZE(6))	Read-write	Backup Gateway MAC Address parameter for the LAN channel.
`lan-configuration-community-string`	18	DisplayString (SIZE(0..255))	Read-write	Community String parameter for the LAN channel.
`lan-configuration-number-of-destinations`	19	INTEGER	Read-only	Number Of Destinations parameter for the LAN channel.
`lan-configuration-destination-type-0`	20	OCTET STRING (SIZE(3))	Read-write	Destination Type with Destination selector 0 for the LAN channel, excluding the Set Selector byte.

Variable	Index	Type	Access Mode	Description
`lan-configuration-destination-type-1`	21	OCTET STRING (SIZE(3))	Read-write	Destination Type with Destination selector 1 for the LAN channel, excluding the Set Selector byte.
`...`	...	...	...	...
`lan-configuration-destination-type-15`	35	OCTET STRING (SIZE(3))	Read-write	Destination Type with Destination selector 15 for the LAN channel, excluding the Set Selector byte.
`lan-configuration-destination-address-0`	36	OCTET STRING (SIZE(1|12))	Read-write	Destination Addresses with Destination selector 0 for the LAN channel, excluding the Set Selector byte.
`lan-configuration-destination-address-1`	37	OCTET STRING (SIZE(1|12))	Read-write	Destination Addresses with Destination selector 1 for the LAN channel, excluding the Set Selector byte.
`...`	...	...	...	...
`lan-configuration-destination-address-15`	51	OCTET STRING (SIZE(1|12))	Read-write	Destination Addresses with Destination selector 15 for the LAN channel, excluding the Set Selector byte.
`lan-configuration-vlan-id`	52	OCTET STRING (SIZE(2))	Read-write	Virtual LAN ID (12 bits), value 0 indicates that VLAN support is disabled for the channel.
`lan-configuration-vlan-priority`	53	OCTET STRING (SIZE(1))	Read-write	Virtual LAN priority (3 bits).
`lan-configuration-cipher-suite-entry-support`	54	OCTET STRING (SIZE(1))	Read-only	The number of supported Cipher Suite IDs.
`lan-configuration-cipher-suite-entries`	55	OCTET STRING (SIZE(0..16))	Read-only	The sequence of supported Cipher Suite IDs; each octet in the string contains one ID.
`lan-configuration-cipher-suite-priv-level`	56	OCTET STRING (SIZE(8))	Read-write	The sequence of supported Cipher Suite IDs; each octet in the string contains one ID.
`lan-configuration-destination-address-vlan-tag-0`	57	OCTET STRING (SIZE(3))	Read-only	Destination VLAN tag (3 bytes) for destination selector 0 for the given LAN channel.

Variable	Index	Type	Access Mode	Description
`lan-configuration-destination-address-vlan-tag-1`	58	OCTET STRING (SIZE(3))	Read-only	Destination VLAN tag (3 bytes) for destination selector 1 for the given LAN channel.
...	...	...	...	...
`lan-configuration-destination-address-vlan-tag-15`	72	OCTET STRING (SIZE(3))	Read-only	Destination VLAN tag (3 bytes) for destination selector 15 for the given LAN channel.

For example, to get the IP address of channel #1, use the following OID:
`<ROOT_OID>.7.1.5.1`

5.2.9 PEF Configuration Variables

The variables defined in this section contain PEF configuration information. This information is provided as several scalar SNMP variables and several SNMP tables.

The following scalar variables are defined for PEF configuration.
They have OIDs of the following form:
`<ROOT_OID>.<var>.0`

Here **`<var>`** is the index of a particular variable in the table entry describing a particular PEF configuration. The variable indices are defined in the table below.

Table 35 Advanced PEF Configuration Variables

Variable	Index	Type	Access Mode	Description
`pef-configuration-set-in-progress`	8	INTEGER	Read-only	Set In Progress parameter.
`pef-configuration-control`	9	INTEGER	Read-write	PEF Control parameter.
`pef-configuration-action-global-control`	10	INTEGER	Read-write	PEF Action global control parameter.
`pef-configuration-startup-delay`	11	INTEGER	Read-write	PEF Startup Delay parameter.
`pef-configuration-alert-startup-delay`	12	INTEGER	Read-write	PEF Alert Startup Delay parameter.
`pef-configuration-number-of-event-filters`	13	INTEGER	Read-only	Number of Event Filters parameter.
`pef-configuration-number-of-alert-policy-entries`	15	INTEGER	Read-only	Number of Alert Policy Entries parameter.

pef-configuration-system-guid	17	OCTET STRING (SIZE(16))	Read-write	System GUID parameter, excluding the "Used to fill in the GUID field in a PET Trap" byte.
pef-configuration-number-of-alert-strings	18	INTEGER	Read-only	Number of Alert Strings parameter.

For example, to get the PEF Startup Delay parameter, use the following OID:
<ROOT_OID>.11.0

A separate SNMP table is defined for PEF Event Filters. Each entry in this table provides information about a single PEF Event Filter. Table entries are indexed by filter numbers. The table entry with index 1 corresponds to filter number #0, table entry 2 – to filter number #1, etc.

PEF Event Filters variables have the following OID:
<ROOT_OID>.14.1.<var>.<filter>

Here **<var>** is the index of a particular variable in the table entry describing a particular PEF Event Filter. The variable indices are defined in the table below. **<filter>** is the filter number plus 1. Event Filter numbers start with 1; thus table entry with index 1 is not populated.

Table 36 PEF Event Filter Variables

Variable	Index	Type	Access Mode	Description
pef-configuration-event-filter-index	1	INTEGER	Read-only	Table entry index, equal to **<filter>**.
pef-configuration-event-filter-data	2	OCTET STRING (SIZE(20))	Read-write	Event Filter Table entry data, excluding the Set Selector byte.

For example, to get the PEF Event Filter Data #8, use the following OID:
<ROOT_OID>.14.1.2.9

A separate SNMP table is defined for PEF Alert Policies. Each entry in this table provides information about a single PEF Alert Policy. Table entries are indexed by policy numbers. The table entry with index 1 corresponds to alert policy #0, table entry 2 – to alert policy #1, etc.

PEF Alert Policy variables have the following OID:
<ROOT_OID>.16.1.<var>.<policy>

Here **<var>** is the index of a particular variable in the table entry describing a particular PEF Alert Policy. The variable indices are defined in the table below. **<policy>** is the policy number plus 1. Alert Policy numbers start with 1; thus table entry with index 1 is not populated.

Table 37 PEF Alert Policy Variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
pef-configuration-alert-policy-index	1	INTEGER	Read-only	Table entry index, equal to **<policy>**.
pef-configuration-alert-policy-data	2	OCTET STRING (SIZE(3))	Read-write	Alert Policy Table entry data, excluding the Set Selector byte.

For example, to get the PEF Configuration Alert Policy Data #8, use the following OID:
<ROOT_OID>.16.1.2.9

A separate SNMP table is defined for PEF Alert Strings. Each entry in this table provides information about a single PEF Alert String. Table entries are indexed by string numbers. The table entry with index 1 corresponds to alert string #0, table entry 2 – to alert string #1, etc.

PEF Alert String variables have the following OID:
<ROOT_OID>.19.1.<var>.<strnum>

Here **<var>** is the index of a particular variable in the table entry describing a particular PEF Alert String. The variable indices are defined in the table below. **<strnum>** is the alert string number plus 1.

Table 38 PEF Alert String Variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
pef-configuration-alert-string-index	1	INTEGER	Read-only	Table entry index, equal to **<strnum>**.
pef-configuration-alert-string-key	2	OCTET STRING (SIZE(2))	Read-write	Alert String Keys entry data, excluding the Set Selector byte.
pef-configuration-alert-string	3	DisplayString	Read-write	Alert Strings entry data, excluding the Set Selector byte.

For example, to get the PEF Configuration Alert String Key for string #8, use the following OID:
<ROOT_OID>.19.1.2.9

5.2.10 FRU Information Variables

The variables defined in this section contain information about the FRU Information in the shelf. This information is provided in the form of an SNMP table. Each entry in this table provides information about a single block of information for the designated FRU. The table lists all blocks of

FRUs for which FRU Device Locator Records (SDR Type 11h) or Management Controller Device Locator Records (SDR Type 12h) are present in the SDR Repository.

FRU device information variables have the following OID:
`<ROOT_OID>.20.1.<var>.<ipmb addr>.<fru_id>.<block number>`

Here **`<var>`** is the index of a particular variable in the table entry describing a particular FRU device. The variable indices are defined in the table below. **`<ipmb addr>`** and **`<fru_id>`** are the corresponding values of the specified FRU, **`<block number>`** is the 32-byte block offset within the FRU Info.

Table 39 FRU Info Block Variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
`fru-info-index`	1	INTEGER	Read-only	Index = (**`<ipmb addr>`** << 24) | (**`<fru_id>`** << 16) | **`<block number>`**.
`fru-info-data`	2	OctetString (SIZE(1..32))	Read-only	A block of data.
`fru-info-data-wo`	3	OctetString (SIZE(1..32))	Write-only	Write any number of bytes up to 32. Due to limitations in the current version of the SNMP agent, **`<block number>`** is interpreted as a byte offset.

For example, to get first 32 bytes (block number 0) of the FRU Information of FRU #254 at IPMB address 20h, use the following OID:
`<ROOT_OID>.20.1.2.32.254.0`

5.2.11 FRU Device by Site Variables

The variables defined in this section contain information about the FRU devices in the shelf. This information is provided in the form of an SNMP table. Each entry in this table provides information about a single FRU. The table lists all FRUs for which FRU Device Locator Records (SDR Type 11h) or Management Controller Device Locator Records (SDR Type 12h) are present in the SDR Repository.

FRU device information variables have the following OID:
`<ROOT_OID>.21.1.<var>.<site type>.<site number>`

Here **`<var>`** is the index of a particular variable in the table entry describing a particular FRU device. The variable indices are defined in the table below. **`<site type>`** and **`<site number>`** are the corresponding values of the specified FRU.

Table 40 Advanced FRU Device Variables

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
`fru-device-by-site-index`	1	INTEGER	Read-only	Table entry index, equal to (**`<site type>`** << 16) |**`<site number>`**.
`fru-device-by-site-sdr-version`	2	INTEGER	Read-only	SDR Version of the FRU Device or Management Controller Device Locator Record for this FRU. If the record is absent, this field is read as -1.
`fru-device-by-site-slave-address`	3	INTEGER	Read-only	Device Slave Address as defined in the FRU Device or Management Controller Device Locator Record for this FRU.
`fru-device-by-site-fru-device-id`	4	INTEGER	Read-only	FRU Device ID as defined in the FRU Device Locator Record for this FRU, or 0 for Management Controller devices. If the record is absent, this field is read as -1.
`fru-device-by-site-channel-number`	5	INTEGER	Read-only	Channel Number as defined in the FRU Device or Management Controller Device Locator Record for this FRU. If the record is absent, this field is read as -1.
`fru-device-by-site-device-type`	6	INTEGER	Read-only	For FRUs with FRU Device ID different from zero: Device Type as defined in the FRU Device Locator Record for this FRU. Since the Management Controller Device Locator Record doesn't provide the Device Type information, for Management Controller devices (FRU #0), this field is set to FRU Inventory Device (10h). If the record is absent, this field is read as -1.

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
`fru-device-by-site-device-type-modifier`	7	INTEGER	Read-only	For FRUs with FRU Device ID different from zero: Device Type Modifier as defined in the FRU Device Locator Record for this FRU. Since the Management Controller Device Locator Record doesn't provide the Device Type information, for Management Controller devices (FRU #0), this field is set to Unspecified (FFh). If the record is absent, this field is read as -1.
`fru-device-by-site-fru-entity-id`	8	INTEGER	Read-only	(FRU) Entity ID as defined in the FRU Device or Management Controller Device Locator Record for this FRU. If the record is absent, this field is read as -1.
`fru-device-by-site-fru-entity-instance`	9	INTEGER	Read-only	(FRU) Entity Instance as defined in the FRU Device or Management Controller Device Locator Record for this FRU. If the record is absent, this field is read as -1.
`fru-device-by-site-id-string`	10	DisplayString (SIZE(0..255))	Read-only	Device ID String as defined in the FRU Device or Management Controller Device Locator Record for this FRU. If the record is absent, this field is read as "N/A".
`fru-device-by-site-hot-swap-state`	11	INTEGER	Read-only	Current PICMG 3.0 FRU state (M1...M7) for this FRU. If this variable is equal to n, that means that the FRU is in state Mn.

VARIABLE	INDEX	TYPE	ACCESS MODE	DESCRIPTION
`fru-device-by-site-activated`	12	INTEGER	Read-write	When reading: 1 means that the FRU device is active (that is, in state M4), 0 is returned otherwise. Writing 1 to this variable triggers sending the "Set FRU Activation (Activate FRU)" command to this FRU, if the FRU is in state M2 or M5, and sending "Set FRU Activation Policy (Clear Locked)" command if the FRU is in state M1. Writing 0 to this variable triggers sending the "Set FRU Activation (Deactivate FRU)" command to this FRU, if the FRU is in state M2, M3, M4, or M5, and sending "Set FRU Activation Policy (Set Locked)" command if the FRU is in state M1 or M6.

For example, to get the Device ID String of the site type 2, site number 1, use the following OID:
`<ROOT_OID>.21.1.10.2.1`

5.3 Accessing the Shelf Manager via SNMP

5.3.1 SNMPv2c

Any SNMP client implementation should be able to access the Shelf Manager defined variables. One specific choice that we've used successfully is the net-snmp 5.0.6 package from: http://net-snmp.sourceforge.net/. This package would be installed on the management computer (running Linux kernel 2.4.2 and higher). It provides some basic management tools. To access the Pigeon Point SNMP agent, the **snmpget** and **snmpwalk** commands can be used.

To install the MIB file on the management system, follow the instructions supplied with the package.

After that, use the **snmpget** and **snmpwalk** commands to access selected variables. For example, to retrieve the variable controller-sdr-version for the controller 20h (BMC), use the following command:

```
snmpget -v 2c <Pigeon Point ipaddr> -c public
.iso.3.6.1.4.1.16394.2.1.1.1.1.2.32
```

The output will be similar to the following:

```
PPS-SENTRY-MIB::ipm-controller-sdr-version.32 = INTEGER: 81
```

To retrieve the entire pps-sentry variables subtree, use the following command:

```
snmpwalk -v 2c <Pigeon Point ipaddr> -c public
.iso.3.6.1.4.1.16394.2.1.1
```

The output will usually contain about 3000 strings for two IPM controllers with about 20 sensors on each of them.

This example assumes that SNMP v2c is used.

5.3.2 SNMPv3

In order to provide SNMPv3 functionality the SNMP agent should be properly configured. An example snmpd.conf file is provided below. It should exist in the **/etc** directory. This example shows how to configure: user: overlord and password: possessor, which has read-write rights and SNMPv3 access to the SNMP-agent. In order to access the SNMP agent in SNMPv3 mode, use the following commands.

To read a variable:

```
snmpget -v 3 -u <user name> -l authNoPriv -a MD5 -A <user password>
<Pigeon Point IP address> <variable OID with index>
```

To set a variable (for read-write variables only):

snmpset -v 3 -u <user name> -l authNoPriv -a MD5 -A <user password> <Pigeon Point IP address> <variable OID with index> <variable value>

For example, to retrieve the variable controller-sdr-version for the controller 20h (Shelf Manager), use the following command:
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor <Pigeon Point ipaddr> .iso.3.6.1.4.1.16394.2.1.1.1.1.2.32

Here is an example configuration file for the ucd-snmp agent:

```
########################################################################
########
#
# EXAMPLE.conf:
#   An example configuration file for configuring the ucd-snmp snmpd
agent.
#
########################################################################
########
#
# This file is intended to only be an example.  If, however, you want
# to use it, it should be placed in /usr/local/share/snmp/snmpd.conf.
# When the snmpd agent starts up, this is where it will look for it.
#
# Note: This file is automatically generated from EXAMPLE.conf.def.
# Do NOT read the EXAMPLE.conf.def file! Instead, after you have run
# configure & make, and then make sure you read the EXAMPLE.conf file
# instead, as it will tailor itself to your configuration.

# All lines beginning with a '#' are comments and are intended for you
# to read.  All other lines are configuration commands for the agent.

#
# PLEASE: read the snmpd.conf(5) manual page as well!
#


########################################################################
########
# Access Control
########################################################################
########

# YOU SHOULD CHANGE THE "COMMUNITY" TOKEN BELOW TO A NEW KEYWORD ONLY
# KNOWN AT YOUR SITE.  YOU *MUST* CHANGE THE NETWORK TOKEN BELOW TO
# SOMETHING REFLECTING YOUR LOCAL NETWORK ADDRESS SPACE.

# By far, the most common question I get about the agent is "why won't
# it work?", when really it should be "how do I configure the agent to
# allow me to access it?"
#
# By default, the agent responds to the "public" community for read
# only access, if run out of the box without any configuration file in
# place.  The following examples show you other ways of configuring
# the agent so that you can change the community names, and give
# yourself write access as well.
#
```

```
# The following lines change the access permissions of the agent so
# that the COMMUNITY string provides read-only access to your entire
# NETWORK (EG: 10.10.10.0/24), and read/write access to only the
# localhost (127.0.0.1, not its real ipaddress).
#
# For more information, read the FAQ as well as the snmpd.conf(5)
# manual page.

####
# First, map the community name (COMMUNITY) into a security name
# (local and mynetwork, depending on where the request is coming
# from):

rwuser overlord

#       sec.name  source          community
com2sec local     localhost       public
com2sec mynetwork 172.16.2.0/24   public

####
# Second, map the security names into group names:

#                sec.model  sec.name
group MyRWGroup v1         local
group MyRWGroup v2c        local
group MyRWGroup usm        local
group MyROGroup v1         mynetwork
group MyROGroup v2c        mynetwork
group MyROGroup usm        mynetwork

####
# Third, create a view for us to let the groups have rights to:

#           incl/excl subtree                          mask
view all    included  .1                               80

####
# Finally, grant the 2 groups access to the 1 view with different
# write permissions:

#                context sec.model sec.level match  read   write  notif
access MyROGroup ""      any       noauth    exact  all    none   none
access MyRWGroup ""      any       noauth    exact  all    all    none

# -----------------------------------------------------------------------
--------

engineID "Love_me_tender_lo"
createUser overlord MD5 possessor DES

#######################################################################
########
# System contact information
#

# It is also possible to set the sysContact and sysLocation system
# variables through the snmpd.conf file:
```

```
syslocation PPS experimental facility
syscontact PPS <support@pigeonpoint.com>


# Example output of snmpwalk:
#   % snmpwalk -v 1 localhost public system
#   system.sysDescr.0 = "SunOS name sun4c"
#   system.sysObjectID.0 = OID: enterprises.ucdavis.ucdSnmpAgent.sunos4
#   system.sysUpTime.0 = Timeticks: (595637548) 68 days, 22:32:55
#   system.sysContact.0 = "Me <me@somewhere.org>"
#   system.sysName.0 = "name"
#   system.sysLocation.0 = "Right here, right now."
#   system.sysServices.0 = 72


# -----------------------------------------------------------------------
--------


########################################################################
########
# Process checks.
#
#  The following are examples of how to use the agent to check for
#  processes running on the host.  The syntax looks something like:
#
#  proc NAME [MAX=0] [MIN=0]
#
#  NAME:  the name of the process to check for.  It must match
#         exactly (ie, http will not find httpd processes).
#  MAX:   the maximum number allowed to be running.  Defaults to 0.
#  MIN:   the minimum number to be running.  Defaults to 0.

#
#  Examples:
#

#  Make sure mountd is running
#proc mountd

#  Make sure there are no more than 4 ntalkds running, but 0 is ok too.
#proc ntalkd 4

#  Make sure at least one sendmail, but less than or equal to 10 are
running.
#proc sendmail 10 1

# -----------------------------------------------------------------------
--------
```

Here the demo script is provided that shows how to access the writeable variables.

```
#!/bin/bash
# fru activation
```

```
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.2.1.12.32.0
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.2.1.12.32.0 i 2

# emulated temp
# unr
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.3.1.35.156.2
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.3.1.35.156.2 i 100
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.3.1.35.156.2
# uc
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.3.1.36.156.2
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.3.1.36.156.2 i 100
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.3.1.36.156.2
# unc
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.3.1.37.156.2
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.3.1.37.156.2 i 100
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.3.1.37.156.2

# lnr
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.3.1.38.156.2
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.3.1.38.156.2 i 100
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.3.1.38.156.2
# lc
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.3.1.39.156.2
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.3.1.39.156.2 i 100
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.3.1.39.156.2
# lnc
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.3.1.40.156.2
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.3.1.40.156.2 i 100
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.3.1.40.156.2

#board reset
#1-16: 86 = 3
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.4.1.4.3
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.4.1.4.3 i 1
```

```
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.4.1.4.3
# auth port enabled
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.4.1
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.4.1 x "00 11 11 11 00"
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.4.1
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.4.1 x "00 15 15 15 00"
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.4.1

# IP addr
# dangerous, may shutdown network interface
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.5.1
#snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor
172.16.2.203 .iso.3.6.1.4.1.16394.2.1.1.7.1.5.1 x "C0 A0 B0 D0"
#snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor
172.16.2.203 .iso.3.6.1.4.1.16394.2.1.1.7.1.5.1

# MAC address
# dangerous, may shutdown network interface
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.7.1

# Subnet Mask
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.8.1

# IPv4
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.9.1

# Primary RMCP port 623
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.10.1
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.10.1 i 10623
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.10.1
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.10.1 i 623
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.10.1

# Secondary RMCP port 624
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.11.1
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.11.1 i 824
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.11.1
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.11.1 i 624
```

```
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.11.1

# BMC Generated ARP Control
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.12.1
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.12.1 i 1
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.12.1
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.12.1 i 2
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.12.1

# ARP Interval
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.13.1
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.13.1 i 25
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.13.1
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.13.1 i 4
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.13.1

# Gateway IP
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.14.1
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.14.1 x "C0 C0 C0 C0"
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.14.1
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.14.1 x "00 00 00 00"
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.14.1

# Gateway MAC
# does not work :(
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.15.1
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.15.1 x "DE AD CA FE DE AD"
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.15.1
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.15.1 x ""
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.15.1

# Backup IP
# absent check on length
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.16.1
```

```
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.16.1 x "C0 B0 A0 90"
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.16.1
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.16.1 x ""
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.16.1

# Backup MAC
# does not work :(
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.17.1
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.17.1 x "DE AD CA FE DE AD"
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.17.1
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.17.1 x ""
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.17.1

# Community
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.18.1
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.18.1 x "DE AD CA FE DE AD"
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.18.1
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.18.1 s "public"
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.18.1

# Destination type
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.20.1
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.20.1 x "05 BB CC"
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.20.1
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.20.1 x "00 00 00"
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.20.1

snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.33.1
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.33.1 x "07 08 44"
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.33.1
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.33.1 x "00 00 00"
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.33.1
```

```
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.3.1.35

# Destination Address
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.36.1
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.36.1 x "11 22 33 44 55 66 77 88 99 AA BB
CC FF"
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.36.1
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.36.1 x ""
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.36.1

snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.43.1
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.43.1 x "D1 D2 D3 D4 D5 D6 D7 D8 D9 AA BB
CC DD"
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.43.1
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.43.1 x ""
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.7.1.43.1

snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.3.1.51

# PEF Control
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.9.0
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.9.0 i 7
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.9.0
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.9.0 i 0
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.9.0

# Action control
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.10.0
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.10.0 i 7
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.10.0
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.10.0 i 0
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.10.0

# Startup Delay
```

```
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.11.0
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.11.0 i 15
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.11.0
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.11.0 i 60
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.11.0

# Alert Startup Delay
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.12.0
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.12.0 i 23
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.12.0
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.12.0 i 60
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.12.0

#event filter table data

# Event filter table data
# 2-64
#
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.14.1.2.5
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.14.1.2.5 x "80 3F 33 44 55 66 77 88 11 22 33
44 55 66 77 88 99 99 99 99"
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.14.1.2.5
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.14.1.2.5 x ""
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.14.1.2.5


snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.14.1.2.8
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.14.1.2.8 x "FF FF 33 44 55 66 77 88 11 22 33
44 55 66 77 88 99 99 99 99"
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.14.1.2.8
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.14.1.2.8 x ""
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.14.1.2.8

# Alert policy table data
# 2-64
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.16.1.2.2
```

```
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.16.1.2.2 x "FF FF FF"
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.16.1.2.2
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.16.1.2.2 x ""
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.16.1.2.2


snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.16.1.2.9 x "FF 11 35"
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.16.1.2.9


# Alert string table key
# 1-64
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.19.1.2.1
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.19.1.2.1 x "FF FF"
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.19.1.2.1
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.19.1.2.1 x ""
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.19.1.2.1

# Alert string table string
# 1-64
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.19.1.3.1
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.19.1.3.1 s "Test Alert String"
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.19.1.3.1
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.19.1.3.1 s ""
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.19.1.3.1



snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.19.1.3.9
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.19.1.3.9 s "@Cry"
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.19.1.3.9
snmpset -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.19.1.3.9 s ""
snmpget -v 3 -u overlord -l authNoPriv -a MD5 -A possessor 172.16.2.203
.iso.3.6.1.4.1.16394.2.1.1.19.1.3.9
```

This example assumes that SNMP v3 is used.

6 RMCP Interface

The IPMI specification defines a LAN Interface to the chassis or shelf, represented in the Pigeon Point context by the Shelf Manager. This interface is based on the RMCP (Remote Management Control Protocol). RMCP messages provide encapsulation for IPMI commands and responses adding special headers. On the transport level, RMCP messages are transferred across the network as UDP packets. AdvancedTCA requires that Shelf Managers support the IPMI LAN Interface and RMCP as part of the System Manager Interface.

The RMCP interface supports multi-session IPMI mode, allowing multiple users to work with the Shelf Manager simultaneously, while still being isolated from each other. User and message authentication and privilege levels are supported for RMCP sessions.

The Pigeon Point Shelf Manager fully supports the RMCP interface for the System Manager's interactions with the shelf in accordance with the IPMI specification v.1.5.1. Refer to chapter 12 of that specification for detailed information about the RMCP interface.

The Shelf Manager also supports the RMCP+ interface in accordance with the 2.0 revision of the IPMI specification. Detailed information about RMCP+ is contained in sections 13 and 24 of that specification.

The Shelf Manager fully supports the mandatory RMCP+ algorithms for authentication (RAKP-none, RAKP-HMAC-SHA1), integrity (none, HMAC-SHA1-96) and confidentiality (none, AES-CBC-128). In addition, the Shelf Manager supports optional encryption algorithms xRC4-128 and xRC4-40.

However, the Shelf Manager software on the ShMM-1500 is optionally delivered with encryption code removed, for reasons having to do with export regulations. If present, the encryption-related code is located in the shared library **`/lib/libpps_encryption.so`**. If the encryption library is absent, the Shelf Manager does not allow opening RMCP+ sessions with encryption support (and only the confidentiality algorithm "none" is supported in that case).

6.1 Supported IPMI Commands

The following table 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 41 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	App	05h	Not supported	Not supported

COMMAND	NETFN	CMD	ARRIVING FROM RMCP INTERFACE	ARRIVING FROM IPM CONTROLLER
On				
Set ACPI Power State	App	06h	Supported	Supported
Get ACPI Power State	App	07h	Supported	Supported
Get Device GUID	App	08h	Supported	Supported
Get NetFn Support	App	09h	Supported	Supported
Get Command Support	App	0Ah	Supported	Supported
Get Command Sub-function Support	App	0Bh	Supported	Supported
Get Configurable Commands	App	0Ch	Supported	Supported
Get Configurable Command Sub-functions	App	0Dh	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
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

COMMAND	NETFN	CMD	ARRIVING FROM RMCP INTERFACE	ARRIVING FROM IPM CONTROLLER
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	Supported	Not supported
Get Channel Payload Version	App	4Fh	Supported	Not supported
Get Channel OEM Payload Info	App	50h	Not supported	Not supported
Master Write-Read	App	52h	Supported (HPDL-based systems only)	Supported (HPDL-based systems only)
Get Channel Cipher Suites	App	54h	Supported	Not supported
Suspend/Resume Payload Encryption	App	55h	Supported (RMCP+ sessions only)	Not supported
Set Channel Security Keys	App	56h	Supported (RMCP+ sessions only)	Not supported
Get System Interface Capabilities	App	57h	Not supported	Not supported
Set System Info	App	58h	Supported (RMCP+ sessions only)	Not supported
Get System Info	App	59h	Supported (RMCP+ sessions only)	Not supported
Set Command Enables	App	60h	Supported	Supported

COMMAND	NETFN	CMD	ARRIVING FROM RMCP INTERFACE	ARRIVING FROM IPM CONTROLLER
Get Command Enables	App	61h	Supported	Supported
Set Command Sub-function Enables	App	62h	Supported	Supported
Get Command Sub-function Enables	App	63h	Supported	Supported
Get OEM NetFn IANA Support	App	64h	Supported	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

Command	NetFn	CMD	Arriving from RMCP Interface	Arriving from IPM Controller
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
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(*)
Set Serial Routing Mux	Transport	1Ch	Not supported	Not 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
Forwarded Command	Transport	30h	Not supported	Not supported
Set Forwarded Commands	Transport	31h	Not supported	Not supported
Get Forwarded Commands	Transport	32h	Not supported	Not supported
Enable Forwarded Commands	Transport	33h	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

COMMAND	NETFN	CMD	ARRIVING FROM RMCP INTERFACE	ARRIVING FROM IPM CONTROLLER
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	2Ah	Not supported	Not supported
Exit SDR Repository Update Mode	Storage	2Bh	Supported	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
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
Get SEL Time UTC Offset	Storage	5Ch	Not supported	Not supported
Set SEL Time UTC Offset	Storage	5Dh	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	S/E	13h	Supported	Supported

Command	NetFn	CMD	Arriving from RMCP Interface	Arriving from IPM Controller
Configuration Parameters				
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	2Ah	Supported	Supported
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
Set Sensor Reading and Event Status	S/E	30h	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

COMMAND	NETFN	CMD	ARRIVING FROM RMCP INTERFACE	ARRIVING FROM IPM CONTROLLER
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 Manager IPMB Address	PICMG	1Bh	Supported	Supported
Set Fan Policy	PICMG	1Ch	Supported	Supported
Get Fan Policy	PICMG	1Dh	Supported	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
Get Shelf Power Allocation	PICMG	22h	Supported	Supported
Get Telco Alarm Capability	PICMG	29h	Supported	Supported
Set Telco Alarm State	PICMG	2Ah	Supported	Supported
Get Telco Alarm State	PICMG	2Bh	Supported	Supported

COMMAND	NETFN	CMD	ARRIVING FROM RMCP INTERFACE	ARRIVING FROM IPM CONTROLLER
Get Target Upgrade Capabilities	PICMG	2Eh	Supported	Supported
Get Component Properties	PICMG	2Fh	Supported	Supported
Abort Firmware Upgrade	PICMG	30h	Supported	Supported
Initiate Upgrade Action	PICMG	31h	Supported	Supported
Upload Firmware Block	PICMG	32h	Supported	Supported
Finish Firmware Upload	PICMG	33h	Supported	Supported
Get Upgrade Status	PICMG	34h	Supported	Supported
Activate Firmware	PICMG	35h	Supported	Supported
Query Self-test Results	PICMG	36h	Supported	Supported
Query Rollback Status	PICMG	37h	Supported	Supported
Initiate Manual Rollback	PICMG	38h	Supported	Supported
Get Telco Alarm Location	PICMG	39h	Supported	Supported
Set FRU Extracted	PICMG	3Ah	Supported	Supported

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

6.1.1 Get Device ID Response

The command "Get Device ID" is intended to return information about the target IPMC. When this command is directed to the Shelf Manager, it returns information about the Shelf Manager, however this information is different in some fields depending on whether the destination address is 20h, which designates the logical Shelf Manager, or the address of the physical Shelf Manager.

The fields in the "Get Device ID" response have the following values and meaning:

Table 42 Fields of Get Device ID Response

FIELD NAME	OFFSET	VALUE FOR THE LOGICAL SHELF MANAGER	VALUE FOR THE PHYSICAL SHELF MANAGER
Device ID	2	Identifier of the target ShMM, as follows: 0 = ISM 3 = ShMM-300 5 = ShMM-500 6 = ShMM-500EL	

Field Name	Offset	Value for the Logical Shelf Manager	Value for the Physical Shelf Manager
		15h = ShMM-1500	
Device Revision	3	Bit [7] = 1: device provides Device SDRs Bits [6:0] = 0	
Firmware Revision 1	4	Bit [7]: 0 = device available, 1 = initialization is in progress Bits [6:0] = major Shelf Manager version, binary encoded (3 for version 3.2.0)	
Firmware Revision 2	5	Minor Shelf Manager version (second and third levels of the version), BCD encoded (20h for version 3.2.0)	
IPMI Version	6	02h , meaning IPMI Version 2.0 is supported	
Additional Device Support	7	BFh, which identifies the following capabilities: • Chassis Device • IPMB Event Generator • IPMB Event Receiver • FRU Inventory Device • SEL Device • SDR Repository Device • Sensor Device	29h, which identifies the following capabilities: • IPMB Event Generator • FRU Inventory Device • Sensor Device
Manufacturer ID	8:10	0Ah 40h 00h = PPS IANA	
Product ID	11:12	00h 00h	
Auxiliary Firmware Revision Information	13:16	Detailed information about the version of the Shelf Manager and type and subversion of the carrier-specific module. Byte 1: If bit [7] = 0, the major Shelf Manager version in bits [6:0], binary encoded (3 for version 3.2.0) If bit [7] = 1, the fourth level of Shelf Manager version in bits [6:0], binary encoded Byte 2: Minor Shelf Manager version (second and third levels of the version), BCD encoded (20h for version 3.2.0) Byte 3:	Data from the file "/var/nvdata/aux-fw-revision", as described in the *User Guide*, section 3.9

FIELD NAME	OFFSET	VALUE FOR THE LOGICAL SHELF MANAGER	VALUE FOR THE PHYSICAL SHELF MANAGER
		Carrier-specific module. Indicates the currently used carrier-specific module. For HPDL systems, the value is 1Ch Byte 4: Bits [7:4] = carrier subtype, if multiple carrier types are serviced by the same carrier-specific module; Bits [3:0] = carrier subversion. This value is normally 0, and is greater than 0 if the current carrier-specific module has been modified within the current release of the Shelf Manager.	

6.1.2 Get Self Test Results Response

The command “Get Self Test Results” returns the results of the POST tests performed by the Monterey Linux U-Boot at the startup of the ShMM. If all tests have passed, the status code 0x55 is returned. If any tests have failed, the device-specific failure code 0x59 is returned. The third byte contains the following bit mask in that case:

[7:5] Reserved
[4] 1b = Ethernet test failed
[3] 1b = UART test failed
[2] 1b = U-Boot CRC test failed
[1] 1b = I2C test failed
[0] 1b = Memory test failed.

Both the logical Shelf Manager and the physical Shelf Manager return the same response to this command,

6.2 Pigeon Point Extension Commands and Sensors

In addition to the specification-defined IPMI commands listed above, the Shelf Manager implements several Pigeon-Point-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 Pigeon Point 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 Pigeon Point 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.

These commands 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). Previously, for certain carriers, these commands were also implemented as Controller-Specific OEM/Group commands (Network Function Code 3Eh), but that latter variant is no longer supported.

Another set of Pigeon Point extensions provide additional functionality, currently including the following:

- Reading the contents of a Shelf FRU Info multirecord;
- Notifying the Shelf Manager about an extracted FRU;
- Initiating a Shelf Manager switchover;
- Subscribing for event notifications;
- DHCP client control;
- Diagnostic support;
- Allowing/disallowing AXIe functionality to enable PCI Express hosts.

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).

One more set of Pigeon Point extensions serves to improve the performance of Pigeon Point OpenHPI, when operating with the Pigeon Point Shelf Manager. The Pigeon Point Shelf Manager

implements an internal cache of the Device SDR Repository for every known IPM controller. The Shelf Manager uses that cache for internal needs and initially did not provide an external interface to access it. The Shelf Manager periodically checks the consistency of this cache.
A set of Pigeon Point OEM commands provides an external interface to this cache, including the following functions:

- Getting information about the cached Device SDR Repository;
- Obtaining an SDR from the cached Device SDR Repository;
- Reserving the cached Device SDR Repository.

6.2.1 TELCO Alarm Sensor

This discrete sensor has sensor type DFh, event/reading type 6Fh (sensor-specific discrete) and is implemented on LUN 0, number 131 of the Shelf Manager IPM controller (20h). The sensor name is “TELCO Alarms”. The following offsets are defined for the sensor:

- 0 – Minor Alarm active,
- 1 – Major Alarm active,
- 2 – Critical Alarm active,
- 3 – Alarm Cutoff active.

6.2.2 Set/Clear TELCO Alarms

This extension command is implemented by the Shelf Manager IPM controller (address 20h).
Network Function Code (NetFN): 3Eh
Command Code: D0h

Table 43 Set/Clear TELCO Alarms Command

	Byte	Data Field
Request Data	1	*Set Alarm Mask* A bit field that defines which alarm to set. [7:3] Reserved [2] Set Critical Alarm [1] Set Major Alarm [0] Set Minor Alarm
	2	*Clear Alarm Mask* A bit field that defines which alarm to clear. [7:3] Reserved [2] Clear Critical Alarm [1] Clear Major Alarm [0] Clear Minor Alarm
Response Data	1	*Completion Code*

6.2.3 Get TELCO Alarm Sensor Number

This extension command is implemented by the Shelf Manager IPM controller (address 20h). It allows the caller to obtain the sensor number for the TELCO Alarm sensor, after which the caller can read the state of TELCO Alarms in an IPMI-compliant way.

Network Function Code (NetFN): 3Eh

Command Code: D1h

Table 44 Get TELCO Alarm Sensor Number Command

	BYTE	DATA FIELD
Request Data	-	
Response Data	1	*Completion Code*
	2	*Sensor Number*. Identifies the TELCO alarm sensor.

6.2.4 Query Digital Output Properties

This extension command returns the number of available digital outputs.

Network Function Code (NetFN): 2Eh
Command Code: D4h

Table 45 Query Digital Output Properties Command

	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	*FRU Device ID*. Indicates an individual FRU device to query.
Response Data	1	*Completion Code*
	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	*Digital Output Count.* The number of digital outputs supported by this FRU.

6.2.5 Get Digital Outputs

This extension command allows the caller to query the current state of selected digital outputs. The state of each digital output is represented with one bit, so one byte represents the state of 8 digital outputs. All digital outputs are logically divided into groups, with each group containing 8 digital outputs and fitting in one byte. Group 0 comprises digital outputs 0-7, group 1 comprises digital outputs 8-15, etc.

The caller specifies the starting and ending group numbers for the digital outputs desired. Both numbers are optional; the default for the starting group number is 0, the default for the ending group number is the last group. Omitting both parameters causes retrieval of the state of all available digital outputs.

Network Function Code (NetFN): 2Eh
Command Code: D3h

Table 46 Get Digital Outputs Command

	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	*FRU Device ID*. Indicates an individual FRU device to query.
	(5)	*Starting group ID*. This parameter is optional; it defaults to 0.
	(6)	*Ending group ID*. This parameter is optional; it defaults to the last group.
Response Data	1	*Completion Code.*
	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	*Data byte 1*. The state of digital outputs for the first requested group
	…	….
	N	*Data byte N-1*. The state of digital outputs for the last requested group.

6.2.6 Set/Clear Digital Outputs

This extension command allows the caller to simultaneously set/clear the current state of digital outputs from the selected group. The state of each digital output is represented with one bit, so one byte represents the state of 8 digital outputs. All digital outputs are logically divided into groups, with each group containing 8 digital outputs and fitting in one byte. Group 0 comprises digital outputs 0-7, group 1 comprises digital outputs 8-15, etc.

The specified digital outputs within the group are set/cleared atomically with one operation.

Network Function Code (NetFN): 2Eh
Command Code: D2h

Table 47 Set/Clear Digital Outputs Command

	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	*FRU Device ID*. Indicates an individual FRU device to access.
	5	*Group ID.* Indicates the group of digital outputs to be set /cleared.
	6	*Digital Outputs Set Mask Bits*. A bit mask of digital outputs to be set within the group specified. [7] – 1b = set digital output 7 in the group specified [6] – 1b = set digital output 6 in the group specified [5] – 1b = set digital output 5 in the group specified [4] – 1b = set digital output 4 in the group specified [3] – 1b = set digital output 3 in the group specified [2] – 1b = set digital output 2 in the group specified [1] – 1b = set digital output 1 in the group specified [0] – 1b = set digital output 0 in the group specified

	7	*Digital Outputs Clear Mask Bits*. A bit mask of digital outputs to be cleared within the group specified. [7] – 1b = clear digital output 7 in the group specified [6] – 1b = clear digital output 6 in the group specified [5] – 1b = clear digital output 5 in the group specified [4] – 1b = clear digital output 4 in the group specified [3] – 1b = clear digital output 3 in the group specified [2] – 1b = clear digital output 2 in the group specified [1] – 1b = clear digital output 1 in the group specified [0] – 1b = clear digital output 0 in the group specified
Response Data	1	*Completion Code.*
	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.

6.2.7 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.)

The number of bytes read by this command is limited by the size of an IPMB packet and must be 20 bytes or fewer.

Network Function Code (NetFN): 2Eh
Command Code: 1h
IANA: 00400Ah (Assigned to PPS)

Table 48 Get Shelf FRU Record Data Command

	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.
Response Data	1	*Completion Code*
	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)

6.2.8 Notify Shelf Manager About an Extracted FRU

This Pigeon Point 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 D5h (Command Not Supported in Present State) is returned.

Network Function Code (NetFN): 2Eh

Command Code: 2h
IANA: 00400Ah (Assigned to PPS)

Table 49 Notify Shelf Manager About an Extracted FRU Command

	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.
Response Data	1	*Completion Code.*
	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.

6.2.9 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 D5h (Command Not Supported in Present State) is returned.

Network Function Code (NetFN): 2Eh
Command Code: 3h
IANA: 00400Ah (Assigned to PPS)

Table 50 Initiate Shelf Manager Switchover Command

	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.* [7:1] Reserved. Shall be set to 0b. [0] Active Reboot Mode. If this command is sent to the active Shelf Manager and this bit is set to 1, the Shelf Manager reboots its ShMM; if this bit is set to 0, the Shelf Manager exits without rebooting the ShMM. This bit is not applicable if the command is sent to the backup Shelf Manager.
Response Data	1	*Completion Code*

	BYTE	DATA FIELD
	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.

6.2.10 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.

Network Function Code (NetFN): 2Eh
Command Code: 4h
IANA: 00400Ah (Assigned to PPS)

Table 51 Subscribe for Event Notifications Command

	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.*
	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.

6.2.11 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.

Network Function Code (NetFN): 2Eh
Command Code: 05h
IANA: 00400Ah (Assigned to PPS)

Table 52 Set Shelf FRU Record Data Command

	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 written.
	11:N+10	*Data.* This variable length field contains data to be written into the record.
Response Data	1	*Completion Code.*
	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 written.* Indicates the number of bytes written to the record.

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 = 09h
Data = 41h 08h 00h 12h 09h 00 43h 0Ah 00 (9 bytes)

6.2.12 Get Cached Device SDR Info

This Pigeon Point extension command can be used by an RMCP client to get information about the device SDRs from a shelf's IPM controllers that the Pigeon Point Shelf Manager caches in its internal data structures during operation. The command request and response data are structured like the corresponding data in the IPMI "Get Device SDR Info" command with minimal changes.

Network Function Code (NetFN): 2Eh
Command Code: 6h

Table 53 Get Cached Device SDR Info

	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 an IPM Controller for which a Device SDR Repository is cached
Response Data	1	*Completion Code.*
	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	*Number of sensors in device for LUN this command is addressed to.*
	6	*Flags:* *Dynamic population* [7]: 0b =static sensor population. The number of sensors handled by this device is fixed, and a query returns records for all sensors. 1b = dynamic sensor population. The device may have its sensor population vary during 'run time' (defined as any time other than when an install operation is in progress). [6:4] Reserved *Device LUNs* [3]: 1b = LUN 3 has sensors, [2]: 1b = LUN 2 has sensors, [1]: 1b = LUN 1 has sensors, [0]: 1b = LUN 0 has sensors.
	7:10	*Sensor population change indicator.* LS byte first. Four byte timestamp or counter. Updated or incremented each time the sensor population changes. This field is not provided if the flags indicate a static sensor population.

6.2.13 Get Cached Device SDR

This Pigeon Point extension command can be used by an RMCP client to obtain an SDR from the collection of device SDRs from various IPM controllers that the Pigeon Point Shelf Manager caches in its internal data structures during operation. The command request and response data are structured like the corresponding data in the IPMI “Get Device SDR” command with minimal changes.

Network Function Code (NetFN): 2Eh
Command Code: 7h

Table 54 Get Cached Device SDR

	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 an IPM Controller for which the Device SDR Repository is cached.
	5	*Reservation ID Low Byte.* Only required for partial reads with non-zero 'Offset into record' field. Use 0000h for Reservation ID otherwise.
	6	*Reservation ID High Byte.*
	7	*Record ID of record to Get. Low Byte.* 0000h returns the first record.
	8	*Record ID of record to Get. High Byte.*
	9	*Offset into record.*
	10	*Bytes to read.* FFh means read entire record.
Response Data	1	*Completion Code*. Generic, plus following command specific: 80h = record changed. This status returned if any of the record contents has been altered since the last time the Requester issued the request with 00h for the *Offset into record* field.
	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	*Record ID for the next record. Low Byte.*
	6	*Record ID for the next record. High Byte.*
	7:N+6	*Requested bytes from record.*

6.2.14 Reserve Cached Device SDR Repository

This Pigeon Point extension command can be used by an RMCP client to obtain a *Reservation ID* for the collection of device SDRs from various IPM controllers that the Pigeon Point Shelf Manager caches in its internal data structures during operation. The Reservation ID is a part of the mechanism that is used to notify the requester that a record may have changed during the process of a multi-part read. The command request and response data are structured like the corresponding data in the IPMI “Reserve Device SDR Repository” command with minimal changes.

Network Function Code (NetFN): 2Eh
Command Code: 8h

Table 55 Reserve Cached Device SDR Repository

	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 IPM Controller which Device SDR Repository is cached.
Response Data	1	*Completion Code.*
	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	*Reservation ID. Low Byte.* 0000h reserved.
	6	*Reservation ID. High Byte.*

6.2.15 Set FRU Activation Parameters

This Pigeon Point extension command can be used by an RMCP client to manage the slot-specific attributes Shelf Manager Controlled Activation, Shelf Manager Controlled Deactivation and Delay Before Next Power On. These attributes are stored in the Shelf FRU Information and normally the changes done to the Shelf FRU Information do not effect Shelf Manager operation immediately. This command doesn't change Shelf FRU Information, but changes the values of power management attributes for the specified slot, so that the Shelf Manager starts using the new values immediately.

Network Function Code (NetFN): 2Eh
Command Code: 9h
IANA: 00400Ah (Assigned to PPS)

Table 56 Set FRU Activation Parameters

	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 (NOT the hardware address) of the slot for which the attributes are changed.
	5	*FRU Device ID*. Indicates the FRU device ID of the slot for which the attributes are changed (the value FEh means “all FRUs”).
	6	*Value Presence Flags:* [7:3]: Reserved [2]: Set the Shelf Manager Controlled Activation bit from the next byte. [1]: Set the Shelf Manager Controlled Deactivation bit from the next byte. [0]: Set the Delay Before Next Power On value from the next byte.
	7	*Values:* [7]: Shelf Manager Controlled Deactivation bit. [6]: Shelf Manager Controlled Activation bit. [5:0]: Delay Before Next Power On value.
Response Data	1	*Completion Code.*
	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.

6.2.16 Set Minimal Fan Level

This command sets the minimal fan level for a specific fan tray or for all fan trays in the shelf.

Network Function Code (NetFN): 2Eh
Command Code: 0Ch
IANA: 00400Ah (Assigned to PPS)

Table 57 Set Min Fan Level

	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	*Level.* A new minimum fan level value.
	(5)	*IPMB Address*. An optional field. Indicates the IPMB address of the Fan Tray for which the minimum fan level is set. In a shelf that does not implement zoned cooling, this parameter is ignored. If this parameter and the *FRU Device ID* are omitted, the specified *Level* is set to all Fan Trays.
	(6)	*FRU Device ID.* An optional field. Indicates the FRU device ID of the Fan Tray for which the minimum fan level is set. In a shelf that does not implement zoned cooling, this parameter is ignored. If this parameter and the *IPMB Address* are omitted, the specified *Level* is set to all Fan Trays.
Response Data	1	*Completion Code.*
	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.

If *Completion Code* is not IPMI_SUCCESS, the rest of the response may not be returned. Parameters *IPMB Address* and *FRU Device ID* must be either both present or both omitted.

6.2.17 Get Minimal Fan Level

This command gets the minimal fan levels, both statically configured and dynamic, for a specific fan tray in the shelf.

Network Function Code (NetFN): 2Eh
Command Code: 0Dh
IANA: 00400Ah (Assigned to PPS)

Table 58 Get Min Fan Level

	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*. An optional field. Indicates the IPMB address of the Fan Tray for which the minimum fan level is queried. If this parameter and the *FRU Device ID* are omitted, the *CFG Min Fan Level* and *Dynamic Min Fan Level* parameters returned in the response relate to the first Fan Tray within the internal Shelf Manager list. In a shelf that does not implement zoned cooling, this parameter is ignored; the returned minimum fan level values apply to all fan trays in the shelf.
	(5)	*FRU Device ID.* An optional field. Indicates the FRU device ID of the Fan Tray for which the minimum fan level is queried. If this parameter and the *IPMB Address* are omitted, the *CFG Min Fan Level* and *Dynamic Min Fan Level* parameters returned in the response relate to the first Fan Tray within the internal Shelf Manager list. In a shelf that does not implement zoned cooling, this parameter is ignored; the returned minimum fan level values apply to all fan trays in the shelf.
Response Data	1	*Completion Code.*
	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	*CFG Minimum Fan Level*. The current value of the minimum level parameter specified manually (via the configuration file, a CLI command or via the RMCP command “Set Min Fan Level”).
	6	*Dynamic Minimum Fan Level*. The current value of the dynamically changed minimum fan level. This value is always greater or equal than the manually specified minimum fan level.

If *Completion Code* is not IPMI_SUCCESS, the rest of the response may not be returned.

6.2.18 DHCP Client Control

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 the backup Shelf Manager. It initiates a DHCP client restart on the active Shelf Manager or provides DHCP Client status from the active or the backup Shelf Manager. If a `dhcp restart` command is sent to backup Shelf Manager or to a Shelf Manager on which DHCP Client functionality is disabled, the completion code D5h (Command Not Supported in Present State) is returned. For other cases, the completion code IPMI_SUCCESS is returned and, in addition, the status of the DHCP client after the restart is reported in the DHCP Client Status byte in the command response data separately for each Ethernet channel (see Table 59).

Network Function Code (NetFN): 2Eh
Command Code: 0Eh
IANA: 00400Ah (Assigned to PPS)

Table 59 DHCP Client Control Command

	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	*Function Code.* 00h = restart DHCP client 01h = get the status of the DHCP client.
Response Data	1	*Completion Code*
	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	*DHCP Client Status:* Bits 7-4: Channel 1 status Bits 3-0: Channel 0 status Status: 0 = Completed (Done) 1 = Start/restart in progress 2 = Not started yet 3 = Stopped (Terminated) 4 = Received from active (Only on the backup ShelfManager) Eh = Error occurred Fh = Channel not used (LAN channel or DHCP Client is not configured)

If *Completion Code* is not IPMI_SUCCESS, the rest of the response may not be returned.

6.2.19 Diagnostics Initiator Commands

The following commands implement the Pigeon Point Diagnostics Initiators (DI) architecture in the Shelf Manager. The DI architecture defines Pigeon Point OEM IPMI commands and data structures that can be used to invoke diagnostics and tests on an arbitrary IPM Controller using HPI Diagnostics Initiator Management Instruments (DIMIs). HPI DIMI API functions and data structures are easily mapped to the DI commands and data structures. (In fact, the Diagnostic Infrastructure has been intentionally designed to facilitate such mapping.)

The DI architecture is described in a separate document.

6.2.19.1 Get Static DI Data

This command provides read-only access to the static DI information for the specified FRU. Using the specified offset and size, the requester can obtain a piece of the DI information structure; using multiple invocations of this command, the requester can read the entire DI information structure.

Network Function Code (NetFN): 2Eh
Command Code: 48h
IANA: 00400Ah (Assigned to PPS)

Table 60 Get Static DI Data Command

	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	*FRU Device ID*. Indicates the target FRU Device ID
	5:6	*Offset*. Offset in the static DI information data structure to read from (LS byte first)
	7	*Byte Count*. How many bytes to read from the specified offset
Response Data	1	*Completion Code.*
	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:N	*Data*. The requested piece of static DI information. The returned data size can be less than the requested size if *Offset* + *Byte Count* is greater than the size of the static DI Information data structure

If the *Completion Code* is not IPMI_SUCCESS, the rest of the response may not be returned.

The static DI information data structure has the following layout.

Table 61 Static DI Information Data Structure

Offset	Size	Data field
0	1	*Format Version*. Format version of this data structure; 0 for this specification
1	2	*Total DI Information Size*. LS byte first
3	1	*Record Checksum*. Holds the zero checksum for the whole record
4	1	*DI Count*. Number of DIs defined for this FRU
5	Variable	*DI Information Descriptors*. The sequence of DI information descriptors for this FRU

Each DI descriptor has the following layout:

Table 62 DI Information Descriptor

OFFSET	SIZE	DATA FIELD
0	2	*DI Information Descriptor Size*. LS byte first
2	1	*DI ID*
3	1	*Test Count*. The number of tests defined for this DI
4	N+1	*DI Name*. Text string in the IPMI FRU Information data format, with the first byte specifying the length (N). The *DI Name* must fit into a single IPMI string.
4+N+1	Variable	*Test Information Descriptors*. The sequence of test information descriptors for this DI

The layout of a test information descriptor is defined in the following table:

Table 63 Test Information Descriptor

OFFSET	SIZE	DATA FIELD
0	2	*Test Information Descriptor Size*. LS byte first
2	1	*Test ID*
3	1	*Test Capabilities*. Maps directly to the HPI type SaHpiDimiTestCapabilityT; see that definition for detailed descriptions of the individual capability bits: Bits 7 – reserved Bit 6 = 1 – this is an INFORMATIONAL test (the test is non-disruptive, always succeeds and reports certain information about the DI in its results string(s)) Bit 5 = 1 – test cancellation is supported (maps to the capability bit SAHPI_DIMITEST_CAPABILITY_TESTCANCEL) Bit 4 = 1 – test supports logging capability; the standard test parameter “Logging” is also defined for this test (maps to the capability bit SAHPI_DIMITEST_CAPABILITY_LOGGING) Bit 3 = 1 – test supports looping for a specified amount of time; the standard test parameter “Loop Time” is also defined for this test (maps to the capability bit SAHPI_DIMITEST_CAPABILITY_LOOPTIME) Bit 2 = 1 – test supports looping for a specified count of times; the standard test parameter “Loop Count” is also defined for this test (maps to the capability bit SAHPI_DIMITEST_CAPABILITY_LOOPCOUNT) Bit 1 = 1 – test supports a basic as well as extended mode; the standard test parameter “Service Mode” is also defined for this test (maps to the capability bit SAHPI_DIMITEST_CAPABILITY_SERVICEMODE) Bit 0= 1 – test supports FINALONLY, ONDEMAND, ASYNC results output capabilities; the standard test parameter “Results Output” is also defined for this test (maps to the capability bit SAHPI_DIMITEST_CAPABILITY_RESULTSOUTPUT)

4	8	*Expected Run Duration*. In nanoseconds, follows the format of Linux struct timespec: Bytes 0:3 – the number of whole seconds (LS Byte first) Bytes 4:7 – the number of nanoseconds (LS byte first)
12	1	*FRU Service Impact*. Specifies the service impact of the test on the entire FRU. Directly maps on the HPI enumeration SaHpiDimiTestServiceImpactT; the following values are defined: 0 = Non-Degrading 1 = Degrading 2 = Vendor-Defined All other values are reserved
13	1	*FRU Management Controller Service Impact*. Same values as for *FRU Service Impact*. In the case of an intelligent subsidiary FRU, the impact applies to the MC of that FRU (e.g. to an AMC's MMC). In the case of a non-intelligent subsidiary FRU or a managing FRU, the impact applies to the MC of the corresponding managing FRU.
14	1	*Impacted Entities Count*. The number of entities (M) affected by this test (up to 4): Bits 7:3 – reserved Bits 2:0 – the number of affected instances (in the range 0..4)
15	1	*Need Service OS* (Service OS String Count). Bits 7:4 - reserved Bits 3:0 – the number of text strings in the Service OS name If this number = 0, no Service OS is needed and the Service OS name is absent
16	1	*Parameter Count*. The number of test parameters defined for this test
17	N+1	*Test Name*. Text string in the FRU Information data format, with the first byte specifying the length (N). The Test Name must fit into a single IPMI string.
17+N+1	M*3	*Entities Impacted*. M (up to 4) Affected Entity Descriptors
17+N+1+M *3	Variable (L)	*Service OS Name*. One or more text strings representing the HPI string for the Service OS Name. Absent if *Need Service OS* = 0
17+N+1+M *3+L+1	Variable	*Parameter*s. The sequence of Parameter Information Descriptors for this test

NOTE: For the test service impact, HPI allows up to 5 additional affected entities to be listed. On the IPMI level, the FRU management controller can be one of these entities, and 4 other entities can be defined explicitly.

The layout of an affected entity descriptor is defined in the following table:

Table 64 Affected Entity Descriptor

OFFSET	SIZE	DATA FIELD

0	1	*Entity ID*. IPMI Entity ID of the affected entity
1	1	*Entity Instance*. IPMI Entity instance of the affected entity
2	1	*Service Impact*. Directly maps on the HPI enumeration SaHpiDimiTestServiceImpactT; the following values are defined: 0 = Non-Degrading 1 = Degrading 2 = Vendor-Defined All other values are reserved.

The layout of each parameter information descriptor is defined in the following table:

Table 65 Parameter Information Descriptor

OFFSET	SIZE	DATA FIELD
0	2	*Parameter Information Descriptor Size*. LS byte first
2	1	*Parameter ID*
3	1	*Parameter Type*. Directly maps to the SaHpiDimiTestParamTypeT enumeration, with several additional integer types: 0 = BOOLEAN: 1 byte 1 = Int32: 4 byte integer, LS byte first 2 = Float64: 8 byte float, LS byte first 3 = Text: HPI text string, represented by zero or more IPMI strings FDh = Int8: 1 byte signed integer FEh = Int16: 2 byte signed integer, LS byte first
4	1	*Parameter Information String Count*. Bits 7:4 - reserved Bits 3:0 – the number of text strings in the parameter information
5	N+1	*Parameter Name*. Text string in the FRU Information data format, with the first byte specifying the length (N). Length must be <= 20 characters and must consist of a single IPMI string.
5+N+1	M	*Min/Max/Default Value Descriptor*. Type-dependent
(5+N+1+M)	Variable	*Parameter Information*. Sequence of zero or more IPMI text strings describing the parameter for informational purposes; mapped to a single HPI string

Table 66 Min/Max/Default Value Descriptor for BOOLEAN Parameters

OFFSET	SIZE	DATA FIELD
0	1	*Default Value*. Default parameter value (0 = FALSE, 1 = TRUE)

Table 67 Min/Max/Default Value Descriptor for Int32 Parameters

OFFSET	SIZE	DATA FIELD
0	4	*Default Value*. Default parameter value. LS byte first
4	4	*Minimum Value*. Minimum parameter value. LS byte first
8	4	*Maximum Value*. Maximum parameter value. LS byte first

Table 68 Min/Max/Default Value Descriptor for Float64 Parameters

OFFSET	SIZE	DATA FIELD
0	8	*Default Value*. Default parameter value. LS byte first
8	8	*Minimum Value*. Minimum parameter value. LS byte first
16	8	*Maximum Value*. Maximum parameter value. LS byte first

Table 69 Min/Max/Default Value Descriptor for Text Parameters

OFFSET	SIZE	DATA FIELD
0	1	*Default Value String Count*. Bits 7:4 - reserved Bits 3:0 – the number of text strings in the parameter information If this number = 0, the rest of the data structure is absent and the default parameter value is an empty string
1	Variable	*Default Value*. Sequence of one or more IPMI strings representing a single HPI string – the default value. Absent if *Default Value String Count* = 0

Table 70 Min/Max/Default Value Descriptor for Int8 Parameters

OFFSET	SIZE	DATA FIELD
0	1	*Default Value*. Default parameter value
1	1	*Minimum Value*. Minimum parameter value
2	1	*Maximum Value*. Maximum parameter value

Table 71 Min/Max/Default Value Descriptor for Int16 Parameters

OFFSET	SIZE	DATA FIELD
0	2	*Default Value*. Default parameter value. LS byte first
2	2	*Minimum Value*. Minimum parameter value. LS byte first
4	2	*Maximum Value*. Maximum parameter value. LS byte first

6.2.19.2 Get DI Test Readiness Command

This command maps on the HPI API saHpiDimiTestReadinessGet() and returns the readiness state of the specified test.

Network Function Code (NetFN): 2Eh
Command Code: 49h
IANA: 00400Ah (Assigned to PPS)

Table 72 Get DI Test Readiness Command

	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	*FRU Device ID*. Indicates the target FRU Device ID
	5	*DI ID.* Indicates the target Diagnostic Instrument (DI) ID
	6	*Test ID*. Indicates the target test ID
Response Data	1	*Completion Code*. All general completion codes, including: C9h = Parameter Out Of Range: *DI ID* or *Test ID* do not match any existing Diagnostic Instrument or Test.
	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	Test Readiness State. This field directly maps on the SaHpiDimiReadyT enumeration: 0 = DI is ready to run the specified test 1 = DI is in a wrong state to run the specified test (e.g. need to load a correct Service OS before this test can be run) 2 = DI is busy and cannot run the specified test now, the user can try again later All other values are reserved

If the *Completion Code* is not IPMI_SUCCESS, the rest of the response may not be returned.

6.2.19.3 Get Start DI Test Reservation ID Command

This command obtains a reservation ID for a test start sequence. A test start sequence begins with a single “Get Start Test Reservation ID” command, followed by zero or more “Specify Test Parameters” commands, followed by a single “Start Test” command.

A Reservation ID is used to preserve the integrity of this sequence. The reservation ID obtained via the “Get Start Test Reservation ID” command must be specified in subsequent commands of the test start sequence. The reservation ID is invalidated by the responder if another “Get Start Test Reservation ID” command is received for the same DI ID and Test ID. In that case, the current test start sequence is invalidated and a new one is initiated.

When this command is received, the responder clears any test parameter value previously set and assigns defaults to all test parameters of the specified test.

Similar to other use cases of reservation IDs in IPMI, successful completion of this command does not guarantee successful execution of the test start sequence. For example, a pair of requesters could compete for a test start in such a manner that each one alternately cancels the reservation that is held by the other - effectively ‘deadlocking’ each other. Additional means of synchronization are needed between the requesters to avoid this situation; these means are outside the scope of this document.

Network Function Code (NetFN): 2Eh
Command Code: 4Ah
IANA: 00400Ah (Assigned to PPS)

Table 73 Get Start DI Test Reservation ID Command

	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	*FRU Device ID*. Indicates the target FRU Device ID
	5	*DI ID*. Indicates the target Diagnostic Instrument (DI) ID
	6	*Test ID*. Indicates the target test ID
Response Data	1	*Completion Code*. All general completion codes, including: C9h = Parameter Out Of Range: *DI ID* or *Test ID* do not match any existing DI or Test. D5h = Command Inconsistent in Present State: The test start sequence cannot be initiated at this time.
	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.
	5:6	*Reservation ID*. LS Byte first, value 0000h is reserved

If the *Completion Code* is not IPMI_SUCCESS, the rest of the response may not be returned.

6.2.19.4 Specify DI Test Parameters Command

This command specifies the parameters for a specific test to be started and is issued as part of the mapping of a call to the HPI API saHpiDimiTestStart(). One instance of this command specifies the value of a single parameter. This command may need to be issued multiple times even for a single parameter, if the parameter value is large.

For each parameter for which the value is not specified, the corresponding default value is used for the test run.

Network Function Code (NetFN): 2Eh
Command Code: 4Bh
IANA: 00400Ah (Assigned to PPS)

Table 74 Specify DI Test Parameters Command

	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	*FRU Device ID*. Indicates the target FRU Device ID
	5	*DI ID*. Indicates the target Diagnostic Instrument (DI) ID
	6	*Test ID*. Indicates the target test ID
	7:8	*Reservation ID*. This is the Reservation ID previously obtained via Get Start Test Reservation ID
	9	*Parameter ID*. Indicates the target Parameter ID

	10	*Offset*. Offset in the parameter value data where to write the following data
	11:N	*Value Data*. The entire or partial parameter value data
Response Data	1	*Completion Code*. All general completion codes, including: C5h = Reservation Broken or Invalid Reservation ID. C9h = Parameter Out Of Range: *DI ID*, *Test ID* or *Parameter ID* do not match any existing DI, Test or Parameter. CCh = Invalid Data In Request: specified *Offset* or/and *Value Data* exceed the parameter data storage capacity.
	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.

If the *Completion Code* is not IPMI_SUCCESS, the rest of the response may not be returned.

The parameter value data format depends on the parameter type specified in the parameter information descriptor, according to the following table:

Table 75 Parameter Value Data Format by Parameter Type

TYPE	SIZE (BYTES)	FORMAT
BOOLEAN	1	0 = FALSE, non-zero = TRUE
Int32	4	Integer value (signed), LS byte first
Float64	8	Floating-point value (IEEE 754-1985), LS byte first
Text	Variable	Byte 1: Bits 7:4 – reserved Bits 3:0 – number of following IPMI text strings, comprising the value Bytes 2:N – Sequence of IPMI text strings representing an HPI text string
Int8	1	Integer value (signed)
Int16	2	Integer value (signed), LS byte first

If a test parameter has the string type, multiple value strings may be specified as the parameter value. All of these must have the same string type and are concatenated to form the actual parameter value (an HPI string of up to 255 bytes).

If a reservation has been broken and subsequently reestablished, it is necessary to retransmit all test parameters.

6.2.19.5 Start DI Test Command

This command instructs the target IPMC to start the specified test; the test parameters must have been already passed as a sequence of “Specify Test Parameters” commands. This command is issued at the last phase of mapping a call to the HPI API saHpiDimiTestStart().

Network Function Code (NetFN): 2Eh

Command Code: 4Ch
IANA: 00400Ah (Assigned to PPS)

Table 76 Start DI Test Command

	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	*FRU Device ID*. Indicates the target FRU Device ID
	5	*DI ID*. Indicates the target Diagnostic Instrument (DI) ID
	6	*Test ID*. Indicates the target test ID
	7:8	*Reservation ID*. This is the Reservation ID previously obtained via Get Start Test Reservation ID
Response Data	1	*Completion Code*. All general completion codes, including: C5h = Reservation Broken or Invalid Reservation ID. C9h = Parameter Out Of Range: *DI ID*, *Test ID* or *Parameter ID* do not match any existing DI, Test or Parameter.
	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.
	5:12	*Timestamp*. The timestamp of the test run that has just been initiated (contains the number of nanoseconds since January 1, 1970, 00:00, LS byte first, follows the format of Linux struct timespec): Bytes 0:3 – the number of whole seconds (LS Byte first) Bytes 4:7 – the number of nanoseconds (LS byte first)

If the *Completion Code* is not IPMI_SUCCESS, the rest of the response may not be returned.

6.2.19.6 Cancel DI Test Command

This command instructs the target IPMC to cancel the specified test that normally would be currently running. It implements a mapping of the HPI API saHpiDimiTestCancel().

Network Function Code (NetFN): 2Eh
Command Code: 4Dh
IANA: 00400Ah (Assigned to PPS)

Table 77 Cancel DI Test Command

	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	*FRU Device ID*. Indicates the target FRU Device ID
	5	*DI ID*. Indicates the target Diagnostic Instrument (DI) ID

	6	*Test ID.* Indicates the target test ID
Response Data	1	*Completion Code*. All general completion codes, including: D5h = Command Inconsistent in Present State if the test is not running. C9h = Parameter Out Of Range: *DI ID*, *Test ID* or *Parameter ID* do not match any existing DI, Test or Parameter.
	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.

If *Completion Code* is not IPMI_SUCCESS, the rest of the response may not be returned.

6.2.19.7 Get DI Test Status Command

This command implements a mapping of the HPI API saHpiDimiTestStatusGet().

Network Function Code (NetFN): 2Eh
Command Code: 4Eh
IANA: 00400Ah (Assigned to PPS)

Table 78 Get DI Test Status Command

	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	*FRU Device ID*. Indicates the target FRU Device ID
	5	*DI ID*. Indicates the target Diagnostic Instrument (DI) ID
	6	*Test ID*. Indicates the target test ID
Response Data	1	*Completion Code*. All general completion codes, including: C9h = Parameter Out Of Range: *DI ID*, *Test ID* or *Parameter ID* do not match any existing DI, Test or Parameter.
	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	*Run Status*. Directly maps to the SaHpiDimiTestRunStatusT enumeration: 0 = Not Run. The test has never been executed on the DI 1 = Finished, No Errors. 2 = Finished with Errors. 3 = Canceled (with the "Cancel Test" command) 4 = Running; test is in progress All other values are reserved

	(6:13)	*Start Timestamp*. The start timestamp of the last executed or currently executing test (contains the number of nanoseconds since January 1, 1970, 00:00, LS byte first, follows the format of Linux struct timespec): Bytes 0:3 – the number of whole seconds (LS Byte first) Bytes 4:7 – the number of nanoseconds (LS byte first); Absent if Run Status = Not Run
	(14)	*Percentage of completion*. The value is in the range 0 to 100 or absent if not available or if *Run Status* = Not Run

If the *Completion Code* is not IPMI_SUCCESS, the rest of the response may not be returned.

6.2.19.8 Get DI Test Results Reservation ID Command

This command obtains a reservation ID for the test results. A Reservation ID is used to preserve the integrity of test results returned via multiple IPMI commands. A reservation ID is invalidated by the responder if test results change during a test results retrieval sequence (e.g. due to initiation of a new test run).

Network Function Code (NetFN): 2Eh
Command Code: 4Fh
IANA: 00400Ah (Assigned to PPS)

Table 79 Get DI Test Results Reservation ID Command

	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	*FRU Device ID*. Indicates the target FRU Device ID
	5	*DI ID*. Indicates the target Diagnostic Instrument (DI) ID
	6	*Test ID*. Indicates the target test ID
Response Data	1	*Completion Code*. All general completion codes, including: C9h = Parameter Out Of Range: *DI ID* or *Test ID* do not match any existing DI or Test.
	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.
	5:6	*Reservation ID*. LS Byte first, value 0000h is reserved

If the *Completion Code* is not IPMI_SUCCESS, the rest of the response may not be returned.

6.2.19.9 Get DI Test Results Command

This command implements a mapping of the HPI API saHpiDimiTestResultsGet().

Due to IPMI command size limitations, multiple commands of this type may be required to retrieve test results. To preserve the integrity of returned data, a Reservation ID is used. A Reservation ID is assigned to the caller via the command "Get Test Results Reservation ID" and is invalidated by the responder if test results change during a test result retrieval sequence (e.g. due to initiation of a new test run).

A requester can correlate the test results with a specific instance of the "Start Test" command by comparing the two timestamps: the one returned by the "Start Test" command and the one contained in the test results data.

Network Function Code (NetFN): 2Eh
Command Code: 50h
IANA: 00400Ah (Assigned to PPS)

Table 80 Get DI Test Results Command

	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	*FRU Device ID*. Indicates the target FRU Device ID
	5	*DI ID*. Indicates the target Diagnostic Instrument (DI) ID
	6	*Test ID*. Indicates the target test ID
	7:8	*Reservation ID*. LS byte first, 0000h reserved
	9:10	*Offset*. Offset in the results data structure to read from (LS byte first)
	11	*Byte Count*. How many bytes to read from the specified offset
Response Data	1	*Completion Code*. All general completion codes, including: D5h = Command Inconsistent in Present State if no results are present. C9h = Parameter Out Of Range: *DI ID* or *Test ID* do not match any existing DI or Test, or offset is beyond the size of results. CCh = Invalid Data In Request: specified *Offset* and/or *Byte Count* data exceed the result storage capacity.
	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:N	*Results Data*. A piece of data from the results data structure. The returned data size can be less than requested size if *Offset* + *Byte Count* is greater than the size of the results data structure

The layout of the DI results data is described in the following table:

Table 81 DI Test Results Data Structure

OFFSET	SIZE	DATA FIELD
0	1	*Format Version*. Format version of this data structure; 0 for this specification
1	2	*Results Data Size*. LS byte first

3	1	*Data Checksum*. Holds the zero checksum for the entire data structure
4	8	*Timestamp*. Timestamp of the last test start (contains the number of nanoseconds since January 1, 1970, 00:00, LS byte first, follows the format of Linux struct timespec): Bytes 0:3 – the number of whole seconds (LS Byte first) Bytes 4:7 – the number of nanoseconds (LS byte first) Note that this timestamp is the test start timestamp while the timestamp reported by the HPI API saHpiDimiTestResultsGet() is the test results generation timestamp; these two timestamps differ by the *Run Duration* value (the value of the next field).
12	8	*Run Duration*. In nanoseconds, follows the format of Linux struct timespec: Bytes 0:3 – the number of whole seconds (LS Byte first) Bytes 4:7 – the number of nanoseconds (LS byte first)
20	1	*Last Run Status*. Directly maps to the SaHpiDimiTestRunStatusT enumeration: 0 = Not Run. The test has never been executed on this DI 1 = Finished, No Errors. 2 = Finished with Errors. 3 = Canceled (with the "Cancel" Test command) 4 = Running; test is in progress All other values are reserved
21	1	*Error Code*. Directly maps to the SaHpiDimiTestErrCodeT enumeration: 0 = No Error 1 = Run-time Error. 2 = Undefined Error. All other values are reserved
22	1	*Text String Flags*. Bit 7 = 1 - the following string is a URI to a file containing the test results Bits 6:4 – reserved Bits 3:0 – the number of following IPMI text strings that contain or point to the test results
23	N	*Text Strings*. Sequence of zero or more IPMI text strings representing the HPI result string

If the *Completion Code* is not IPMI_SUCCESS, the rest of the response may not be returned.

6.2.20 AXIe Allow Enable PCIe Hosts

This Pigeon Point extension command can be used by the System Manager to allow the Shelf Manager to send the AXIe-specific IPMI command "Set PCIe Host State (enable)" to AXIe modules E-keyed as PCIe hosts. When the Shelf Manager receives this command with Byte 4 = 1, the Shelf Manager transitions from the AXIe sequencing state **`Wait for System Manager Ready`** to the state **`Sending Set PCIe Host State`** (see section 3.8 for details).

Network Function Code (NetFN): 2Eh

Command Code: 53h

Table 82 AXIe Allow Enable PCIe Hosts

	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*. 0 = Do not allow the Shelf Manager to send Set PCIe Host State 1 = Allow the Shelf Manager to send Set PCIe Host State Other values are reserved
Response Data	1	*Completion Code.*
	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.

6.2.21 CPLD/HWRI State Sensor

This discrete sensor has sensor type DEh, event/reading type 6Fh (sensor-specific discrete) and is implemented on LUN 0, number 128 of the physical ShMM IPM controller (typically 10h/12h). It exposes the current CPLD (ShMM-500/ShMM-1500 only) and hardware redundancy state of the ShMM. On the ShMM-500/ShMM-1500, the sensor name is “CPLD State”. On the ShMM-700, where there is no CPLD and the programmatic interface to HRI is different, this sensor has the name “HWRI State”, though it is fully state-compatible with its counterpart on the ShMM-500/1500.

The following offsets are defined for the sensor:

High-level redundancy state offsets:
0 – The current Shelf Manager is Active with no Backup
1 – The current Shelf Manager is Active, with a Backup
2 – The current Shelf Manager is a Backup.

Low-level exception state bits (should be treated as low-level errors if set):
4 – The Shelf Manager is a Backup but the remote presence bit is not set
5 - The Shelf Manager is a Backup but the remote switchover request bit (the remote active bit on ShMM-700) is not set
6 - The Shelf Manager is a Backup but the HRI Active bit is set
7 - The Shelf Manager is Active with a Backup but the remote presence bit is not set
8 – The Shelf Manager is Active with a Backup but the remote healthy bit is not set
9 - The Shelf Manager is Active with a Backup but the HRI Active bit is not set
10 - The local presence bit is not set for the current Shelf Manager
11 - The Shelf Manager is Active with no Backup, but the remote healthy bit is set
12 - The Shelf Manager is Active with no Backup but the remote switchover request bit (the remote active bit on ShMM-700) is set
13 - The Shelf Manager is Active but senses the HRI Active bit set on other ShMM.

6.2.22 Reboot Reason Sensor

This discrete sensor has sensor type DDh, event/reading type 6Fh (sensor-specific discrete) and is implemented on LUN 0, number 129 of the physical ShMM IPM controller (typically 10h/12h). It exposes the reason for the last reboot of the ShMM and can be used for diagnostics of spontaneous ShMM reboots. The sensor name is “Reboot Reason”. The following offsets are defined for the sensor:

0 – The reboot reason is unknown
1 - The reboot was caused by a switchover operation
2 - The reboot was caused by a forced switchover operation
3 - The reboot was caused by the CLI command **`terminate`**
4 - The reboot was caused by loss of the HEALTHY bit
5 - The reboot was caused by loss of the ACTIVE bit
6 - The reboot of the Backup ShMM happened because the redundancy connection was broken but the Active ShMM was still alive
7 - The reboot happened due to an error during the Shelf Manager startup
8 - The reboot was caused by the ShMM hardware watchdog
9 - The reboot was initiated by software (a **`reboot()`** system call)
10 - The ShMM has been power cycled
11 - The reboot happened because of an SDRAM error
12 - The reboot was caused by the IPMI Cold Reset command.

6.2.23 HPI System Event Sensor

This discrete sensor has sensor type DBh, event/reading type 6Fh (sensor-specific discrete) and is implemented on LUN 3, number 0 of the logical Shelf Manager IPM controller (20h). The sensor name is “HPI Sys Event”. Its purpose is to enhance interaction between the Shelf Manager and Pigeon Point HPI implementations: IntegralHPI and Pigeon Point OpenHPI. Since this sensor is intended to be used for internal communication between the Shelf Manager and Pigeon Point HPI implementation, its behavior, set of states and event format are subject to change. Direct user access to this sensor is not recommended.

6.2.24 Fan Tray Operational Status Sensor

This discrete sensor has sensor type 28h (“Management Subsystem Health”), event/reading type Bh (“Redundancy States”) and is implemented on LUN 0, number 135 of the logical Shelf Manager IPM controller (20h). The sensor name is “Ft Oper. Status”. This sensor indicates whether all of the fan trays defined in the Shelf Address Table are present and operational. See the *Shelf Manager User Guide* for more information about this sensor.

6.2.25 Cooling State Sensor

This discrete sensor has sensor type 28h (“Management Subsystem Health”), event/reading type 7h (“Severity”) and is implemented on LUN 0, number 136 of the logical Shelf Manager IPM controller (20h). The sensor name is “Cooling State”. This sensor exposes the current cooling state of the shelf. See the *Shelf Manager User Guide* for more information about this sensor.

6.2.26 Fans State Sensor

This discrete sensor has sensor type 28h (“Management Subsystem Health”), event/reading type 7h (“Severity”) and is implemented on LUN 0, number 137 of the logical Shelf Manager IPM controller (20h). The sensor name is “Fans State”. This sensor exposes the current state of fans in the shelf. See the *Shelf Manager User Guide* for more information about this sensor.

6.2.27 Shelf Manager Redundancy Sensor

This discrete sensor has sensor type 28h (“Management Subsystem Health”), event/reading type Bh (“Redundancy States”) and is implemented on LUN 0, number 138 of the logical Shelf Manager IPM controller (20h). The sensor name is “SHM Redundancy”. This sensor exposes the current Shelf Manager redundancy state. The following offsets are defined for the sensor:
0 – Fully Redundant: Both Active and Backup Shelf Managers are operational and form a redundant pair
1 – Redundancy Lost: Only Active Shelf Manager is operational
2 – Redundancy Degraded: Backup Shelf Manager is operational but cannot take an Active role in this state.

6.3 Deactivation Scenarios for the Shelf Manager

The ATCA command “Set FRU Activation (Deactivate)” can be applied to the IPM controllers representing the active Shelf Manager – both the logical Shelf Manager (IPMB address 0x20, FRU 0) and the physical Shelf Manager IPM controller (with IPMB address derived from the hardware address of the active Shelf Manager). As a result of this command, the corresponding IPM controllers are deactivated (brought to state M1) but Shelf Manager functional operation is not affected and no switchover to the backup Shelf Manager is initiated.

Another possibility for the deactivation of the physical IPM controller on the active Shelf Manager is opening the Hot Swap handle on the ShMC. In that case, the physical IPM controller transitions to the state M1 as well.

Starting from version 2.5.0, in both of these cases a switchover to the backup Shelf Manager occurs, if the backup Shelf Manager exists and the configuration variable **`SWITCHOVER_ON_HANDLE_OPEN`** is set to **`TRUE`** in the Shelf Manager configuration file. Previously, a switchover took place only in the case of deactivation via the Hot Swap handle.

The backup Shelf Manager is not considered available for switchover if it is in state M1 (either due to the handle being opened or due to programmatic deactivation). In this situation, programmatic requests for switchover fail; a switchover only takes place if the active Shelf Manager is physically removed from the shelf.

The scenarios above are not explicitly covered in the AdvancedTCA (PICMG 3.0) specification. The implementation described above became possible only after the deactivation of the physical IPM controller of the active Shelf Manager was excluded from the shelf-wide power down and power cycle operations, that could happen during the normal operation of the shelf, for example in the following cases:

- powering off all FRUs in the shelf as a result of a critical temperature alert;

- receiving the IPMI command “Chassis Control” with the control option “Power Down” or “Power Cycle”.

Performing a switchover in these cases is undesirable and may be even impossible (because the backup Shelf Manager may also be in state M1).

However, starting from release 2.4.1, dedicated Shelf Manager FRUs are not affected by the shelf-wide power down operations mentioned above.

7 Pigeon Point OEM Records in FRU Information

The Shelf Manager recognizes several Pigeon Point OEM records that can be located in the Multirecord area of the Shelf FRU Information and Board FRU Information for the ShMM carrier. The format of these OEM-specific records is similar to the format of PICMG-defined records, except that they use the Manufacturer ID assigned to Pigeon Point Systems (00400Ah), while PICMG-defined records use the PICMG Manufacturer ID (00315Ah).

The following table lists Pigeon Point OEM-specific records recognized by the Shelf Manager.

Table 83 Pigeon Point OEM-specific Records in FRU Information

RECORD NAME	PIGEON POINT RECORD ID	DESCRIPTION	LOCATION
Air Filter Replacement Date	03h	Contains the dates of the last and the next replacement of the shelf air filter	Shelf FRU Information
Shelf FRU Update	04h	Contains the date of the last update of the Shelf FRU Information	Shelf FRU Information
IPMB Topology	05h	Contains information about the IPMB topology supported by the ShMM Carrier (bused or radial)	ShMM Carrier FRU Information
Slot Power Capabilities	0Bh	Identifies power limits for slots based on their non-cooling related physical characteristics	Shelf FRU Information

7.1 Air Filter Replacement Date Record

This record is only supported on specific ShMM carriers. It holds the date when the air filter was replaced and the date when the air filter should be replaced again.

The Air Filter Replacement Date record has the following format:

Table 84 Air Filter Replacement Date Record

OFFSET	LENGTH	DESCRIPTION
0	1	Record Type ID. Value C0h (OEM).
1	1	[7:7] End of list. Set to one for the last record. [6:4] Reserved, write as 0h. [3:0] Record format version (=2h for this

		definition).
2	1	Record Length.
3	1	Record Checksum. Holds the zero checksum of the record.
4	1	Header Checksum. Holds the zero checksum of the header.
5	3	Manufacturer ID. LS Byte first. Write as the three byte ID assigned to PPS (0Ah 40h 00h).
8	1	Pigeon Point Record ID. The value 3h is used for this definition.
9	1	Record Format Version. Write as 0h.
10	1	Type/Length byte(FRU Information variant) for the following date: [6:7] Type (=11b for this definition), [0:5] Length (=19 or 13h for this definition).
11	19	8-bit ASCII string representation that specifies the date when the air filter was replaced. The date format is 'hh:mm:ss dd.mm.yyyy'.
30	1	Type/Length byte(FRU Information variant) for the following date: [6:7] Type (=11b for this definition), [0:5] Length (=19 or 13h for this definition)
31	19	8-bit ASCII string representation that specifies the date when the air filter should be replaced. The date format is 'hh:mm:ss dd.mm.yyyy'.

7.2 Shelf FRU Update Record

The Shelf FRU Update record stores the date of the latest Shelf FRU Information update and has the following format:

Table 85 Shelf FRU Update Record

Offset	Length	Description
0	1	Record Type ID. Value C0h (OEM).
1	1	[7:7] End of list. Set to one for the last record [6:4] Reserved, write as 0h. [3:0] Record format version (=2h for this definition).
2	1	Record Length.

3	1	Record Checksum. Holds the zero checksum of the record.
4	1	Header Checksum. Holds the zero checksum of the header.
5	3	Manufacturer ID. LS Byte first. Write as the three byte ID assigned to PPS (0Ah 40h 00h).
8	1	Pigeon Point Record ID. The value 4h is used for this definition.
9	1	Record Format Version. Write as 0h.
10	1	Type/Length byte(FRU Information variant): [6:7] Type (=11b for this definition) [0:5] Length (=19 or 13h for this definition).
11	19	8-bit ASCII string representation that specifies the date when the Shelf FRU Info was updated. The date format is 'hh:mm:ss dd.mm.yyyy'.

7.3 IPMB Topology Record

The IPMB Topology record contains the Carrier Type, which indicates the IPMB-0 topology implemented by the ShMM carrier; the following format is used:

Table 86 IPMB Topology Record

Offset	Length	Description
0	1	Record Type ID. Value C0h (OEM).
1	1	[7:7] End of list. Set to one for the last record [6:4] Reserved, write as 0h. [3:0] Record format version (=2h for this definition).
2	1	Record Length.
3	1	Record Checksum. Holds the zero checksum of the record.
4	1	Header Checksum. Holds the zero checksum of the header.
5	3	Manufacturer ID. LS Byte first. Write as the three byte ID assigned to PPS (0Ah 40h 00h).
8	1	Pigeon Point Record ID. The value 5 is used for this definition.
9	1	Record Format Version. Write as 0h.
10	1	Carrier Type:

		1 – Bused Carrier 2 – Dual Star Carrier 3 – Redundant Dual Star Carrier 4 – Pseudo-Radial Carrier.

Carrier types defined in this record specify the IPMB topology that the carrier supports:

- Bused carriers support bused IPMB topology, where all IPM Controllers, including the Shelf Manager(s), are attached to a single multi-drop IPMB. With this topology, the Shelf Manager(s) connect to IPMB in the same way as other IPM Controllers in the shelf. All IPMB-0 traffic is visible to all nodes on the single logical IPMB-0.
- Dual Star carriers support a logical star, or radial, IPMB topology, where each IPM Controller in the shelf is connected to the Shelf Manager via a logically distinct point-to-point IPMB link. Each of the two redundant Shelf Managers implements one radial connection, so that for any IPM Controller its IPMB-A link is connected to one Shelf Manager and its IPMB-B link is connected to the other Shelf Manager. With this topology, the simultaneous presence of both redundant Shelf Managers is required for redundant operation of IPMB in accordance with the AdvancedTCA requirements. Each IPMB link is independent of the others, enabling reliability and performance benefits.
- Redundant Dual Star carriers also support a logical star or radial IPMB topology. The difference from the Dual Star carrier type is that each of the redundant Shelf Managers is connected to each IPM Controller via *both* IPMB-A and IPMB-B links. With this topology, both IPMB-A and IPMB-B are operational for an IPM Controller even if only a single Shelf Manager is installed in the shelf. In this topology also, each IPMB link is independent of the others, enabling reliability and performance benefits.
- Pseudo-Radial carriers support a single logical IPMB-0 topology, but provide additional controls that allow attach specific IPM Controllers to IPMB-0 and detach them from it. In this topology, as with the bused carrier, all IPMB-0 traffic is visible to all nodes that are attached to the single logical bus. This topology delivers some of the advantages of a radial topology. For example, a malfunctioning IPM Controller that violates bus integrity can be detached from the bus and allow other IPM Controllers to resume normal communication.

In each of the radial or star topologies described above, one or more of the radial links may have more than one IPM Controller on the link. The PICMG-defined Radial IPMB-0 Link Mapping record defines these aspects of the overall shelf architecture. The focus of the Pigeon-Point-defined IPMB Topology record is on the topology supported by the ShMM carrier, itself.

7.4 Slot Power Capabilities Record

Generally, the Max FRU Power value specified for each slot in the PICMG Power Management Record is limited by the power which can be handled by the cooling system for the slot. For a multi-slot board, cooling applies to all the slots it occupies, so this limitation becomes less crucial for them. The power that the slot itself can physically deliver (defined by the physical characteristics of the slot, such as the backplane traces) becomes more important for such a slot, since a multi-slot board, according to ATCA rules, always get power through a single slot (the one where its IPMC takes its hardware address).

The Slot Power Capabilities OEM record identifies the power limits for slots due to their non-cooling-related physical characteristics and has the following format:

Table 87 Slot Power Capabilities Record

Offset	Length	Description
0	1	Record Type ID. Value C0h (OEM).
1	1	[7:7] End of list. Set to one for the last record [6:4] Reserved, write as 0h. [3:0] Record format version (=2h for this definition).
2	1	Record Length.
3	1	Record Checksum. Holds the zero checksum of the record.
4	1	Header Checksum. Holds the zero checksum of the header.
5	3	Manufacturer ID. LS Byte first. Write as the three byte ID assigned to PPS (0Ah 40h 00h).
8	1	Pigeon Point Record ID. The value Bh is used for this definition.
9	1	Record Format Version. Write as 0h.
10	1	Number of Slot Power Capabilities Descriptor Entries (N)
10	4 * N	Slot Power Capability Descriptor Entries

The Slot Power Capability Descriptor has the following format:

Table 88 Slot Power Capability Descriptor

Offset	Length	Description
0	1	Slot Hardware Address
1	1	Slot FRU Device ID. A value of FEh indicates that all FRU Device IDs at the Hardware Address are considered as a unit.
2	2	Maximum Slot Power Capability. This field contains the maximum wattage that can be routed to the Hardware Address/FRU Device ID location for FRUs that draw power from this location. This is stored as a number of Watts in 1 W increments, LS Byte first.

The Shelf FRU Information can contain several records of this type. The data from all such records must be merged together in order and interpreted as a single logical record.

8 Hardware Platform Interface (HPI)

The Pigeon Point Shelf Manager optionally includes IntegralHPI, an implementation of the Service Availability Forum (SAF, www.saforum.org) Hardware Platform Interface (HPI), operating as a subsystem within the Shelf Manager. IntegralHPI requires a variant of the ShMM-1500 or the ShMM-500 that has additional memory: specifically, 128 megabytes of RAM and 64 megabytes of Flash.

The SAF HPI is a System Manager interface that is more abstract than the Remote Management Control Protocol (RMCP), the mandatory System Manager interface for AdvancedTCA shelves. A separate SAF specification defines the mapping between HPI and AdvancedTCA systems. IntegralHPI complies with this HPI-to-AdvancedTCA Mapping specification.

Integral HPI implements revision B.03.02 of the base HPI specification and revision B.03.02 of the HPI-to-xTCA Mapping specification.

By default, the IntegralHPI interface in the Shelf Manager is turned off. To turn it on, it is necessary to set the configuration variable **`ENABLE_INTEGRALHPI`** to **`TRUE`**, and ensure that the corresponding shared library **`libintegralhpi.so`** (along with several other libraries that IntegralHPI depends on) is present on the ShMM.

8.1 HPI Functions Support in IntegralHPI

IntegralHPI implements the HPI functions defined in the HPI specification, as detailed in the following table. Some of the functions that are designated as optional in the HPI specification are not implemented.

Table 89 HPI Functions Implemented by IntegralHPI

Function	Support in IntegralHPI
saHpiVersionGet()	Supported
saHpiSessionOpen()	Supported
saHpiSessionClose()	Supported
saHpiDiscover()	Supported
saHpiDomainInfoGet()	Supported
saHpiDrtEntryGet()	Supported
saHpiDomainTagSet()	Supported
saHpiRptEntryGet()	Supported
saHpiRptEntryGetByResourceId()	Supported
saHpiResourceSeveritySet()	Supported
saHpiGetIdByEntityPath()	Supported
saHpiGetChildEntityPath	Supported
saHpiResourceFailedRemove()	Supported
saHpiResourceTagSet()	Supported
saHpiResourceIdGet()	Not supported
saHpiGetIdByEntityPath()	Supported

FUNCTION	SUPPORT IN INTEGRALHPI
saHpiGetChildEntityPath()	Supported
saHpiResourceFailedRemove()	Supported
saHpiEventLogInfoGet()	Supported
saHpiEventLogCapabilitiesGet()	Supported
saHpiEventLogEntryGet()	Supported
saHpiEventLogEntryAdd()	Supported
saHpiEventLogEntryClear()	Supported
saHpiEventLogTimeGet()	Supported
saHpiEventLogTimeSet()	Supported
saHpiEventLogStateGet()	Supported
saHpiEventLogStateSet()	Supported
saHpiEventLogOverflowReset()	Supported
saHpiSubscribe()	Supported
saHpiUnsubscribe()	Supported
saHpiEventGet()	Supported
saHpiEventAdd()	Supported
saHpiAlarmGetNext()	Supported
saHpiAlarmGet()	Supported
saHpiAlarmAcknowledge()	Supported
saHpiAlarmAdd()	Supported
saHpiAlarmDelete()	Supported
saHpiRdrGet()	Supported
saHpiRdrGetByInstrumentId()	Supported
saHpiRdrUpdateCounterGet()	Supported
saHpiSensorReadingGet()	Supported
saHpiSensorThresholdsGet()	Supported
saHpiSensorThresholdsSet()	Supported
saHpiSensorTypeGet()	Supported
saHpiSensorEnableGet()	Supported
saHpiSensorEnableSet()	Supported
saHpiSensorEventMasksGet()	Supported
saHpiSensorEventMasksSet()	Supported
saHpiControlTypeGet()	Supported
saHpiControlGet()	Supported
saHpiControlSet()	Supported
saHpiIdrInfoGet()	Supported
saHpiIdrAreaHeaderGet()	Supported
saHpiIdrAreaAdd()	Not supported
saHpiIdrAreaAddById()	Not supported
saHpiIdrAreaDelete()	Not supported
saHpiIdrFieldGet()	Supported
saHpiIdrFieldAdd()	Not supported
saHpiIdrFieldAddById()	Not supported

FUNCTION	SUPPORT IN INTEGRALHPI
saHpiIdrFieldSet()	Not Supported
saHpiIdrFieldDelete()	Not Supported
saHpiWatchdogTimerGet()	Supported
saHpiWatchdogTimerSet()	Supported
saHpiWatchdogTimerReset()	Supported
saHpiAnnunciatorGetNext()	Supported
saHpiAnnunciatorGet()	Supported
saHpiAnnunciatorAcknowledge()	Supported
saHpiAnnunciatorAdd()	Supported
saHpiAnnunciatorDelete()	Supported
saHpiAnnunciatorModeGet()	Supported
saHpiAnnunciatorModeSet()	Supported
saHpiDimiInfoGet()	Supported
saHpiDimiTestInfoGet()	Supported
saHpiDimiTestReadinessGet()	Supported
saHpiDimiTestStart()	Supported
saHpiDimiTestCancel()	Supported
saHpiDimiTestStatusGet()	Supported
saHpiDimiTestResultsGet()	Supported
saHpiFumiSpecInfoGet()	Supported
saHpiFumiServiceImpactGet()	Supported
saHpiFumiSourceSet()	Supported
saHpiFumiSourceInfoValidateStart()	Supported
saHpiFumiSourceInfoGet()	Supported
saHpiFumiSourceComponentInfoGe t()	Supported
saHpiFumiTargetInfoGet()	Supported
saHpiFumiTargetComponentInfoGet ()	Supported
saHpiFumiLogicalTargetInfoGet()	Supported
saHpiFumiLogicalTargetComponent InfoGet()	Supported
saHpiFumiBackupStart()	Not supported
saHpiFumiBankBootOrderSet()	Not supported
saHpiFumiBankCopyStart()	Not supported
saHpiFumiInstallStart()	Supported
saHpiFumiUpgradeStatusGet()	Supported
saHpiFumiTargetVerifyStart()	Supported
saHpiFumiTargetVerifyMainStart()	Supported
saHpiFumiUpgradeCancel()	Supported
saHpiFumiAutoRollbackDisableGet	Supported
saHpiFumiAutoRollbackDisableSet	Supported
saHpiFumiRollback()	Supported

Function	Support in IntegralHPI
saHpiFumiActivate()	Not supported
saHpiFumiActivateStart()	Supported
saHpiHotSwapPolicyCancel()	Supported
saHpiResourceActiveSet()	Supported
saHpiResourceInactiveSet()	Supported
saHpiAutoInsertTimeoutGet()	Supported
saHpiAutoInsertTimeoutSet()	Supported
saHpiAutoExtractTimeoutGet()	Supported
saHpiAutoExtractTimeoutSet()	Supported
saHpiHotSwapStateGet()	Supported
saHpiHotSwapActionRequest()	Supported
saHpiHotSwapIndicatorStateGet()	Not supported
saHpiHotSwapIndicatorStateSet()	Not supported
saHpiParmControl()	Not supported
saHpiResourceLoadIdGet()	Not supported
saHpiResourceLoadIdSet()	Not supported
saHpiResourceResetStateGet()	Supported
saHpiResourceResetStateSet()	Supported
saHpiResourcePowerStateGet()	Not supported
saHpiResourcePowerStateSet()	Not supported

8.2 HPI Mapping Sensors and Controls in IntegralHPI

The HPI-to-xTCA Mapping specification, in addition to rules for mapping HPI functionality to AdvancedTCA shelves, also defines certain HPI instruments that must be included in conforming implementations. The table below lists these instruments, as they are implemented in IntegralHPI.

Table 90 Standard Instruments Implemented by IntegralHPI

Instrument Name	Owner Resource	Instrument Number
IPMI Sensors	Virtual Shelf Manager, ATCA FRU	(1024 * IPMI Sensor LUN) + IPMI Sensor Number
E-Keying Link State Sensor	Virtual Shelf Manager, ATCA IPMC	3584..3839
AMC Clock E-Keying Link State Sensor	AMC Module	3824..3828
Bused E-Keying Link State Sensor	Shelf	3824..3828
Shelf FRU Information Valid Sensor	Shelf	4096
Shelf Manager Redundancy Sensor	Virtual Shelf Manager	4097
Active Shelf Manager Sensor	Virtual Shelf Manager	4098
Standby Shelf Manager Sensor	Virtual Shelf Manager	4099
Assigned Power Sensor	Slot	4113
IPMB-0 Sensor	Virtual Shelf Manager, ATCA IPMC	4352..4447

Instrument Name	Owner Resource	Instrument Number
LED Controls	ATCA FRU	0.. 255
Shelf Address Control	Shelf	4096
Shelf Manager IP Address Control	Shelf	4097
Chassis Status Control	Shelf	4098
Shelf Manager Failover Control	Virtual Shelf Manager	4112
FRU Activation Control	Slot	4128
Desired Power Control	Virtual Shelf Manager, ATCA FRU	4144
IPMB-0 State Control	Virtual Shelf Manager, ATCA IPMC	4353, 4354
FRU Reboot and Diagnostics Control	Virtual Shelf Manager, ATCA FRU	4608
IPM Controller Reset Control	Virtual Shelf Manager, ATCA IPMC	4609
Fan Control	ATCA FRU	5120
IPMI Watchdog	Virtual Shelf Manager ATCA IPMC	0
ShelfFRU Information Inventory	Shelf	0
FRU Information Inventory	ATCA FRU	0
Configuration Data Inventory	Shelf	1
Telco Alarm Annunciator	Shelf	0
Telco Alarm Location Sensor	Virtual Telco Alarm	5632
Telco Alarm Input Sensor	Virtual Telco Alarm	5633
Telco Alarm Controls: Critical Alarm, Major Alarm, Minor Alarm, Power Alarm and Alarm Cutoff	Virtual Telco Alarm	5632..5636
HPM.1 FUMI	ATCA FRU	0
Global IPMC Upgrade Capabilities Sensor	ATCA FRU	5888
Upgrade Image Capabilities Sensor	ATCA FRU	5889
Rollback Timeout Period Sensor	ATCA FRU	5890

IntegralHPI also implements a proprietary PPS HPI version control to provide an HPI user with the version of supported interface and the version of IntegralHPI service.

Table 91 Additional Instruments Implemented by IntegralHPI

Instrument Name	Owner Resource	Instrument Number
PPS HPI Version Control (PPS proprietary)	Shelf	32768
IPMI Self Test DIMI (PPS proprietary)	ATCA IPMC	32768
IPMI Command Control (PPS proprietary)	Shelf	32767
AMC IPMI Command Control (PPS proprietary)	Shelf	32766

Instrument Name	Owner Resource	Instrument Number
IPMB-0 Address Sensor (PPS proprietary)	ATCA FRU	32768
FRU Device Id Sensor (PPS proprietary)	ATCA FRU	32766
IPMB-L Address Sensor (PPS proprietary)	AMC Module	32767
DI DIMI	ATCA/AMC FRU	DI ID as defined in Static DI Data.
ShMM FUMI	Physical ShMM	1

8.3 Redundancy in IntegralHPI

IntegralHPI, as part of a Shelf Manager that natively supports redundancy, also supports redundant operation. The following IntegralHPI data items are maintained as redundant; redundancy is achieved using the Shelf Manager’s built-in redundancy facility:

- HPI Session Information: Session Id and HPI User IP Address, Session Subscription Data
- Domain Tag
- Domain Auto-Insertion Timeout
- Resource Tree
- Domain Event Log State
- Domain Event Log Time
- Domain Event Log Content
- Domain Event Log Overflow Time
- Domain Alarm Table Content
- Resource Tag
- Resource Severity
- Resource Auto-Extraction Timeout
- Resource Hot Swap Policy cancellation
- Resource Hot Swap Policy Execution time
- FRU Power On Sequence Control State
- FRU Power On Sequence Commit Status Sensor Event State
- AMC Power On Sequence Control State
- AMC Power On Sequence Commit Status Sensor Event State
- FRU Power On Sequence Commit Status Sensor Status and Event Masks
- AMC Power On Sequence Commit Status Sensor Status and Event Masks
- Slot State Status Sensor Status and Event Masks
- Telco Alarm Annunciator Announcements
- HPM.1 FUMI upgrade image URI
- HPM.1 FUMI upgrade status

Status and event masks for IPMI sensors, content of FRU Information areas, as well as sensors and controls that are based on Shelf FRU Information and IPM Controller FRU Information are redundant by design; no special mechanism is needed to make them redundancy-aware.

The standby IntegralHPI service instance does not perform hot swap management of the FRUs and does not provide HPI service until it become active.

9 Revision History

This section records the major revisions in this document, starting with release 2.1.0 of the Shelf Manager.

9.1 Release 2.1.0

- Section 3.2: adds CLI commands **gethysteresis**, **getipmbstate** to the table that summarizes the CLI commands.
- Section 3.29: adds the description of the CLI command **gethysteresis**.
- Section 3.30: adds the description of the CLI commands **getipmbstate**.
- Section 3.56: adds the description of the CLI command **sethysteresis**.
- Section 3.57: adds the description of the CLI command **setipmbstate**.
- Section 4: adds descriptions of Web interface for the CLI commands **gethysteresis**, **sethysteresis**.

9.2 Release 2.2.0

- Section 3.2: augments the table that summarizes the CLI commands to identify those that are available on the Backup Shelf Manager.
- Section 3.28: adds the description of the CLI command **getfruledstate**.
- Section 3.40: adds the description of the CLI command **poll**.
- Section 3.54: adds the description of the CLI command **setfruledstate**.
- Section 3.61: adds the description of the CLI command **setpowerlevel**.
- Section 5.1.1: modifies the access mode of the MIB variable **board-basic-powered** from ‘read-only’ to ‘read-write’.

9.3 Release 2.3.0

- Overall: implements a change in Shelf Manager product name from ‘IPM Sentry’ to ‘Pigeon Point’.
- Section 3.2: in the table that summarizes the CLI commands, augments the commands available on the Backup Shelf Manager.
- Sections 3.6, 3.23: introduces a new option **info** for CLI commands **alarm** and **frucontrol**.
- Section 3.36: the CLI command **ipmc** shows information about FRUs in the state M1, if they were known previously to the Shelf Manager. Before the version 2.3, information about such FRUs was not shown by this command.
- Section 3.44: introduces the CLI command **sendcmd**.
- Section 3.46: adds the option **-t** for the CLI command **sensordata**. If the option **-t**, is specified, information is displayed only for threshold-based sensors that have at least one of their thresholds crossed.
- Section 3.65.3: adds the option **-v** (verbosity) for the CLI command **shelf power_management**.

- Section 4: adds descriptions of Web interface for the CLI commands **alarm**, **getfruledstate**, **getipmbstate**, **getsensoreventenable**, **session**, **setfruledstate**, **setipmbstate**, **setsensoreventenable**.
- Section 4.33: updates the Web interface for the CLI command **sel**.
- Section 5.1.4: modifies the descriptions of the MIB variables **shelf-manager-active** and **shelf-manager-reset**.
- Section 6: adds a table of IPMI commands implemented by the Shelf Manager. This table specifies whether a command is supported if it arrives from RMCP interface or from an IPM controller.
- Section 6.2.6: corrects the command code for the extension command "Set/Clear Digital Outputs" to D2h.
- Section 6.2.15: describes the deactivation scenarios for the active Shelf Manager.

9.4 Release 2.4.0

- Sections 3.24, 3.27, 3.43, 3.49, 3.53: includes descriptions for CLI commands **getbootdev**, **getfanpolicy**, **sendamc**, **setbootdev**, **setfanpolicy**.
- Section 3.54: corrects the description of the CLI command **setfruledstate**.
- Section 3.66: updates the **shelfaddress** command description to reflect its use of full ASCII rather than packed ASCII when encoding the characters of a new shelf address string.
- Section 4.24: corrects the description of the Web interface for the CLI command **setfruledstate**.
- Section 6.2: adds the description of the extension commands "Notify Shelf Manager About an Extracted FRU", "Initiate Shelf Manager Switchover", "Subscribe for Event Notifications", "Get Shelf FRU Record Data".

9.5 Release 2.4.1

- Section 3.5: adds a description of the CLI command **airfilterreplaced**.
- Section 3.72: changes the CLI **version** command output to reflect carrier subtype and version.
- Section 4.35: changes the Web **version** command output to reflect carrier subtype and version.
- Section 5.1.3: changes the description of the SNMP MIB variable **powersupply-fail** (see Table 17) to clarify that this variable is only available in PICMG 2.x systems.
- Section 6.2.9: changes the name of the last bit in "Initiate Shelf Manager Switchover" command (see Table 50) to "Active Reboot Mode". Note that this change reverses the polarity of this bit.

9.6 Release 2.4.2

- Section 3.5: modifies the description of the CLI command **airfilterreplaced** to reflect the fact that an IPMI event is no longer automatically generated when the expiration date is reached or passed. Another modification reflects the usage of a Pigeon Point defined multirecord to store the dates associated with air filter replacements.

- Section 3.66: adds a description of the option **-x** to the CLI command **shelfaddress**. This option allows a user to specify the shelf address as a sequence of hexadecimal bytes.

9.7 Release 2.4.3

- Section 3.40: adds a description of the CLI command **networkelementid**.
- Section 3.4.2: changes the description of the CLI command **activate** to indicate that the IPMI command “FRU Activation Policy (Clear Locked)” is also sent.
- Section 6.2.11: adds a description of the Pigeon Point extension command “Set Shelf FRU Record Data“.

9.8 Release 2.4.4

- Section 3.13: changes the description of the CLI command **debuglevel** to reflect that separate debug levels are now supported for system log output and for console output.
- Section 3.35: modifies the **clia shelf help** example to reflect updates in the help text.

9.9 Release 2.5.0

- Sections 3.4, 3.12, 3.16, 3.18, 3.19, 3.20, 3.21, 3.22, 3.23, 3.25, 3.28, 3.51, 3.54, 3.59: changes the syntax for the following CLI commands: **activate**, **deactivate**, **fans**, **fru**, **frudata**, **frudatar**, **frudataw**, **frucontrol**, **getfanlevel**, **getfruledstate**, **setextracted**, **setfruledstate**, **setlocked**. These commands can now address a specific AMC by its number.
- Section 3.7: Adds the description of the new CLI command **amcportstate**.
- Section 3.12: changes the description of the CLI command **deactivate** to reflect that programmatic deactivation of the active Shelf Manager now causes a switchover.
- Section 3.23: adds clarification how the command “FRU Control (Cold Reset)” is implemented on the Shelf Manager.
- Sections 3.29, 3.33, 3.34, 3.45, 3.46: adds the description of the option **–f <fru_id>** to sensor-related CLI commands **gethysteresis**, **getsensoreventenable**, **getthreshold**, **sensor**, **sensordata**. This option enables the user to select sensors that belong to a specified FRU.
- Section 3.62: adds substantial detail about the parameters of the command **setsensoreventenable**.
- Section 3.69: adds a description of the forced switchover (option **–forced**).
- Section 4.11: adds a description of the new Web interface command “Get Pigeon Point MIB Files”.
- Section 5.1.2: adds a new read-write MIB variable **fantray-fan-level**. Retrieving and setting of fan levels can be accomplished via this variable.
- Section 5.1.3: corrects the description of **powersupply-slot-number** variable.
- Section 5.1.7: adds new MIB variables: **rmcp-interface-status**, **shelf-fru-found-status**, **active-status**. These variables report the Shelf Manager status.
- Section 5.2.10: corrects the OID example for FRU Information MIB variables.
- Section 6.2.15: reflects substantial changes due to the fact that programmatic deactivation of the active Shelf Manager now causes a switchover.

9.10 Release 2.5.2

- Section 3.4: adds information about storing, replacing and deleting HPDL binary data and SDRs in FRU Information
- New section 3.32.19: describes the new parameter **`pet_format`** for the command **`getpefconfig`**.
- Section 3.36: adds coverage of the additional three-part representation for the firmware revision.
- Section 3.39: adds information about the command **`minfanlevel`** in shelves with zoned cooling.
- Section 3.53: clarifies the purpose of the command **`setfanpolicy`**.
- New section 3.60.14: describes the new parameter **`pet_format`** for the command **`setpefconfig`**.
- New section 3.62: describes the command **`setsensordata`**.
- Section 3.65.3: adds information about the message that is displayed when some fan trays are not operational.
- Section 3.65.6: clarifies the description of the **`shelf activation`** command.
- Section 3.65.7: clarifies the description of the **`shelf deactivation`** command, corrects the example of the execution of the command.
- Section 4.11: changes the web "Get Pigeon Point MIB Files" command. A user can choose either of two Pigeon Point MIB files: PPS-SENTRY-MIB or PPS-PET-MIB.
- New section 5.1.8: describes the SNMP variables that return the version of the Shelf Manager.
- New section 5.1.9: describes the SNMP variables that access TELCO alarms.
- New sections 6.2.12, 6.2.13, 6.2.14: describe Pigeon Point extension commands to access the cached device SDRs in the Shelf Manager.

9.11 Release 2.5.3

- Section 4.31: changed the request web page for the "Shelf Information" command to reflect an implementation modification.

9.12 Release 2.6.0

- New section 3.17: describes the CLI command **`firewall`**.
- Section 3.31: describes how to use CLI command **`getlanconfig`** for LAN Configuration variables supported by IPMI specification version 2.0.
- Section 3.46: adds a description of the new option **`-d`** for the command **`sensordata`**.
- New section 3.50: describes the CLI command **`setcommandpolicy`**.
- New section 3.55: describes the CLI command **`setfunctionpolicy`**.
- Section 3.58: describes how to use CLI command **`setlanconfig`** for LAN Configuration variables supported by IPMI specification version 2.0.
- Section 4: updates the Web interface command screen shots to reflect the current implementation, which includes the formally registered stylized lighthouse logo.
- Section 5.2.8: describes SNMP LAN Configuration variables supported by IPMI specification version 2.0.

- Section 6.1: adds IPMI 2.0 commands and PICMG 3.0 R3.0 commands to the table of IPMI commands implemented by the Shelf Manager.
- New section 6.2.15: describes the Pigeon Point extension command to set FRU activation parameters for a slot.

9.13 Release 2.6.1

- Section 3.16: excludes the obsolete “maximum sustained fan level” from the list of data items reported by this command.
- Section 5.1.1: adds descriptions of the MIB variables **`board-basic-fruinfo-board-serial-number`** and **`board-basic-fruinfo-board-manufacture-time`**.
- Section 5.1.6: clarifies the description of the SNMP variable **`event-class`**.
- Section 5.1.8: adds a description of the SNMP variable **`functional-level`**.
- New section 5.1.10: describes the SNMP variables that contain information about the SEL state.
- New section 5.1.11: introduces carrier-specific SNMP variables.
- Section 5.2.9: corrects the description of the SNMP PEF Configuration variables in Table 36 and Table 37.
- Section 6.2.4: corrects the description of the “Query Digital Output Properties” command.
- Section 6.2.5: corrects the description of the “Get Digital Outputs” command.
- Section 6.2.6: corrects the description of the “Set/Clear Digital Outputs” command.
- New section 7: describes the IntegralHPI interface.

9.14 Release 2.6.4

- Section 3.21: adds descriptions of the new options **`–p`** and **`–l`** for the CLI command **`frudataw`**.
- Section 3.28: updates the description of the output of the CLI command **`getfruledstate`**.
- New section 3.25: describes the CLI command **`getconfigparam`**.
- Section 3.35: updates the examples to cover new commands.
- Section 3.39: updates the description of the output of the CLI command **`minfanlevel`**.
- Section 3.48: adds a description of the new option **`–v`** for the CLI command **`session`**.
- Section 3.72: updates the description of the output of the CLI command **`version`**.
- Session 3.65: adds descriptions of the new subcommands **`shm_cfg_params`** and **`board_lan_cfg_params`** of the CLI command **`shelf`**.
- Session 3.72: updates the description of the output of the CLI command **`version`**.
- New section 5.2.5: describes SNMP variables for FRU LED management.
- Section 6: adds a note about RMCP+ on ShMMs with encryption code removed.
- New section 6.1.1: describes the Shelf Manager response to the command “Get Device ID”.
- New section 6.1.2: describes the Shelf Manager response to the command “Get Self Test Results”.
- New section 6.2.16: describes the Pigeon Point extension command to set minimal fan level.
- New section 6.2.17: describes the Pigeon Point extension command to get minimal fan level.

- Section 8.2: merges the tables of sensors and controls and adds descriptions of the new HPI instruments.

9.15 Release 2.6.4.2

- New section 3.14: describes the CLI command **`dhcp restart | status`**.
- New section 6.2.18: describes the Pigeon Point extension command to control and check status of the DHCP client.

9.16 Release 2.7.0

- Sections 3.2, 3.35.3, 3.43, 3.44: the CLI commands **`sendamc`** and **`sendcmd`** now accept a **`<lun>`** parameter.
- Section 3.34.2: sensor upper thresholds in raw format are shown as lower thresholds in processed format when the sensor has a decreasing linearization function.
- Section 3.46.2: a processed value is no longer shown by the CLI command **`sensordata`** if the sensor is in the state "Initial Update In Progress/Sensor Reading Unavailable".
- Sections 3.65.4.2, 3.65.5: FRUs may now be deactivated if the currently used power exceeds the available power provided by the feed (as a result of execution of either the CLI command **`shelf maxcurrent`** or the CLI command **`shelf minvoltage`**).
- Section 3.65.9.2: FRUs may now be deactivated if the currently used power for a slot exceeds the power limit for that slot (as a result of execution of the CLI command **`shelf pwrcapability`**).
- Section 7: IntegralHPI now supports the SAI-HPI-B.03.02 and SAIM-HPI-xTCA-B.03.02 specification versions.
- Section 7.1: covers the new SAI-HPI-B.03.02 functions and additional SAI-HPI-B.02 functions that are now supported (The latter were not supported in previous IntegralHPI releases.)
- Section 7.2: covers SAIM-HPI-xTCA-B.03.02 instruments.
- Section 7.3: covers SAIM-HPI-xTCA-B.03.02 instruments.

9.17 Release 2.7.1

- Section 3.3: modified to indicate that the **`power_supply`** notation is now also supported in ATCA systems.
- New section 3.36: covers new HPI-related CLI commands.
- Section 3.37: introduces verbose extended mode for the CLI command **`ipmc`**; also this command now shows HPM.1 component information in verbose mode.
- Section 5.1.3: the table "Power Supply Variables" is now obsolete in non-2.x contexts.
- New section 5.1.12: describes the table "xTCA PEM Variables".
- New section 5.1.13: describes the table "xTCA Power Supply Variables".
- Sections 6.2.12, 6.2.13, 6.2.14: corrects command descriptions to include PPS IANA bytes.
- Sections 8.1, 8.2: updated the tables to reflect better support of the new version of the HPI to xTCA mapping specification (B.03.02) in IntegralHPI.

9.18 Release 2.7.2

- Section 3.45: modified to indicate that the CLI command **`sensor`** now also shows information about event-only sensors.
- Section 5.2.3: modified to indicate that event-only (type 3) SDRs are now accessible via SNMP.

9.19 Release 2.7.3

- Section 5.2.8: modified to indicate that two LAN IPMI channels are now supported.

9.20 Release 2.7.4

- Section 3.40: modified to indicate that three Network Element Identifiers are now supported.
- New section 7: describes Pigeon Point OEM-specific FRU Information records.

9.21 Release 2.8.0

- Section 3.46: added an example for the ShMM Reboot Reason sensor.
- Section 6.1: modified to indicate that HPM.1 IPMI commands are now supported by the Shelf Manager.
- Section 6.2.15: correction: this command does not update Shelf FRU Information.
- New section 6.2.19: describes Diagnostics Initiator IPMI commands.
- Section 8.2: defines HPI instrument numbers for DI DIMI and ShMM FUMI.

9.22 Release 2.8.2

- Section 5.2.3: adds descriptions of the MIB variables **`sensor-processed-unr-threshold`**, **`sensor-processed-uc-threshold`**, **`sensor-processed-unc-threshold`**, **`sensor-processed-lnr-threshold`**, **`sensor-processed-lc-threshold`**, **`sensor-processed-lc-threshold`**.

9.23 Release 3.0.0

- Section 3.72: introduces a new CLI command **`variable`**.
- Section 5.1.2: adds descriptions of the MIB variables **`board-basic-fruinfo-product-asset-tag`**, **`board-basic-fruinfo-product-fru-file-id`**, **`board-basic-fruinfo-board-fru-file-id`**.
- Section 5.1.3: adds descriptions of the MIB variables **`fantray-fruinfo-product-asset-tag`**, **`fantray-fruinfo-product-fru-file-id`**, **`fantray-ps-fruinfo-board-fru-file-id`**.
- Section 5.1.4: adds descriptions of the MIB variables **`shelf-manager-fruinfo-product-asset-tag`**, **`shelf-manager-fruinfo-product-fru-file-id`**, **`shelf-manager-fruinfo-board-fru-file-id`**.
- Section 5.1.5: adds descriptions of the MIB variables **`chassis-product-asset-tag`**, **`chassis-product-fru-file-id`**, **`chassis-board-fru-file-id`**.

- Section 5.1.12: adds descriptions of the MIB variables **`xtca-pem-fruinfo-product-asset-tag`**, **`xtca-pem-fruinfo-product-fru-file-id`**, **`xtca-pem-fruinfo-board-fru-file-id`**.
- Section 5.1.13: adds descriptions of the MIB variables **`xtca-ps-fruinfo-product-asset-tag`**, **`xtca-ps-fruinfo-product-fru-file-id`**, **`xtca-ps-fruinfo-board-fru-file-id`**.

9.24 Release 3.2.0

- New section 3.8. describes the **`axie sequencing`** command.
- Section 3.22: clarifies usage of the command options **`–v`** and **`-x`**.
- Section 3.73: adds IntegralHPI spec version to the command output.
- New section 6.2.20: describes a new PPS IPMI command “AXIe Allow Enable PCIe Hosts”.
- New sections 6.2.21-6.2.27: describe PPS-specific sensors exposed by the logical Shelf Manager and by the physical ShMM IPM controller.