Pigeon Point Shelf Manager
External Interface Reference

PIGEON POINT SYSTEMS™

Release 2.3
July 28, 2006

Pigeon Point Shelf Manager and ShMM-300/ShMM-500

Table of Contents

Chapter

1

1. 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 provide the details of implementation for each of these interfaces.

The Pigeon Point Shelf Manager User Guide provides 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-300 and ShMM-500. References to the ShMM-500 include the ShMM-500R, which complies with the Restriction of Hazardous Substances (RoHS) directive, but is software equivalent from a Shelf Manager perspective.

Chapter

2

2. Command Line Interface

The Command Line Interface (CLI) can be used to communicate with the intelligent management controllers of the shelf, with boards, and with the Shelf Manager itself, 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.

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

#
```

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

```
# clia

Pigeon Point Shelf Manager Command Line Interpreter

CLI> ipmc 20

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


CLI> exit
#
```

2.2. Command Line Interface Commands

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 next chapter, with a subsection for each command, in alphabetical order of the command names.

Command	Parameters	Description	Useable on Backup Shelf Manager
activate	IPMB address FRU device ID	Activates the specified FRU.	No
alarm	alarm type	Activates or clears TELCO alarms.	No
board	slot number (optional)	Shows information about boards.	No
boardreset	slot number	Resets the specified CompactPCI/ATCA 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 level for the Shelf Manager or sets a new debug level.	Yes
exit/quit		Exits from the interpreter in interactive mode.	Yes

fans	IPMB address (optional) FRU device ID (optional)	Shows information about fans.	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[*]
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[*]
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[*]
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[*]
fruinfo	IPMB address FRU device ID	Provides user friendly FRU Information output.	Yes[*]
getfanlevel	IPMB address (optional) FRU device ID (optional)	Shows the current level of the fan controlled by the specified FRU.	No
getfruledstate	IPMB address (optional) FRU device ID (optional) LED ID or "ALL"(optional)	Shows the FRU LED state.	Yes[*]

[*] When executed on the backup Shelf Manager, this command reports information only about entities that are local to the backup Shelf Manager.

gethysteresis	IPMB address (optional) sensor name (optional) sensor number (optional)	Shows both the positive and negative hystereses of the specified sensor.	Yes*
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 controllers is an IPMB hub, information about a specific link is shown.	Yes*
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*
getthreshold, threshold	IPMB address (optional) sensor name (optional) sensor number (optional)	Shows threshold information about a specific sensor.	Yes*
help		Shows the list of supported commands.	Yes
ipmc	IPMB address (optional)	Shows information about one or all IPM controllers in the shelf.	Yes*
localaddress		Retrieves the IPMB address of the current Shelf Manager.	Yes
minfanlevel	fan level (optional)	Shows or sets the minimum fan level.	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

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[*]
sensordata	IPMB address (optional) sensor name (optional) sensor number (optional)	Shows value information for a specific sensor.	Yes[*]
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[*]
session		Shows information about active RMCP sessions.	No
setextracted	IPMB address FRU device ID	Notifies the Shelf Manager that the specified FRU has been physically extracted from the shelf.	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[*]
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[*]
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[*]

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*
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*
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*
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 commands are 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”.

In the command syntax used in the document, optional elements are enclosed in square brackets (“[”, “]”), variable elements in the command line (e.g. IPMB address, FRU device ID) are enclosed in angle brackets “<”, “>”. A vertical bar “|” separates parameter alternatives.

Chapter

3

3. Command Reference

This chapter summarizes the details of the individual commands of the CLI and provides 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 on the IPMB address and numerical FRU identifier::

- board <N>
- power_supply <N>[1]
- fan_tray <N>

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>"

Also, the special abbreviation "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. Here "shm 1" relates to the Shelf Manager with the numerically smaller hardware address, and "shm 2" relates to the Shelf Manager with the numerically greater hardware address.

In redundant configurations, not all commands listed below are supported by the backup Shelf Manager. See section 2.2 for a list of all the CLI commands, including identification of which ones are supported by the backup Shelf Manager

[1] Note: the reference notation power_supply <N>, plus its abbreviation, is supported only in CompactPCI shelves.

3.1. activate

Syntax:

activate <IPMB-address> <FRU-id>

activate board <N>

activate shm <N>

Purpose:

This command sends the IPMI command “Set FRU Activation (Activate FRU)” to the specified FRU. The FRU is specified using the IPMB address of the owning IPM controller and the FRU device ID. FRU device ID 0 designates the IPM controller proper in PICMG 3.0 contexts. In 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 were not listed in the power management table in the Shelf FRU Information, or for which the “Shelf Manager Controlled Activation” attribute is set to 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.

Example:

Activate the IPM controller proper at address 9C.

```
# clia activate 9c 0

Pigeon Point Shelf Manager Command Line Interpreter

    Command executed successfully

#
```

3.2. alarm

Syntax:

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

Purpose:

This command provides access to the TELCO alarm outputs. Parameters "minor", "major" and "critical" allow the user to set the corresponding alarm output. These actions are cumulative; that is, after the commands "clia alarm minor" and "clia alarm major", both minor and major alarms will be set. The action "clear" clears the minor and major alarm outputs; critical alarm output cannot be cleared. The action "info" displays information about the last alarm that occurred in the system.

Command invocation without parameters will return the status of TELCO alarm outputs.

Example:

```
# 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
```

3.3. board

Syntax:

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

Purpose:

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

- makes them easier for the end user;
- allows their use in PICMG 2.x contexts, where boards may not carry an intelligent IPM controller on them and therefore, are not easily addressable using the IPMB address – FRU device ID pair.

The command "board" shows information about each IPM controller in the range of IPMB addresses allocated to CompactPCI/ATCA slots, and about each additional FRU controlled by these controllers. The list of items to be shown is given in sections 0 and 3.10. 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, the following mapping table (mapping of logical slot numbers) is used.

Slot number	IPMB address
1	82
2	84
3	86
4	88
5	8A
6	8C
7	8E
8	90
9	92
10	94
11	96

12	98
13	9A
14	9C
15	9E
16	A0

For PICMG 2.9 based systems, the following CompactPCI Peripheral address mapping table is used, where “Slot Number” refers to the PICMG 2.x concept of “Physical Slot Number”:

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

Example:

Get standard information about all boards in the system (where only the boards in physical slots 1 and 14 are present).

```
# clia board

Pigeon Point Shelf Manager Command Line Interpreter

Physical Slot # 1

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

82: 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"

Physical Slot # 14

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

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 a board in physical slot 14.

```
# clia board -v 14

Pigeon Point Shelf Manager Command Line Interpreter

Physical Slot # 14

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

#
```

3.4. boardreset

Syntax:

boardreset <physical-slot-address>

Purpose:

This command resets the board in the specified physical slot, sending it the IPMI command FRU Control (Cold Reset).

The physical address should be specified as a decimal number. For PICMG 3.0 systems, correspondence between physical addresses and IPMB addresses is specified in the Shelf FRU Information. If the Shelf FRU information does not contain an address table, the following mapping table (mapping of logical slot numbers) is used. FRU device ID is 0.

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

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

Example:

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

```
# clia boardreset 14

Pigeon Point Shelf Manager Command Line Interpreter

Board 14 reset, status returned 0

#
```

3.5. busres

Syntax:

```
busres <subcommand>
```

The following subcommands are supported:

```
info [<resource>]
release <resource>
force <resource>
lock <resource>
unlock <resource>
query [-v] <resource> [<target> [noupdate]]
setowner <resource> <target>
sendbusfree <resource> <target>
```

Purpose:

This command shows information about the current state of the Bused E-Keying-managed resources and allows changing that state.

All subcommands accept a resource ID as one of the parameters. The resource ID is either a 0-based resource number or a short resource name. The following resource names and numbers are defined:

Number	Short Name	Description
0	mtb1	Metalic Test Bus pair 1
1	mtb2	Metalic Test Bus pair 2
2	clk1	Synch Clock group 1
3	clk2	Synch Clock group 2
4	clk3	Synch Clock group 3

The following subsections describe the syntax of the "busres" command for several key uses.

3.5.1. Displaying the State of Bused E-Keying-Managed Resources

Syntax:

busres info [<resource>]

Purpose:

This command displays information about the current state of the specified resource or all resources, if the resource ID is not specified.

The parameter <resource> is the resource ID. The list of supported resource IDs is provided at the beginning of section 3.5.

Example:

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

#
```

3.5.2. Releasing a Specified Resource

Syntax:

busres release | force <resource>

Purpose:

This command sends the Bused Resource Control request to the current owner of the resource, instructing it to release the resource. If the command syntax is `busres release <resource>`, the Bused Resource Control (Release) command is sent. If the command syntax is `busres force <resource>`, the Bused Resource Control (Force) command is sent. See section 3.7.3.4 of the PICMG 3.0 R1.0 specification for a detailed description of these ATCA commands.

The parameter <resource> is the resource ID. The list of supported resource IDs is provided at the beginning of section 3.5.

Example:

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

#
```

3.5.3. Locking/unlocking a Specified Resource

Syntax:

busres lock | unlock <resource>

Purpose:

This command locks (`busres lock <resource>`) or unlocks (`busres unlock <resource>`) the specified resource. If the resource is locked, when another IPM Controller sends the Bused Resource Control (Request) command to the Shelf Manager, the Shelf Manager responds with the Deny status. If the resource is unlocked, when another IPM Controller sends the Bused Resource Control (Request) command to the Shelf Manager, the Shelf Manager responds with Busy status and sends the Bused Resource Control (Release) to the current owner. If the current owner releases the resource, on the next request, this resource will be granted to the requestor.

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 list of supported resource IDs is provided at the beginning of section 3.5.

Example:

Lock Synch Clock group 3.

```
# clia busres lock clk3

Pigeon Point Shelf Manager Command Line Interpreter

Lock operation succeeded

#
```

3.5.4. Send Bused Resource Control (Query) Command

Syntax:

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

Purpose:

This command sends the Bused Resource Control (Query) request to the specified IPM Controller. If the IPM Controller is not specified in the command line, the request is sent to the current owner of the resource. Upon receiving the response, appropriate changes are made in the resource table (for example, if the IPM Controller that is believed to be the current owner responds with the No Control status, the table is modified to reflect that fact), unless the noupdate flag is provided. If this flag is passed in the command line, no changes to the resource table are made based on the received information.

The parameter <resource> is the resource ID. The list of supported resource IDs is provided at the beginning of section 3.5.

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 (see section 3.20)[1].

The flag noupdate, if present, indicates that the information received in response to the Query request should not be used to update the resource table.

In the current revision of the Shelf Manager, no additional information is provided if –v flag is specified.

Example:

Send query for Metalic Test Bus pair 1 to the IPM Controller with address 0x82. 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

#
```

3.5.5. Set Owner for the Resource

Syntax:

busres setowner <resource> <target>

[1] Note: the reference notation power_supply <N>, plus its abbreviation, is supported only in CompactPCI shelves.

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 any Bused Resource Control commands, even if the resource had a different owner before executing the command. This is a low-level command that should be used for testing and recovery purposes only.

The parameter <resource> is the resource ID. The list of supported resource IDs is provided at the beginning of section 3.5.

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 `clia ipmc` command (see section 3.20)[1]. Use 0 as the IPMB address to specify that the resource is not owned by any IPM Controller.

Example:

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

```
# clia busres setowner mtb1 board 1

Pigeon Point Shelf Manager Command Line Interpreter

New owner is set successfully

#
```

3.5.6. Send Bused Resource Control (Bus Free) Command

Syntax:

busres sendbusfree <resource> <target>

Purpose:

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

[1] Note: the reference, power_supply <N>, plus its abbreviation, is supported only in CompactPCI shelves.

This command sends the Bused Resource Control (Bus Free) request to the specified IPM Controller. No operation is performed on the resource before sending the request even if a different IPM Controller owns it. However, the resource table is updated based on the response to this request. That is, if the IPM Controller accepts ownership of the resource, it is set as the new owner in that table. This is a low-level command that should be used for testing and recovery purposes only.

The parameter <resource> is the resource ID. The list of supported resource IDs is provided at the beginning of section 3.5.

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 (see section 3.20)[1]. Use 0 as the IPMB address to specify that the resource is not owned by any IPM Controller.

Example:

Send Bus Free request for Metalic Test Bus pair 1 to the IPM Controller with address 0x82.

```
# clia busres sendbusfree mtb1 0x82

Pigeon Point Shelf Manager Command Line Interpreter

IPMC rejected ownership of the resource

#
```

[1] Note: the reference notation power_supply <N>, plus its abbreviation, is supported only in CompactPCI shelves.

3.6. deactivate

Syntax:

deactivate <IPMB-address> <FRU-id>

deactivate board <N>

deactivate shm <N>

Purpose:

This command sends the IPMI command “Set FRU Activation (Deactivate FRU)” to the specified FRU. The FRU is specified using the IPMB address of the owning IPM controller and the FRU device ID. FRU device ID 0 designates the IPM controller proper in PICMG 3.0 contexts. 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 (either the BMC or the physical Shelf Manager IPMC) does not affect the Shelf Manager functionality and does not cause a switchover to the other Shelf Manager. Please see section 6.3 for more information on this topic.

Example:

Deactivate the IPM controller proper at address 9C.

```
# clia deactivate 9c 0

Pigeon Point Shelf Manager Command Line Interpreter

    Command executed successfully

#
```

3.7. debuglevel

Syntax:

debuglevel [<new-value>]

Purpose:

This command shows the current debug level for the Pigeon Point Shelf Manager, or sets it to a new value if a new value is specified.

The debug level is a hexadecimal number in the range 0x0000 to 0x00FF that is treated as a bit mask. Each bit in the mask, when set, enables debug output of a specific type:

0x0001	Error messages
0x0002	Warning messages
0x0004	Informational messages
0x0008	Verbose informational messages
0x0010	Trace messages
0x0020	Verbose trace messages
0x0040	Messages displayed for important commands sent to the IPM controllers during their initialization
0x0080	Verbose messages about acquiring and releasing internal locks

The default debug level for the Shelf Manager is 0x0007, but this value can be overridden during Shelf Manager startup, using the "-v" option in the command line. CLI provides an additional capability to change the debug level during runtime.

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

Example:

Get current debug level, and then set it to 0x001F.

```
# clia debuglevel

Pigeon Point Shelf Manager Command Line Interpreter

Debug Mask is 0x0007

#
# clia debuglevel 1f

Pigeon Point Shelf Manager Command Line Interpreter

Debug Mask is set to 0x001f
```

```
#
# clia debuglevel

Pigeon Point Shelf Manager Command Line Interpreter

Debug Mask is 0x001f

#
```

3.8. exit/quit

Syntax:

exit | quit

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.

Example:

```
CLI> exit

#
```

3.9. fans

Syntax:

fans [-v] [<IPMB-address> [<FRU-device-ID>]]

fans fan_tray <N>

Purpose:

This command shows information about the specified fan FRUs. If FRU device ID is omitted, the command shows information about all fan FRUs controlled by the IPM controller at the specified address. If the IPMB address is also omitted, the command shows information about all fan FRUs known to the Shelf Manager. The following information is shown:

- IPMB address and FRU device ID
- Minimum Speed Level
- Maximum Speed Level
- Maximum Sustained Speed Level
- Current Level (the pair of Override and Local Control levels if both are available)

Example:

Get fan information about all fan FRUs at IPMB address 9C.

```
# clia fans 9c

Pigeon Point Shelf Manager Command Line Interpreter

    No known fans at controller 0x9c

#
```

3.10. fru

Syntax[1]:

```
fru [-v] [<addr> [id=<fru_id> | type=<site_type>]] | [type=<site_type> [/<site_number>]]

fru board <N>

fru shm  <N>

fru power_supply <N>

fru fan_tray <N>
```

Purpose:

This command shows information about a specific FRU. If the FRU device ID is omitted, the command shows information about all FRUs controlled by the IPM controller at the specified address. If the IPMB address is also omitted, the command shows information about all FRUs known to the Shelf Manager.

Additionally, the site type can select FRUs. Site type should be specified in command parameters in hexadecimal. Associations between FRUs and their site types are stored in the Shelf FRU information. Site types are defined in the PICMG 3.0 specification as follows:

- 00h = AdvancedTCA Board
- 01h = Power Entry Module
- 02h = Shelf FRU Information
- 03h = Dedicated 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 reserved.

In the CompactPCI systems, the following OEM-defined site types are used to describe CompactPCI sites:

- C4h = CompactPCI Board

[1] Note: the reference notation power_supply <N>, plus its abbreviation, is supported only in CompactPCI shelves.

- 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.

Examples:

Get standard information about all FRUs at address 9C.

```
# clia fru 9c 0

Pigeon Point Shelf Manager Command Line Interpreter

9c: FRU # 0
    Entity: (0xd0, 0x0)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In Process), Last State
Change Cause: Normal State Change (0x0)
    Device ID String: "Pigeon Point 6"
```

```
#
```

Get verbose information about all FRUs at address 9C.

```
# clia fru -v 9c 0

Pigeon Point Shelf Manager Command Line Interpreter

9c: FRU # 0
    Entity: (0xd0, 0x0)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In Process), Last State
Change Cause: Normal State Change (0x0)
    Device ID String: "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 20.

```
# 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

#
```

3.11. frudata

Syntax[1]:

frudata [<addr> [<fru_id> [<block_offset>]]]

frudata <addr> <fru_id> <byte_offset> <byte 1> [byte2 … [byte 16] …]

<addr> <fru id> can be replaced with the following:

board <N>

shm <N>

power_supply <N>

fan_tray <N>

Purpose:

This command provides access to the FRU Information in raw form. Depending on the command format, it is used to read or write the FRU Information.

In the read format, the command takes an optional 32-byte block number.

In the write format it requires a byte offset parameter. The user can modify up to 65535 bytes of FRU Information.

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.

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)
```

[1] Note: the reference notation power_supply <N>, plus its abbreviation, is supported only in CompactPCI shelves.

```
    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
#
```

3.12. frudatar

Syntax[1]:

frudatar <addr> <fru_id> <file name>

<addr> <fru id> can be replaced with the following:

board <N>

shm <N>

power_supply <N>

fan_tray <N>

Purpose:

This command reads FRU Information from the specified FRU and stores it in a file on the ShMM flash file system in a raw format (in other words, uploads FRU Information from the specified FRU to a flash file). The parameter <file name> specifies the path to the destination file. The number of bytes read from the FRU and written to the destination file is equal to the number of bytes returned in the response to the IPMI command Get FRU Inventory Area Info for the specified FRU.

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.

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
```

[1] Note: the reference notation power_supply <N>, plus its abbreviation, is supported only in CompactPCI shelves.

```
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
```

3.13. frudataw

Syntax[1]:

frudataw <addr> <fru_id> <file name>

<addr> <fru id> can be replaced with the following:

board <N>

shm <N>

power_supply <N>

fan_tray <N>

Purpose:

This command downloads FRU Information to the specified FRU from a file on the ShMM flash file system. The file contains the raw binary image of the FRU Information. The parameter <file name> specifies the path to the source file.

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.

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
#
```

[1] Note: the reference notation power_supply <N>, plus its abbreviation, is supported only in CompactPCI shelves.

3.14. fruinfo

Syntax[1]:

fruinfo [-v] [-x]<addr> <fru_id>

<addr> <fru id> can be replaced by the following:

board <N>

shm <N>

power_supply <N>

fan_tray <N>

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

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
```

[1] Note: the reference notation power_supply <N>, plus its abbreviation, is supported only in CompactPCI shelves.

```
    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

#
```

3.15. frucontrol

Syntax[1]:

frucontrol <IPMB-address> <FRU-id> <option>

frucontrol board <N> <option>

frucontrol shm <N> <option>

frucontrol power_supply <N> <option>

frucontrol fan_tray <N> <option>

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.

For the option "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.

Example:

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

[1] Note: the reference notation power_supply <N>, plus its abbreviation, is supported only in CompactPCI shelves.

```
# 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 0E.

```
# 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
#
#
```

3.16. getfanlevel

Syntax:

getfanlevel <IPMB-address> <FRU-device-ID>

getfanlevel fan_tray <N>

Purpose:

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

Example:

Get fan level for the fan residing at FRU #2 at IPMB address 0x20.

```
# clia getfanlevel 20 2

Pigeon Point Shelf Manager Command Line Interpreter

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

#
```

3.17. getfruledstate

Syntax:

getfruledstate [-v] [<IPMB-addr> [<fru_id> [<LedId>|ALL]]]

Purpose:

This command shows 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. 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.

Example:

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 LED state for the IPM controller at IPMB address FCh:

```
# clia getfruledstate -v  FC

Pigeon Point Shelf Manager Command Line Interpreter
```

```
fc: FRU # 0, Led # 0 ("BLUE LED"):
    Local Control LED State: LED OFF
    LED's color capabilities:
        Colors supported(0x02): BLUE
        Default LED Color in Local Control State(0x01): BLUE
        Default LED Color in Override State(0x01): BLUE

fc: FRU # 0, Led # 1 ("LED 1"):
    Local Control LED State: LED OFF
    LED's color capabilities:
        Colors supported(0x0C): RED GREEN
        Default LED Color in Local Control State(0x03): GREEN
        Default LED Color in Override State(0x03): GREEN

fc: FRU # 0, Led # 2 ("LED 2"):
    Local Control LED State: LED OFF
    LED's color capabilities:
        Colors supported(0x0C): RED GREEN
        Default LED Color in Local Control State(0x03): GREEN
        Default LED Color in Override State(0x03): GREEN

fc: FRU # 0, Led # 3 ("LED 3"):
    Local Control LED State: LED OFF
    LED's color capabilities:
        Colors supported(0x0C): RED GREEN
        Default LED Color in Local Control State(0x02): RED
        Default LED Color in Override State(0x02): RED

fc: FRU # 0, Led # 4 ("Application Specific LED# 1"):
    Local Control LED State: LED ON, color: GREEN
    LED's color capabilities:
        Colors supported(0x0C): RED GREEN
        Default LED Color in Local Control State(0x02): RED
        Default LED Color in Override State(0x02): RED
```

Show LED state for FRU #0 of the IPM controller at IPMB address 20h:

```
# clia getfruledstate   20 0

Pigeon Point Shelf Manager Command Line Interpreter

20: FRU # 0, Led # 0 ("BLUE LED"):
    Local Control LED State: LED ON, color: BLUE

20: FRU # 0, Led # 1 ("LED 1"):
    Local Control LED State: LED OFF
```

Show LED state for LED #1 from FRU #0 of the IPM controller at IPMB address 20h

```
#clia getfruledstate  -v  20 0 1

Pigeon Point Shelf Manager Command Line Interpreter

20: FRU # 0, Led # 1 ("LED 1"):
    Local Control LED State: LED OFF
    LED's color capabilities:
        Colors supported(0x04): RED
        Default LED Color in Local Control State(0x02): RED
Default LED Color in Override State(0x02): RED
```

3.18. gethysteresis

Syntax:

gethysteresis [<IPMB-address> [[<lun>:] <sensor id> | <sensor name>]]

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 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.

Example:

Show the hysteresis values for sensor # 2 on the IPM controller at IPMB address FCh.

```
# clia gethysteresis FC 2

Pigeon Point Shelf Manager Command Line Interpreter

fc: LUN: 0, Sensor # 2 ("lm75 temp")
    Type: Threshold (0x01), "Temperature"(0x01)
        Positive hysteresis, Raw data: 0x00   Processed data: 0.00000 degrees C
        Negative hysteresis, Raw data: 0x00   Processed data: 0.00000 degrees C
```

3.19. getipmbstate

Syntax:

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

Purpose:

This command shows the current state of IPMB-0 on the target IPM Controller. The state is taken from the sensor data provided by the IPMB Link sensor on the target IPM controller (sensor type F1). Information about both buses A and B is printed.

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

- If <link> is omitted, the command prints information about the state of all radial IPMB links. The state is taken from the sensor data of the multiple IPMB link sensors on the IPM controller.
- If the <link> is present, the command prints information about the specific radial IPMB link (1 to 95). The state of the link is taken from the state of the corresponding IPMB link sensor on the IPM controller.

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

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.

Example:

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 link 8 for the Shelf Manager in the radial environment.

```
# clia getipmbstate 20 8

Pigeon Point Shelf Manager Command Line Interpreter

20: Link: 8, LUN: 0, Sensor # 12 ("IPMB LINK 8")
 Bus Status: 0x8  (IPMB-A Enabled, IPMB-B Enabled)
 IPMB A State: 0x8  (LocalControl, No failure)
 IPMB B State: 0x8  (LocalControl, No failure)
```

3.20. getlanconfig

Syntax:

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

Purpose:

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

The following table lists 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 types support flags for the LAN channel.
auth_enables	2	Five 8-bit values that contain authentication types enable flags for Callback, User, Operator, Administrator, and OEM privilege levels for the LAN channel.
ip	3	A string value that contains the IP address assigned to the LAN channel in dotted decimal notation (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])

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)

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

Example:

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

#
```

3.20.1. *auth_support*

Syntax:

getlanconfig <channel> auth_support | getlanconfig <channel> 1

Purpose:

This command shows the current value of the LAN parameter "auth_support". This parameter specifies which authentication types are supported by the Shelf Manager, represented by a single byte, treated as a bit mask with the 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.

Example:

```
# clia getlanconfig 1 auth_support

Pigeon Point Shelf Manager Command Line Interpreter

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

#
```

3.20.2. auth_enables

Syntax:

getlanconfig <channel> auth_enables | getlanconfig <channel> 2

Purpose:

This command shows the current value of the LAN parameter "auth_enables". This parameter specifies which authentication types are currently enabled by the Shelf Manager for each of five supported privilege levels (Callback, User, Administrator, Operator and OEM), represented by a sequence of five bytes, each corresponding to the respective privilege level, treated as a bit mask with the 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 values, symbolic values for the bits that are set are also shown.

Example:

```
# clia getlanconfig 1 auth_enables

Pigeon Point Shelf Manager Command Line Interpreter

Authentication Type Enables:
    Callback level: 0x00
    User level: 0x15 ( None MD5 Straight Password/Key )
    Operator level: 0x15 ( None MD5 Straight Password/Key )
```

```
        Administrator level: 0x15 ( None MD5 Straight Password/Key )
        OEM level: 0x00
    #
```

3.20.3. ip

Syntax:

getlanconfig <channel> ip | getlanconfig <channel> 3

Purpose:

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

Example:

```
# clia getlanconfig 1 ip

Pigeon Point Shelf Manager Command Line Interpreter

IP Address: 172.16.2.203

#
```

3.20.4. ip_source

Syntax:

getlanconfig <channel> ip_source | getlanconfig <channel> 4

Purpose:

This command shows the current value of the LAN parameter "ip_source". This parameter specifies the source of the IP Address used by the Shelf Manager, represented by a single byte, which can have one of the following values:

0	Unspecified
1	Static address (manually configured)
2	address obtained by Shelf Manager running DHCP
3	address loaded by BIOS or system software
4	address obtained by Shelf Manager running other address assignment protocol

Other values are reserved.

Besides the raw hexadecimal value, the symbolic value is also shown.

Example:

```
# clia getlanconfig 1 ip_source

Pigeon Point Shelf Manager Command Line Interpreter

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

#
```

3.20.5. *mac*

Syntax:

getlanconfig <channel> mac | getlanconfig <channel> 5

Purpose:

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

Example:

```
# clia getlanconfig 1 mac

Pigeon Point Shelf Manager Command Line Interpreter

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

#
```

3.20.6. *subnet_mask*

Syntax:

getlanconfig <channel> subnet_mask | getlanconfig <channel> 6

Purpose:

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

Example:

```
# clia getlanconfig 1 subnet_mask

Pigeon Point Shelf Manager Command Line Interpreter

Subnet Mask: 255.255.255.0

#
```

3.20.7. ipv4_hdr_param

Syntax:

getlanconfig <channel> ipv4_hdr_param | getlanconfig <channel> 7

Purpose:

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

Example:

```
# clia getlanconfig 1 ipv4_hdr_param

Pigeon Point Shelf Manager Command Line Interpreter

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

#
```

3.20.8. pri_rmcp_port

Syntax:

getlanconfig <channel> pri_rmcp_port | getlanconfig <channel> 8

Purpose:

This command shows the current RMCP primary port used by the channel, in hexadecimal. This is the port used for regular interactions via RMCP.

Example:

```
# clia getlanconfig 1 pri_rmcp_port

Pigeon Point Shelf Manager Command Line Interpreter

Primary RMCP Port Number: 0x026f

#
```

3.20.9. sec_rmcp_port

Syntax:

getlanconfig <channel> sec_rmcp_port |getlanconfig <channel> 9

Purpose:

This command shows the current RMCP secondary port used by the channel, in hexadecimal. This is the port used for secure interactions via RMCP.

Example:

```
# clia getlanconfig 1 sec_rmcp_port

Pigeon Point Shelf Manager Command Line Interpreter

Primary RMCP Port Number: 0x0298

#
```

3.20.10. arp_control

Syntax:

getlanconfig <channel> arp_control | getlanconfig <channel> 10

Purpose:

This command shows the current value of the LAN parameter "arp_control". This parameter specifies additional ARP support provided by the Shelf Manager, represented by a single byte, treated as a bit mask with the following meaning of the bits:

1	Enable Shelf Manager-generated Gratuitous ARPs
2	Enable Shelf Manager-generated ARP responses

Other bits are reserved and should be set to 0.

Besides the raw hexadecimal value, symbolic values for the bits that are set are also shown.

Example:

```
# clia getlanconfig 1 arp_control

Pigeon Point Shelf Manager Command Line Interpreter

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

#
```

3.20.11. arp_interval

Syntax:

getlanconfig <channel> arp_interval | getlanconfig <channel> 11

Purpose:

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

Example:

```
# clia getlanconfig 1 arp_interval

Pigeon Point Shelf Manager Command Line Interpreter

Gratuitous ARP Interval: 2.0 seconds

#
```

3.20.12. dft_gw_ip

Syntax:

getlanconfig <channel> dft_gw_ip | getlanconfig <channel> 12

Purpose:

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

Example:

```
# clia getlanconfig 1 dft_gw_ip

Pigeon Point Shelf Manager Command Line Interpreter

Default Gateway Address: 0.0.0.0

#
```

3.20.13. dft_gw_mac

Syntax:

getlanconfig <channel> dft_gw_mac | getlanconfig <channel> 13

Purpose:

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

Example:

```
# clia getlanconfig 1 dft_gw_mac

Pigeon Point Shelf Manager Command Line Interpreter

Default Gateway MAC Address: N/A

#
```

3.20.14. backup_gw_ip

Syntax:

getlanconfig <channel> backup_gw_ip | getlanconfig <channel> 14

Purpose:

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

Example:

```
# clia getlanconfig 1 backup_gw_ip

Pigeon Point Shelf Manager Command Line Interpreter

Backup Gateway Address: 0.0.0.0

#
```

3.20.15. backup_gw_mac

Syntax:

getlanconfig <channel> backup_gw_mac | getlanconfig <channel> 15

Purpose:

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

Example:

```
# clia getlanconfig 1 backup_gw_mac

Pigeon Point Shelf Manager Command Line Interpreter

Backup Gateway MAC Address: N/A

#
```

3.20.16. community

Syntax:

getlanconfig <channel> community | getlanconfig <channel> 16

Purpose:

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

Example:

```
# clia getlanconfig 1 community

Pigeon Point Shelf Manager Command Line Interpreter

Community String: "public"

#
```

3.20.17. destination_count

Syntax:

getlanconfig <channel> destination_count | getlanconfig <channel> 17

Purpose:

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

Example:

```
# clia getlanconfig 1 destination_count

Pigeon Point Shelf Manager Command Line Interpreter

Number of Destinations: 16

#
```

3.20.18. destination_type

Syntax:

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

Purpose:

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

- 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.

Example:

```
# 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
```

3.20.19. destination_address

Syntax:

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

Purpose:

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

- 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.

Example:

```
# clia getlanconfig 1 destination_address 2

Pigeon Point Shelf Manager Command Line Interpreter


DST Addresses # 2, Address Format: IPv4 IP Address followed by DIX ethernet / 802.3
MAC Address (0x00)
    Gateway: Default (0x00), Alerting IP: 172.16.2.100, Alerting MAC:
90:93:93:93:93:93
```

3.21. getpefconfig

Syntax:

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

Purpose:

This command shows the value of the specified PEF configuration parameter. If neither the configuration parameter name nor the parameter-number is specified, all PEF configuration parameters are shown.

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

Parameter Name	Number	Description
control	1	An 8-bit value that represents control flags for PEF (enable PEF, enable PEF startup delay, etc.)
action_control	2	An 8-bit value that represents PEF action global control flags (enable reset, enable power down, etc.)
startup_delay	3	Time to delay PEF after system power-ups and resets, in seconds
alert_startup_delay	4	Time to delay alerts after system power-ups and resets, in seconds
event_filter_count	5	Maximum number of event filters
event_filter	6	An event filter table entry identified by the specified set selector. If no set selector is given, all active event filters are shown.
event_filter_data1	7	The first byte of the event filter table entry identified by the specified set selector. If no set selector is given, all active event filters are shown.
alert_policy_count	8	Maximum number of alert policies
alert_policy	9	An alert policy table entry identified by the specified set selector. If no set selector is given, all active alert policies are shown.
system_guid	10	A GUID used to fill in the GUID field in the PET trap
alert_string_count	11	Maximum number of alert strings
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.

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

Example:

Get and show the whole PEF parameter table.

```
# clia getpefconfig

Pigeon Point Shelf Manager Command Line Interpreter

PEF parameters:
    PEF control: 0x00
    PEF Action Global Control: 0x00
    PEF Startup Delay: 60 seconds
    PEF Alert Startup Delay: 60 seconds
    PEF Number of Event Filters: 64
    PEF Number of OEM Filters: 16
    Active Event Filters:
        None
    Active 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
#
```

3.21.1. *control*

Syntax:

getpefconfig control | getpefconfig 1

Purpose:

This command shows the current value of the PEF parameter “control”. This parameter is a single byte, treated as a bit mask with the 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.

Example:

```
# clia getpefconfig control

Pigeon Point Shelf Manager Command Line Interpreter

    PEF control: 0x07
        Enable PEF
        Enable Event Message for PEF Actions
        Enable PEF Startup Delay

#
```

3.21.2. action_control

Syntax:

getpefconfig action_control | getpefconfig 2

Purpose:

This command shows the current value of the PEF parameter "action_control". This parameter is a single byte, treated as a bit mask with the 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.

Example:

```
# clia getpefconfig action_control

Pigeon Point Shelf Manager Command Line Interpreter

    PEF Action Global Control: 0x3f
        Enable Alert Action
        Enable Power Down Action
        Enable Reset Action
        Enable Power Cycle Action
        Enable OEM Action
    Enable Diagnostic Interrupt
#
```

3.21.3. startup_delay

Syntax:

getpefconfig startup_delay | getpefconfig 3

Purpose:

This command shows the current value of the PEF parameter “startup_delay”. This parameter is a single byte, representing the number of seconds that the PEF facility delays at startup.

Example:

```
# clia getpefconfig startup_delay

Pigeon Point Shelf Manager Command Line Interpreter

  PEF Startup Delay: 60 seconds

#
```

3.21.4. alert_startup_delay

Syntax:

getpefconfig startup_delay | getpefconfig 4

Purpose:

This command shows the current value of the PEF parameter “alert_startup_delay”. This parameter is a single byte, representing the number of seconds that the alerting facility delays at startup.

Example:

```
# clia getpefconfig alert_startup_delay

Pigeon Point Shelf Manager Command Line Interpreter

  PEF Alert Startup Delay: 60 seconds

#
```

3.21.5. event_filter_count

Syntax:

getpefconfig event_filter_count | getpefconfig 5

Purpose:

This command shows the current value of the PEF parameter "event_filter_count". This read-only value is the size of the event filter table. This value is a configuration parameter for the Pigeon Point Shelf Manager and can be changed only through the shelfman configuration file.

Example:

```
# clia getpefconfig event_filter_count

Pigeon Point Shelf Manager Command Line Interpreter

    PEF Number of Event Filters: 64

#
```

3.21.6. *event_filter*

Syntax:

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

Purpose:

This command shows the element of the event filter table with index equal to <set-selector>. Indexes are 1-based. The following information is shown about each event filter:

- filter configuration: whether the filter is software configured or manufacturer pre-configured
- event filter action mask
- alert policy number
- event severity
- Event source address to match (255 = any address)
- Source Channel/LUN to match (255 = match any source channel/LUN)
- Sensor type to match
- Sensor number to match
- Event trigger (event/reading type) to match
- Event offset mask
- AND, Compare 1 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.

Example:

```
# clia getpefconfig event_filter 2

Pigeon Point Shelf Manager Command Line Interpreter
    Active Event Filters:
```

```
    0x02: Software Configurable Filter
        Action Mask: 0x01
        Policy Number: 1, Severity: Critical Condition
        Source Address: 0x20, LUN: 3, Channel: 15
        Sensor Type: Hot Swap (0xf0), Sensor # 255 (ANY)
        Event Trigger: 0xff (ANY), Event Offset Mask: 0xffff
        0: AND: 0x0f, CMP1: 0xff, CMP2: 0x00
        1: AND: 0x00, CMP1: 0x00, CMP2: 0x00
        2: AND: 0xff, CMP1: 0xff, CMP2: 0x00


#
```

3.21.7. event_filter_data1

Syntax:

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

Purpose:

This command shows the first byte of the element of the event filter table with the index equal to <set-selector>. Indexes are 1-based. This byte is shown in hexadecimal. Bits in this byte have the following meaning:

0x80	This filter is enabled
0x40	This filter is pre-configured by the manufacturer and should not be altered by software

Other bits are reserved and should be 0.

If the set selector is omitted, first byte for each of the active event filter table entries is shown, with the corresponding filter numbers.

Example:

```
# clia getpefconfig event_filter_data1 2

Pigeon Point Shelf Manager Command Line Interpreter

    Active event filter data:
        0x02: 0x80 Enabled 1, Configuration: 0 ("Software Configurable Filter")


#
```

3.21.8. alert_policy_count

Syntax:

getpefconfig alert_policy_count | getpefconfig 8

Purpose:

This command shows the current value of the PEF parameter “alert_policy_count”. This read-only value is the size of the alert policy table. This value is a configuration parameter for the Pigeon Point Shelf Manager and can be changed only through the shelfman configuration file.

Example:

```
# clia getpefconfig alert_policy_count

Pigeon Point Shelf Manager Command Line Interpreter

    Alert Policies Count: 64

#
```

3.21.9. alert_policy

Syntax:

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

Purpose:

This command shows the element of the alert policy table with index equal to <set-selector>. Indexes are 1-based. The following information is shown about each alert policy:

- the policy number
- the policy type (with respect to the alert sent to the previous destination)
- destination channel number
- destination selector
- alert string key.

If the set selector is omitted, all active alert policy table entries are shown, with their numbers.

Example:

```
# clia getpefconfig alert_policy 2

Pigeon Point Shelf Manager Command Line Interpreter

    Policy:
        0x02: Policy# 5, Policy Type: 0, Channel: 1, DST: 1, Alert String Sel: 1

#
```

3.21.10. system_guid

Syntax:

getpefconfig system_guid | getpefconfig 10

Purpose:

This command shows the current value of the PEF parameter “system_guid”. This parameter represents the GUID that is sent in a PET Trap PDU to an alert destination. This GUID may be defined as a separate GUID or as being equal to the System GUID (which can be obtained via the Get System GUID IPMI command). See section 3.42.8 for a description of the getpefconfig system_guid command.

Example:

```
# clia getpefconfig system_guid

Pigeon Point Shelf Manager Command Line Interpreter

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

#
```

3.21.11. alert_string_count

Syntax:

getpefconfig alert_string_count | getpefconfig 11

Purpose:

This command shows the current value of the PEF parameter “alert_string_count”. This read-only value is the size of the alert string table, which is the maximum number of alert strings in simultaneous use. This value is the configuration parameter for the Pigeon Point Shelf Manager and can be changed only through the shelfman configuration file.

Example:

```
# clia getpefconfig alert_string_count

Pigeon Point Shelf Manager Command Line Interpreter

    Alert Strings Count: 64

#
```

3.21.12. alert_string_key

Syntax:

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

Purpose:

This command shows the element of the alert string key table with index equal to <set-selector>. Indexes are 1-based. Index 0 can be used to designate the volatile alert string. Each key associates an event filter with an alert string for alert generation purposes. The following information is shown about each alert string key:

- the alert string key number
- the associated event filter number
- the associated alert string number

If the set selector is omitted, all active alert string key table entries are shown with their numbers.

Example:

```
# clia getpefconfig alert_string_key 2

Pigeon Point Shelf Manager Command Line Interpreter

    Alert string key: set selector 2, event_filter 0x10, string_set 0x11

#
```

3.21.13. alert_string

Syntax:

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

Purpose:

This command shows the element of the alert string table with index equal to <set-selector>. Indexes are 1-based. Index 0 can be used to designate the volatile alert string. This command shows the whole string at once.

If the set selector is omitted, all defined alert strings are shown with their numbers.

Example:

```
# clia getpefconfig alert_string 2

Pigeon Point Shelf Manager Command Line Interpreter

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

```
#
```

3.21.14. oem_filter_count

Syntax:

getpefconfig oem_filter_count | getpefconfig 96

Purpose:

This command shows the current value of the PEF parameter "oem_filter_count". This read-only value is the size of the OEM filter table. This value is a configuration parameter for the 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).

Example:

```
# clia getpefconfig oem_filter_count

Pigeon Point Shelf Manager Command Line Interpreter

    PEF Number of OEM Filters: 16

#
```

3.21.15. oem_filter

Syntax:

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

Purpose:

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

Example:

```
# clia getpefconfig oem_filter

Pigeon Point Shelf Manager Command Line Interpreter

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

#
```

3.22. getsensoreventenable

Syntax:

getsensoreventenable [<IPMB-address> [<sensor-name> | [<lun>:]<sensor-number>]]

getsensoreventenable board <N> [<sensor-name> | [<lun>:]<sensor-number>]]

getsensoreventenable shm <N> [<sensor-name> | [<lun>:]<sensor-number>]]

Purpose:

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

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.

Examples:

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

```
# clia getsensoreventenable -v fe "Local Temp"

Pigeon Point Shelf Manager Command Line Interpreter

fe: LUN: 0, Sensor # 3 ("Local Temp")
```

```
    Type: Threshold (0x01), "Temperature" (0x01)
    Assertion event mask:   0x0a80
        Assertion event for "Upper Non-Recoverable Going High" enabled
        Assertion event for "Upper Critical Going High" enabled
        Assertion event for "Upper Non-Critical Going High" enabled
    Deassertion event mask: 0x0a80
        Deassertion event for "Upper Non-Recoverable Going High" enabled
        Deassertion event for "Upper Critical Going High" enabled
        Deassertion event for "Upper Non-Critical Going High" enabled

#
```

Get event enable information for the same sensor but specify sensor LUN and number.

```
# clia getsensoreventenable -v fe 0:3

Pigeon Point Shelf Manager Command Line Interpreter

fe: LUN: 0, Sensor # 3 ("Local Temp")
    Type: Threshold (0x01), "Temperature" (0x01)
    Assertion event mask:   0x0a80
        Assertion event for "Upper Non-Recoverable Going High" enabled
        Assertion event for "Upper Critical Going High" enabled
        Assertion event for "Upper Non-Critical Going High" enabled
    Deassertion event mask: 0x0a80
        Deassertion event for "Upper Non-Recoverable Going High" enabled
        Deassertion event for "Upper Critical Going High" enabled
        Deassertion event for "Upper Non-Critical Going High" enabled

#
```

3.23. getthreshold/threshold

Syntax:

getthreshold [<IPMB-address> [<sensor-name> | [<lun>:]<sensor-number>]]

getthreshold board <N> [<sensor-name> | [<lun>:]<sensor-number>]]

getthreshold shm <N> [<sensor-name> | [<lun>:]<sensor-number>]]

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

Purpose:

This command shows the current threshold values for the supported thresholds of the specified 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

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 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.

Examples:

Get threshold values for a temperature sensor "Local Temp" on IPM controller FE.

```
# clia getthreshold -v fe "Local Temp"

Pigeon Point Shelf Manager Command Line Interpreter

fe: LUN: 0, Sensor # 3 ("Local Temp")
    Type: Threshold (0x01), "Temperature" (0x01)
```

```
        Lower Critical Threshold, Raw Data: 0x80, Processed Data: -128.000000 degrees
C
        Upper Non-Critical Threshold, Raw Data: 0x50, Processed Data: 80.000000
degrees C
        Upper Critical Threshold, Raw Data: 0x50, Processed Data: 80.000000 degrees C
        Upper Non-Recoverable Threshold, Raw Data: 0x50, Processed Data: 80.000000
degrees C
```

Get threshold information for the same sensor but specify sensor LUN and number.

```
# clia getthreshold -v fe 0:3

Pigeon Point Shelf Manager Command Line Interpreter

fe: LUN: 0, Sensor # 3 ("Local Temp")
    Type: Threshold (0x01), "Temperature" (0x01)
        Lower Critical Threshold, Raw Data: 0x80, Processed Data: -128.000000 degrees
C
        Upper Non-Critical Threshold, Raw Data: 0x50, Processed Data: 80.000000
degrees C
        Upper Critical Threshold, Raw Data: 0x50, Processed Data: 80.000000 degrees C
        Upper Non-Recoverable Threshold, Raw Data: 0x50, Processed Data: 80.000000
degrees C
```

3.24. help

Syntax:

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

Purpose:

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

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

Example:

```
# 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>
  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>]
  exit
  fans <addr> <fru id>
  fru [<addr> [id=<fru_id> | type=<site_type>]] | [type=<site_type> [/<site_number>]]
  frucontrol <addr> <fru_id> <command>
  frudata [<addr>] [<fru id>] [<block number>]
  frudata shm <N> [<block number>]
  frudata <addr> <fru id> <byte offset> <byte_1> [byte2 .. [byte_16]]
  frudatar <addr> <fru id> <file name>
  frudataw <addr> <fru id> <file name>
  fruinfo <addr> <fru_id>
  getfanlevel <addr> <fru_id>
  getfruledstate [-v] [<addr> [<fru_id> [<LedId>|ALL]]]
  gethysteresis [ <addr> [ [ lun: ]<sensor id> | <sensor name> ] ]
  getipmbstate <addr> [<link>]
  getlanconfig <channel number> <parameter number> | <parameter name>
  getpefconfig <parameter name> | <parameter number> [<set selector>]
  getsensoreventenable [ <addr> [ [ lun: ]<sensor_id> | <sensor name>  ] ]
  getthreshold [ <addr> [ [ lun: ]<sensor id> | <sensor name> ] ]
  help [<command>]
  ipmc [<addr>]
  localaddress
  minfanlevel [<min fan level>]
  poll
  quit
  sel [clear] [ <addr> [ <number of items> [<number of first item>] ] ]
  sel info [<addr>]
  sensor [ <addr> [ [ lun: ]<sensor id> | <sensor name> ] ]
  sensordata [ <addr> [ [ lun: ]<sensor id> | <sensor name> ] ]
  sensorread <addr> [ lun: ]<sensor id>
  session
  setextracted <addr> <fru_id>
  setfanlevel <addr> <fru_id> <state>
  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
...>
  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
  sethysteresis <addr> [ lun: ]<sensor_id> | <sensor name> pos | neg <value>
  setfruledstate <addr> <fru_id> <LedId>|ALL <LedOp|tail> [LedColor]
  setpowerlevel <addr> <fru_id> [<pwr_lvl>|OFF] [Copy]
  shelf <parameters>
  shelfaddress ["<shelf address>"]
  shmstatus
  showunhealthy
  switchover
  terminate [-reboot]
  threshold [ <addr> [ [ lun: ]<sensor id> | <sensor name> ] ]
  user [<user id>]
  user add <user id> <user name> <flags> <privilege level> <password>
  user channel <user id> <channel number> <flags> <privilege level>
  user delete <user id>
  user 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 <addr> <fru_id> 1/0
        address_table
        Allowance <seconds>
        BDSelGrounded <slot number> 1/0
                1 means Enabled, 0 means Disabled
        cooling_state
        Deactivation <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 <addr> <fru_id> <Watts>
        PwrDelay <addr> <fru_id> <10ths_of_second>
        PwrReorder <addr1> <fru_id1> before/after <addr2> <fru_id2>

  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> (valid in 2.x systems only)
        fan_tray <N>

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

3.25. ipmc

Syntax[1]:

```
ipmc [-v] [<IPMB-address>]

ipmc board <N>

ipmc fan_tray <N>

ipmc power_supply <N>
```

Purpose:

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

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

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

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

- Information returned by the "Get Device ID" IPMI command, including manufacturer ID, product ID, device ID, device firmware revision and supported IPMI version

[1] Note: the reference notation power_supply <N>, plus its abbreviation, is supported only in CompactPCI shelves.

- Device ID string from the controller SDR
- Power state notification attribute from the controller SDR, as a hexadecimal number
- Global initialization attribute from the controller SDR, as a hexadecimal number
- Device capabilities attribute from the controller SDR, as a hexadecimal number
- Whether the controller provides Device SDRs
- Supported features mask, with a textual explanation of each bit
- The list of ports subject to E-Keying, with their states (Enabled/Disabled)

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.

Examples:

Get information about the IPM controller at address 9C.

```
# clia ipmc 9c

Pigeon Point Shelf Manager Command Line Interpreter

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

#
```

Get verbose information about the IPM controller at address 9C.

```
# clia ipmc -v 9c

Pigeon Point Shelf Manager Command Line Interpreter

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

#
```

Get verbose information about the IPM controller at address 10.

```
# clia ipmc -v 10

Pigeon Point Shelf Manager Command Line Interpreter

10: Entity: (0xf0, 0x60) Maximum FRU device ID: 0x08
    PICMG Version 2.1
    Hot Swap State: M4 (Active), Previous: M3 (Activation In Process), Last State
Change Cause: Normal State Change (0x0)
    Device ID: 0x00, Revision: 0, Firmware: 2.30, IPMI ver 1.5
    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"
    10: Base Interface (0x00), Channel: 1
        Link: Disabled Ports: 1
    10: Base Interface (0x00), Channel: 2
        Link: Disabled Ports: 1
```

3.26. localaddress

Syntax:

localaddress

Purpose:

This command shows the IPMB address of the current Shelf Manager, based on its hardware address (as opposed to its generic BMC address 0x20). These addresses will be different between redundant Shelf Managers (while the BMC address is shared between them).

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

Example:

```
# clia localaddress

Pigeon Point Shelf Manager Command Line Interpreter

Local IPMB Address = 0xFC

#
```

3.27. minfanlevel

Syntax:

minfanlevel [<level>]

Purpose:

This command shows or sets the minimum fan level. Under normal conditions, the cooling management algorithm gradually decreases the level for the fans in the system while thermal conditions stay normal. However the cooling management algorithm 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 SetFanLevel submitted over RMCP. The minimum fan level affects only the automatic management of the fan level by the cooling management facility.

This command without parameters shows the current minimum fan level.

This command with an integer parameter sets the minimum fan level to the value of the parameter.

Example:

```
# clia minfanlevel 3

Pigeon Point Shelf Manager Command Line Interpreter

Minimal Fan Level is set to 3

# clia minfanlevel

Pigeon Point Shelf Manager Command Line Interpreter

Minimal Fan Level is 3
```

3.28. poll

Syntax:

poll

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” makes the Shelf Manager to recognize new IPM controllers.

In AdvancedTCA shelves, this command is not necessary, because a new IPM controller is recognized by the Shelf Manager automatically when it sends its first Hot Swap event. Nevertheless, this command can be used in AdvancedTCA shelves if an IPMB-0 population rediscovery cycle is needed.

Example:

```
# clia poll

Pigeon Point Shelf Manager Command Line Interpreter

IPMB polling thread started
```

3.29. sel

Syntax:

sel [-v] [<IPMB-address> [<record-count> [starting-entry]]] |

sel clear [<IPMB-address>]

sel info [<IPMB-address>]

<IPMB addr> can be replaced by the "board <N>" or "shm <N>" abbreviations

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 from the record number <starting-entry> in the SEL are shown. The optional parameter <starting-entry> is the entry number of the first SEL record to print, relative to the beginning of the SEL. Both <record-count> and <starting-entry> must be within the range from 1 to the total number of records in the SEL. The default value of the optional parameter <starting-entry> is 1. The <starting-entry> is independent of the RecordID field of the SEL record.

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

- Record ID
- Record type (currently only events are supported, for which the word "Event" is shown
- Timestamp (for timestamped records)
- Source address parameters: IPMB address, LUN and channel number
- Type and number of the sensor that generated the event
- Event/reading type code
- 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.

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 from entry # 15 (0x0f).

```
# clia sel 20 5 15

Pigeon Point Shelf Manager Command Line Interpreter

0x000F: Event: at Nov 19 16:49:21 2003; from:(0x20,0,0); sensor:(0xf0,3);
event:0x6f(asserted): HotSwap: FRU 2 M2->M3, Cause=0x1
0x0010: Event: at Nov 19 16:49:22 2003; from:(0x20,0,0); sensor:(0xf0,2);
event:0x6f(asserted): HotSwap: FRU 1 M2->M3, Cause=0x1
0x0011: Event: at Nov 19 16:49:22 2003; from:(0x20,0,0); sensor:(0xf0,2);
event:0x6f(asserted): HotSwap: FRU 1 M3->M4, Cause=0x0
0x0012: Event: at Nov 19 16:49:22 2003; from:(0xfc,0,0); sensor:(0xf0,0);
event:0x6f(asserted): HotSwap: FRU 0 M3->M4, Cause=0x0
0x0013: Event: at Nov 19 16:49:22 2003; from:(0x20,0,0); sensor:(0xf0,3);
event:0x6f(asserted): HotSwap: FRU 2 M3->M4, Cause=0x0
#
```

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

#
```

3.30. sendcmd

Syntax:

sendcmd <IPMB-address> <net-fn> <command-code> [<parameter1> …<parameterN>]

Purpose:

This command allows the user to send an IPMI command to an IPM contoller 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 <net-fn> parameter. The code of the command is specified by the <command-code-parameter>. The request data bytes of the command are represented by <parameter1>, <parameter2>, etc.

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

Examples:

Send the GetDeviceId command to the Shelf Manager (IPMB address 0x20). The NetFn of the command is 0x6, the code of the command is 0x1. 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
#
```

3.31. sensor

Syntax:

```
sensor [ -v ] [ <IPMB-address> [<sensor-name> | [<lun>:]<sensor-number> ] ]

sensor [-v] board <N> [<sensor-name> | [<lun>:]<sensor-number> ] ]

sensor [-v] shm  <N> [<sensor-name> | [<lun>:]<sensor-number> ] ]
```

Purpose:

This command shows information about specific 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 allows the user to qualify the sensor number with the Logical Unit Number (LUN) if the target controller supports sensors on multiple LUNs. If the LUN is omitted, information about sensors with the specified sensor number on all LUNs is shown. <Lun> can take the value 0, 1 or 3 (LUN 2 is reserved.)

Sensor names are not qualified with LUN numbers, since it is assumed that sensor names will normally be unique within the controller. However, if there are several sensors with the same name within the controller, information is shown about all of them.

The following information is shown for each sensor in standard mode:

- IPMB address of the owning IPM controller
- Sensor number, sensor name (device ID string from the SDR) and the LUN by which the sensor can be accessed
- The sensor type and event/reading type code
- The Entity ID, Entity Instance of the related entity (the FRU device ID if the sensor is associated with a FRU)

The following information is shown for the sensor in verbose mode only (see the IPMI specification for information about these attributes):

- Assertion mask, deassertion mask and settable/readable mask for sensor states (in the case of a discrete sensor) or thresholds (in the case of a threshold-based sensor)

The following information is shown in verbose mode for threshold-based sensors only:

- Sensor units: base and modified
- Unit percentage, modifier and rate
- Analog format and flags
- Linearization parameters, M, B, K1, K2 coefficients
- Tolerance and accuracy coefficients
- Nominal, normal maximum, normal minimum, maximum and minimum values
- Upper thresholds: non-critical, critical and non-recoverable
- Lower thresholds: non-critical, critical and non-recoverable
- Hysteresis values: positive and negative.

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.

Examples:

Get standard information about sensor "FAN 4" on IPM controller FE.

```
# 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 9C.

```
# 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)

#
```

3.32. sensordata

Syntax:

```
sensordata [-v] [-t] [ <IPMB-address> [<sensor-name> | [<lun>:]<sensor-number> ] ]

sensordata [-v] [-t] board <N> [<sensor-name> | [<lun>:]<sensor-number> ] ]

sensordata [-v] [-t] shm  <N> [<sensor-name> | [<lun>:]<sensor-number> ] ]
```

Purpose:

This command shows the actual value of the specified sensor/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.

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.

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

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").

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.

Examples:

Get sensor data values for a temperature sensor "Local Temp" on IPM controller FE.

```
# clia sensordata FE "Local Temp"

Pigeon Point Shelf Manager Command Line Interpreter

fe: LUN: 0, Sensor # 3 ("Local Temp")
    Type: Threshold (0x01), "Temperature" (0x01)
    Status: 0xc0
        All event messages enabled from this sensor
        Sensor scanning enabled
        Initial update completed
    Raw data: 22 (0x16)
    Processed data: 22.000000 degrees C
    Status: 0x00

#
```

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

```
# clia sensordata 9c 0

Pigeon Point Shelf Manager Command Line Interpreter

9c: LUN: 0, Sensor # 0 ("FRU 0 HOT_SWAP")
    Type: Discrete (0x6f), "Hot Swap" (0xf0)
    Status: 0xc0
        All event messages enabled from this sensor
        Sensor scanning enabled
        Initial update completed
    Sensor reading: 0x00
    Current State Mask 0x0010

#
```

Get sensor data values for the same sensor, but qualifying it explicitly with the LUN.

```
# clia sensordata 9c 0:0

Pigeon Point Shelf Manager Command Line Interpreter

9c: LUN: 0, Sensor # 0 ("FRU 0 HOT_SWAP")
    Type: Discrete (0x6f), "Hot Swap" (0xf0)
    Status: 0xc0
```

```
        All event messages enabled from this sensor
        Sensor scanning enabled
        Initial update completed
    Sensor reading: 0x00
    Current State Mask 0x0010

#
```

3.33. sensorread

Syntax:

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

Purpose:

This command shows the raw value of the specified sensor. The only difference between the commands "sensorread" and "sensordata" is that the command "sensorread" does not check the presence of the target IPM controller or the validity of the sensor number, but just sends the "Get Sensor Reading" request directly via IPMB. This command does not retrieve the SDR of the sensor and thus it can not process the obtained data.

This command allows the user to qualify the sensor number with the Logical Unit Number (LUN) if the target controller supports sensors on multiple LUNs. If the LUN is omitted, LUN 0 is used. <Lun> can take values 0, 1 or 3. (LUN 2 is reserved.)

The following information is shown for each sensor:

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

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.

Examples:

Get sensor data values for sensor 4 on IPM controller FC. Notice that the "sensorread" command provides only unprocessed sensor values. Also notice the command example with an explicit LUN.

```
# clia sensordata fc 4

Pigeon Point Shelf Manager Command Line Interpreter

fc: LUN: 0, Sensor # 4 ("3.3STBY voltage")
    Type: Threshold (0x01), "Voltage" (0x02)
    Status: 0xc0
        All event messages enabled from this sensor
        Sensor scanning enabled
        Initial update completed
    Raw data: 193 (0xc1)
    Processed data: 3.396800 Volts
    Status: 0x00

# clia sensorread fc 4

Pigeon Point Shelf Manager Command Line Interpreter

fc: LUN: 0, Sensor # 4
    Raw data: 193 (0xc1)
    Status: 0xc0
        All event messages enabled from this sensor
        Sensor scanning enabled
        Initial update completed
    Threshold Sensor Status: 0x00
    Discrete Sensor Current State Mask 0x0000

# clia sensorread fc 0:4

Pigeon Point Shelf Manager Command Line Interpreter

fc: LUN: 0, Sensor # 4
    Raw data: 193 (0xc1)
    Status: 0xc0
        All event messages enabled from this sensor
        Sensor scanning enabled
        Initial update completed
    Threshold Sensor Status: 0x00
    Discrete Sensor Current State Mask 0x0000
```

3.34. session

Syntax:

session

Purpose:

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

- the maximum possible number of sessions and the number of currently active sessions;
- for each currently active session:
  - session handle
  - the user ID and name used during session activation
  - maximum session privilege level
  - the IPMI channel number and type
  - for LAN sessions, peer IP address and port number.

Example:

```
# clia session

Pigeon Point Shelf Manager Command Line Interpreter

32 sessions possible, 2 sessions currently active

Session: 1
   User: ID 1, Name: ""; Privilege Level: "Administrator"
   Channel: 1 ("LAN_802_3"); Peer IP address: 172.16.2.203, Port: 1764

Session: 2
   User: ID 1, Name: ""; Privilege Level: "Administrator"
   Channel: 1 ("LAN_802_3"); Peer IP address: 172.16.2.203, Port: 1765
#
```

3.35. setextracted

Syntax:

setextracted <IPMB-address> <FRU-id>

Purpose:

This command notifies the Shelf Manager that the specified FRU has been physically extracted from the shelf. If the specified FRU is in state M7, the Shelf Manager places it in state M0 (FRU physically absent).

Example:

```
# clia setextracted 9c 0

Pigeon Point Shelf Manager Command Line Interpreter

    Set FRU extracted state successfully

#
```

3.36. setfanlevel

Syntax:

setfanlevel <IPMB-address> <FRU-device-ID> <level>

setfanlevel fan_tray <N> <level>

setfanlevel all <level>

Purpose:

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

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

Example:

Set fan level for the fan residing at FRU #2 at IPMB address 0x20 to 5.

```
# clia setfanlevel 20 2 5

Pigeon Point Shelf Manager Command Line Interpreter

20: FRU # 2 Set Fan Level to: 5
#
```

Set fan level to 4 for all known fans in the shelf:

```
# clia setfanlevel all 4

Pigeon Point Shelf Manager Command Line Interpreter

72: FRU # 0 Set Fan Level to: 4
76: FRU # 0 Set Fan Level to: 4
#
```

3.37. setfruledstate

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>

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 the local control of the LED
- BLINK – cause the LED to blink, repeatedly turning it on for <onTime> milliseconds and then turning it off for <offTime> milliseconds
- TEST – run a lamp test for <onTime> milliseconds.

For the TEST operation <onTime> must be less then 12800 ms (12.8 sec); for the BLINK operation both <onTime> and <offTime> values must be within 10 – 2500 ms range.

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

- BLUE = 1
- RED = 2
- GREEN = 3
- AMBER = 4
- ORANGE = 6
- NONE = 14 (don't change color).

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.

Example:

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 colour. 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
```

3.38. sethysteresis

Syntax:

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

Purpose:

This command sets the value for the specified hysteresis for the specified sensor. The sensor must be a threshold-based sensor. It must support the designated threshold hysteresis and the hysteresis must be settable.

The command allows the user to qualify the sensor number with the Logical Unit Number (LUN) if the target controller supports sensors on multiple LUNs. The command sets the positive hysteresis if the "pos" argument is present and sets the negative hysteresis if the "neg"argument is present.

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.

Example:

Set positive hysteresis for sensor #2 of the IPM controller at IPMB address 0xFC

```
# clia sethysteresis FC 2 pos 10

Pigeon Point Shelf Manager Command Line Interpreter

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

3.39. setipmbstate

Syntax:

setipmbstate <IPMB-address> A|B [<link>] 1|0 (in radial IPMB-0 environment)
setipmbstate <IPMB-address> A|B 1|0 (in bused IPMB-0 environment)

Purpose:

This command enables/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 environments. In a bused environment, and in radial environment for target IPM controllers other than an IPMB Hub, argument <link> is not used. For an IPMB hub controller in a radial environment, the argument <link> is optional.

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

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.

Example:

Disable IPMB-A link.on the IPM controller at IPMB address 92h

```
# clia setipmbstate 92 A 0

Pigeon Point Shelf Manager Command Line Interpreter

     Command executed successfully
```

Enable radial IPMB link 3, bus B on the Shelf Manager (which is the IPMB hub):

```
# clia setipmbstate 20 B 3 1

Pigeon Point Shelf Manager Command Line Interpreter

     Command executed successfully
```

3.40. setlanconfig

Syntax:

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

Purpose:

This command sets the value of the specified LAN configuration parameter on the specified channel. The channel number, the configuration parameter name or number, and the parameter value should be explicitly specified.

The following table lists 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 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)

3.40.1. auth_enables

Syntax:

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

Purpose:

This command sets the current value of the LAN parameter "auth_enables". This parameter specifies which authentication types are currently enabled by the Shelf Manager for each of five supported privilege levels (Callback, User, Administrator, Operator and OEM) and is represented by a sequence of five bytes, each corresponding to the respective privilege level, treated as a bit mask with the 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.

Example:

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

Pigeon Point Shelf Manager Command Line Interpreter
```

```
Authentication Type Enables set successfully

#
```

3.40.2. ip

Syntax:

setlanconfig <channel> ip <value>| setlanconfig <channel> 3 <value>

Purpose:

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

Example:

```
# clia setlanconfig 1 ip 172.16.2.203

Pigeon Point Shelf Manager Command Line Interpreter

IP set successfully

#
```

3.40.3. subnet_mask

Syntax:

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

Purpose:

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

Example:

```
# clia setlanconfig 1 subnet_mask 255.255.255.0

Pigeon Point Shelf Manager Command Line Interpreter

Subnet Mask set successfully

#
```

3.40.4. ipv4_hdr_param

Syntax:

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

Purpose:

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

Example:

```
# clia setlanconfig 1 ipv4_hdr_param 37 E0 11

Pigeon Point Shelf Manager Command Line Interpreter

IPv4 Header Parameters set successfully

#
```

3.40.5. arp_control

Syntax:

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

Purpose:

This command sets the current value of the LAN parameter "arp_control". This parameter specifies additional ARP support provided by the Shelf Manager, and is represented by a single byte, treated as a bit mask with the 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.

Example:

```
# clia setlanconfig 1 arp_control 3

Pigeon Point Shelf Manager Command Line Interpreter

BMC-generated ARP control set successfully

#
```

3.40.6. arp_interval

Syntax:

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

Purpose:

This command sets the current ARP interval used by the channel. The value should represent the number of seconds/milliseconds in fixed-point numeric format (with a possible fractional part). Due to the definition of this parameter in IPMI, it is truncated to the largest time interval that is divisible by 500 milliseconds.

Example:

```
# clia setlanconfig 1 arp_interval 3.5

Pigeon Point Shelf Manager Command Line Interpreter

Gratuitous ARP interval set successfully

#
```

3.40.7. dft_gw_ip

Syntax:

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

Purpose:

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

Example:

```
# clia setlanconfig 1 dft_gw_ip 172.16.2.100

Pigeon Point Shelf Manager Command Line Interpreter

Default Gateway Address set successfully

#
```

3.40.8. backup_gw_ip

Syntax:

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

Purpose:

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

Example:

```
# clia setlanconfig 1 backup_gw_ip 172.16.2.100

Pigeon Point Shelf Manager Command Line Interpreter

Backup Gateway Address set successfully

#
```

3.40.9. community

Syntax:

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

Purpose:

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

Example:

```
# clia setlanconfig 1 community “Community”

Pigeon Point Shelf Manager Command Line Interpreter

Community string set successfully

#
```

3.40.10. destination_type

Syntax:

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

Purpose:

This command sets the element of the destination table with the index equal to <set-selector>. Indexes are 0-based. Selector 0 is used to address the volatile destination. Values <value1>, <value2> and

<value3> supply information about the new destination according to section 19.2 of the IPMI specification. The following information is supplied:

- the alert destination type (PET Trap or OEM destination; whether the alert should be acknowledged)
- alert acknowledge timeout
- retry count

Example:

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

Pigeon Point Shelf Manager Command Line Interpreter

Destination Type set successfully

#
```

3.40.11. destination_address

Syntax:

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

Purpose:

This command sets the element of the destination address table with the index equal to <set-selector>. Indexes are 0-based. Selector 0 is used to address the volatile destination. The command parameters supply the necessary information:

- <gateway_sel> - gateway to use: 0 for default gateway, 1 for backup gateway
- <IP-address> - the destination IP address in dotted-decimal notation
- <MAC-address> the destination MAC address, six hexadecimal bytes separated by colons

Example:

```
# clia setlanconfig 1 destination_address 2 0 172.16.2.100 90:93:93:93:93:93

Pigeon Point Shelf Manager Command Line Interpreter

Destination Addresses set successfully

#
```

3.41. setlocked

Syntax[1]:

setlocked <IPMB-address> <FRU-id> 0 |
setlocked <IPMB-address> <FRU-id> 1

setlocked board <N> 0 | 1

setlocked shm <N> 0 | 1

setlocked power_supply <N> 0 | 1

setlocked fan_tray <N> 0 | 1

Purpose:

This command sets the Locked bit for the specified FRU to the specified state (0 for unlock or 1 for lock). The FRU is specified using the IPMB address of the owning IPM controller and the FRU device ID. FRU device ID 0 designates the IPM controller proper in PICMG 3.0 contexts.

The Locked bit controls, according to the PICMG 3.0 specification, whether the FRU is allowed to autonomously progress from state M1 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 a specified state for FRUs that are local to the backup Shelf Manager.

Example:

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

```
# clia setlocked 9c 0 0

Pigeon Point Shelf Manager Command Line Interpreter

    Lock set successfully to 0x0
```

[1] Note: the reference notation power_supply <N>, plus its abbreviation, is supported only in CompactPCI shelves.

```
#
```

3.42. setpefconfig

Syntax:

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

Purpose:

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

event_filter	6	An event filter table entry identified by the specified set selector. Consists of the following 19 numeric values, in hexadecimal, encoded according to the definition in table 15-2 of the IPMI specification version 1.5: • filter configuration • event filter action • alert policy number • event severity • generator ID byte 1 • generator ID byte 2 • sensor type • sensor number • event trigger (event/reading type) • event data 1 event offset mask • event data 1 AND mask • event data 1 compare 1 • event data 1 compare 2 • event data 2 AND mask • event data 2 compare 1 • event data 2 compare 2 • event data 3 AND mask • event data 3 compare 1 • event data 3 compare 2
event_filter_data1	7	The first byte of the event filter table entry identified by the specified set selector.
alert_policy	9	An alert policy table entry identified by the specified set selector. Consists of the following 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.

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.

3.42.1. control

Syntax:

setpefconfig control <value> | setpefconfig 1 <value>

Purpose:

This command sets a new value of the PEF parameter "control". This parameter is a single byte, treated as a bit mask with the 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.

Example:

```
# clia setpefconfig control 7

Pigeon Point Shelf Manager Command Line Interpreter

PEF control set successfully

#
```

3.42.2. action_control

Syntax:

setpefconfig action_control <value> | setpefconfig 2 <value>

Purpose:

This command sets a new value of the PEF parameter "action_control". This parameter is a single byte, treated as a bit mask with the 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

Example:

```
# clia setpefconfig action_control 3f

Pigeon Point Shelf Manager Command Line Interpreter

PEF action control set successfully
#
```

3.42.3. *startup_delay*

Syntax:

setpefconfig startup_delay <value> | setpefconfig 3 <value>

Purpose:

This command sets the new value of the PEF parameter "startup_delay". This parameter is a single byte, representing the number of seconds that the PEF facility delays at startup. The value is specified as a decimal number of seconds.

Example:

```
# clia setpefconfig startup_delay 45

Pigeon Point Shelf Manager Command Line Interpreter

PEF startup delay set successfully

#
```

3.42.4. *alert_startup_delay*

Syntax:

setpefconfig startup_delay <value> | setpefconfig 4 <value>

Purpose:

This command sets the current value of the PEF parameter "alert_startup_delay". This parameter is a single byte, representing the number of seconds that the alerting facility delays at startup. The value is specified as a decimal number of seconds.

Example:

```
# clia setpefconfig alert_startup_delay 45

Pigeon Point Shelf Manager Command Line Interpreter

Alert startup delay set successfully

#
```

3.42.5. *event_filter*

Syntax:

setpefconfig event_filter <set-selector> <value1> … <value19> |
setpefconfig 6 <set-selector> <value1> … <value19>

Purpose:

This command sets the element of the event filter table with the index equal to <set-selector>. Indexes are 1-based. The contents of the new element are specified by 19 numeric values <value1> to <value19>, in hexadecimal, encoded according to the definition in table 15-2 of the IPMI specification version 1.5:

- filter configuration
- event filter action
- alert policy number
- event severity
- generator ID byte 1
- generator ID byte 2
- sensor type
- sensor number
- event trigger (event/reading type)
- event data 1 event offset mask
- event data 1 AND mask
- event data 1 compare 1
- event data 1 compare 2
- event data 2 AND mask
- event data 2 compare 1
- event data 2 compare 2
- event data 3 AND mask
- event data 3 compare 1
- event data 3 compare 2

Example:

Setting event filter 2 to trigger an alert action when an IPM Controller at address 9C, FRU 0, reaches state M0 (the alert will be sent according with the Alert Policy #1):

```
# clia setpefconfig event_filter 2 80 1 1 10 9C FF F0 FF FF FF FF 0F FF 0 0 0 0 FF FF
0

Pigeon Point Shelf Manager Command Line Interpreter

Event filter set successfully

#
```

3.42.6. event_filter_data1

Syntax:

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

Purpose:

This command sets the first byte of the element of the event filter table with the index equal to <set-selector>. Indexes are 1-based. This byte should be specified in hexadecimal. Bits in this byte have the following meaning:

0x80	This filter is enabled
0x40	This filter is 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.

Example:

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

#
```

3.42.7. alert_policy

Syntax:

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

Purpose:

This command sets an alert policy table entry identified by the specified set selector. The contents of the new element are specified by the following 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

Example:

The following example sets up the alert policy table entry 2 with the following attributes:

- Policy number=5,
- Enabled,
- policy=always send alert to this destination
- Destination channel = 1
- Destination selector = 1
- Alert String selector = use string 1 for all events.

```
# clia setpefconfig alert_policy 2 5 8 1 1 1

Pigeon Point Shelf Manager Command Line Interpreter

Policy set successfully

#
```

3.42.8. system_guid

Syntax:

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

Purpose:

This command sets the current value of the PEF parameter "system_guid". This parameter represents the GUID that is sent in a PET Trap PDU to an alert destination. This GUID may be defined as a separate GUID or as being equal to the System GUID.

The <guid-value> can be specified as an actual GUID, conforming to the standard GUID format "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx", or as a symbolic value "none" In the first case, the PEF facility uses the specified GUID in PET Traps. In the second case, the PEF facility defaults to the System GUID (the result of the IPMI Get System GUID command) for PET Traps.

Example:

```
# 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

#
```

3.42.9. *alert_string_key*

Syntax:

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

Purpose:

This command sets the element of the alert string key table with the index equal to <set-selector>. Indexes are 1-based. Set-selector 0 can be used to designate the volatile alert string. Each key associates an event filter with an alert string for the alert generation purposes, and consists of the event filter number and the alert string number. Both values are 8-bit values and are specified by the parameters value1 and value2 respectively, in hexadecimal.

Example:

```
# clia setpefconfig alert_string_key 2 10 11
```

```
Pigeon Point Shelf Manager Command Line Interpreter

Alert string keys set successfully

#
```

3.42.10. alert_string

Syntax:

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

Purpose:

This command sets the element of the alert string table with the index equal to <set-selector>. Indexes are 1-based. Index 0 can be used to designate the volatile alert string. The string value should be enclosed in double quotes (") and may contain special characters and line feeds inside.

Example:

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

Pigeon Point Shelf Manager Command Line Interpreter

Alert string set successfully

#
```

3.42.11. oem_filter

Syntax:

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

Purpose:

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

Example:

```
# clia getpefconfig oem_filter

Pigeon Point Shelf Manager Command Line Interpreter

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

#
# clia setpefconfig oem_filter 4 0xdc 0xf3 5

Pigeon Point Shelf Manager Command Line Interpreter

OEM filter set successfully

# clia getpefconfig oem_filter

Pigeon Point Shelf Manager Command Line Interpreter

    Active OEM Filters:
    0x01: OEM range boundary 0xff:0xff, alert policy # 1
    0x04: OEM range boundary 0xdc:0xf3, alert policy # 5

#
```

3.43. setpowerlevel

Syntax:

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

Instead of the <IPMB-address> the user may use:

board <N>

shm <N>

Purpose:

This command allows controlling the power level of a FRU and results in the Shelf Manager issuing a "Set Power Level" command on IPMB-0 to the designated IPM controller. Since the Shelf Manager core is responsible for managing power levels and tracking the corresponding power budgets, this command must be used with extreme care, especially when specifying a non-zero power level. Users of this command must be thoroughly familiar with the AdvancedTCA power management architecture as defined in the AdvancedTCA specification.

The target FRU is specified by the IPMB address of its IPM controller, plus the FRU device ID. 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.

Example:

Turn off power for the board at 0x84:

```
# clia setpowerlevel 84 0 OFF

Pigeon Point Shelf Manager Command Line Interpreter

    Operation completed with status = 0x0
```

3.44. setsensoreventenable

Syntax:

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

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

Instead of the <IPMB-address> the user may use:

board <N>

shm <N>

Purpose:

This command changes the event enable mask for the specified sensor. The sensor is specified by the IPMB address of the owning IPM controller and the sensor name or number. Alternatively, the board number or dedicated Shelf Manager number can be used to designate the target IPM controller.

This command allows the user to qualify the sensor number with the Logical Unit Number (LUN) if the target controller supports sensors on multiple LUNs. <Lun> can take value 0, 1 or 3. (LUN 2 is reserved.) If the LUN is omitted, the command is applied to the sensor with the specified sensor number on the lowest LUN. (For example, if the command specifies sensor 3 without explicit LUN qualification, and the target controller exposes sensor 3 on LUN 1 and another sensor 3 on LUN 3, the command is applied to the sensor 3 on LUN 1.)

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.

Example:

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

```
# clia setsensoreventenable fe "Local Temp" 0x90 0x01 0x00

Pigeon Point Shelf Manager Command Line Interpreter

    Event enable mask set successfully
#

# clia getsensoreventenable -v fe "Local Temp"
```

```
Pigeon Point Shelf Manager Command Line Interpreter

fe: LUN: 0, Sensor # 3 ("Local Temp")
    Type: Threshold (0x01), "Temperature" (0x01)
    Sensor scanning disabled
    Assertion event mask:   0x0001
        Assertion event for "Lower Non-Critical Going Low" enabled
    Deassertion event mask: 0x0000
#
```

Perform the same operation on the same sensor, but specify the sensor using LUN and sensor number:

```
# clia setsensoreventenable fe 0:3 0x90 0x01 0x00

Pigeon Point Shelf Manager Command Line Interpreter

    Event enable mask set successfully
#
```

3.45. setthreshold

Syntax:

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

Instead of <IPMB-address> the user may use:

board <N>

shm <N>

Purpose:

This command changes the current threshold value for the specified threshold of the specified sensor. The sensor is specified by the IPMB address of the owning IPM controller and the sensor name or number. The target sensor must be a threshold-based sensor. The parameter <threshold-type> can be specified as one of the following symbolic values:

- "upper_non_recoverable" (can be abbreviated to "unr")
- "upper_critical" (can be abbreviated to "uc")
- "upper_non_critical" (can be abbreviated to "unc")
- "lower_non_recoverable" (can be abbreviated to "lnr")
- "lower_critical" (can be abbreviated to "lc")
- "lower_non_critical" (can be abbreviated to "lnc")

By default, the target value is specified in processed form (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.

Example:

Set the upper non-critical threshold value for the temperature sensor "emulated temp" on IPM controller 9C to 99 degrees Celsius.

```
# clia threshold 9c 2

Pigeon Point Shelf Manager Command Line Interpreter

9c: LUN: 0, Sensor # 2 ("emulated temp")
    Type: Threshold (0x01), "Temperature" (0x01)
        Lower Non-Critical Threshold, Raw Data: 0x03, Processed Data: 3.000000
degrees C
        Lower Critical Threshold, Raw Data: 0x14, Processed Data: 20.000000 degrees C
        Lower Non-Recoverable Threshold, Raw Data: 0xfb, Processed Data: -5.000000
degrees C
        Upper Non-Critical Threshold, Raw Data: 0x46, Processed Data: 70.000000
degrees C
        Upper Critical Threshold, Raw Data: 0x50, Processed Data: 80.000000 degrees C
        Upper Non-Recoverable Threshold, Raw Data: 0x5a, Processed Data: 90.000000
degrees C

#

# clia setthreshold 9c 0:2 unc 99

Pigeon Point Shelf Manager Command Line Interpreter

    Threshold set successfully

#

# clia threshold 9c 0:2

Pigeon Point Shelf Manager Command Line Interpreter

9c: LUN: 0, Sensor # 2 ("emulated temp")
    Type: Threshold (0x01), "Temperature" (0x01)
        Lower Non-Critical Threshold, Raw Data: 0x03, Processed Data: 3.000000
degrees C
        Lower Critical Threshold, Raw Data: 0x14, Processed Data: 20.000000 degrees C
        Lower Non-Recoverable Threshold, Raw Data: 0xfb, Processed Data: -5.000000
degrees C
        Upper Non-Critical Threshold, Raw Data: 0x63, Processed Data: 99.000000
degrees C
        Upper Critical Threshold, Raw Data: 0x50, Processed Data: 80.000000 degrees C
        Upper Non-Recoverable Threshold, Raw Data: 0x5a, Processed Data: 90.000000
degrees C

#
```

3.46. shelf

Syntax:

```
shelf <subcommand>

 The following subcommands are supported.
            address_table
           cooling_state
           fans_state
           power_distribution
           power_management
           pci_connectivity
           ha_connectivity
           h110_connectivity
           point-to-point_connectivity
           MaxCurrent [feed] <Amps>
           MinVoltage [feed] <Volts>
            Activation <addr> <fru_id> 1|0
           Deactivation <addr> <fru_id> 1|0
            BDSelGrounded <slot number> 1|0
                   1 means Enabled, 0 means Disabled
            PwrCapability <addr> <fru_id> <Watts>
            PwrDelay <addr> <fru_id> <10ths_of_second>
            Allowance <seconds>
            PwrReorder <addr1> <fru_id1> before|after <addr2> <fru_id2>
            info_refresh
             info_force_update
```

Purpose:

The command "shelf" shows key Shelf FRU information, plus selected current operating data for the shelf, and allows modifying some fields in the Shelf FRU information. The type of the information this command shows or modifies is specified in the command parameter.

The following subsections describe the syntax of the "shelf" command for different applications of this command.

3.46.1. Displaying Shelf FRU Information

Syntax:

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

Purpose:

This syntax of the command "shelf" shows key Shelf FRU information, plus selected current operating data for the shelf. The type of the information this command shows is specified in the command parameter. The following table lists the parameters supported by the "shelf" command:

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

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

For the command parameters "cooling_state", "fans_state" and "power_management", the verbosity option "-v" is available. It should be entered before the command parameter: "clia shelf –v cooling_state". If used, the command with parameter either "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” option displays the amount of power currently assigned to each of the FRUs covered by FRU Power Descriptors in the Shelf FRU Info.

Example:

Get shelf cooling status.

```
# clia shelf cooling_state

Pigeon Point Shelf Manager Command Line Interpreter

    Cooling state: "Normal"

#
```

Get shelf fan tachometer status (verbose).

```
# clia shelf –v fans_state

Pigeon Point Shelf Manager Command Line Interpreter

    Fans state: "Major Alert"
    Sensor(s) at this state: (0x7e,10) (0x7e,11) (0x7e,12) (0x7e,13)
                             (0x7e,14) (0x7e,15) (0x7e,16) (0x7e,17)
#
```

Get address table.

```
# clia shelf address_table

Pigeon Point Shelf Manager Command Line Interpreter

    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
Pigeon Point
#
```

3.46.2. Modifying Maximum External Available Current

Syntax:

shelf maxcurrent [<feed>] <current>

Purpose:

This command sets the Maximum External Available Current for the specified feed number and updates all known instances of Shelf FRU 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 description of that feed.

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

Example:

Changing the Maximum Available External Current for Feed 0 from 44 Amps to 99 Amps.

```
# clia shelf pd

Pigeon Point Shelf Manager Command Line Interpreter

    PICMG Shelf Power Distribution Record (ID=0x11)
        Version = 0
    Feed count: 1
    Feed 00:
        Maximum External Available Current: 50.0 Amps
        Maximum Internal Current: Not specified
        Minimum Expected Operating Voltage: -40.5 Volts
        Actual Power Available: 2025.000 Watts
        Currently Used Power: 200.000 Watts
        Feed-to-FRU Mapping entries count: 16
           FRU Addr: 41, FRU ID: 0xfe
           FRU Addr: 42, FRU ID: 0xfe
           FRU Addr: 43, FRU ID: 0xfe
           FRU Addr: 44, FRU ID: 0xfe
           FRU Addr: 45, FRU ID: 0xfe
           FRU Addr: 46, FRU ID: 0xfe
           FRU Addr: 47, FRU ID: 0xfe
           FRU Addr: 48, FRU ID: 0xfe
           FRU Addr: 49, FRU ID: 0xfe
           FRU Addr: 4a, FRU ID: 0xfe
           FRU Addr: 4b, FRU ID: 0xfe
           FRU Addr: 4c, FRU ID: 0xfe
           FRU Addr: 4d, FRU ID: 0xfe
           FRU Addr: 4e, FRU ID: 0xfe
           FRU Addr: 4f, FRU ID: 0xfe
           FRU Addr: 50, FRU ID: 0xfe

# clia shelf maxcurrent 0 99

Pigeon Point Shelf Manager Command Line Interpreter

    Updating Shelf FRU Info

    Cached information updated

# clia shelf pd

Pigeon Point Shelf Manager Command Line Interpreter

    PICMG Shelf Power Distribution Record (ID=0x11)
        Version = 0
    Feed count: 1
    Feed 00:
        Maximum External Available Current: 99.0 Amps
        Maximum Internal Current: Not specified
        Minimum Expected Operating Voltage: -40.5 Volts
        Actual Power Available: 2025.000 Watts
        Currently Used Power: 200.000 Watts
        Feed-to-FRU Mapping entries count: 16
           FRU Addr: 41, FRU ID: 0xfe
           FRU Addr: 42, FRU ID: 0xfe
           FRU Addr: 43, FRU ID: 0xfe
           FRU Addr: 44, FRU ID: 0xfe
           FRU Addr: 45, FRU ID: 0xfe
           FRU Addr: 46, FRU ID: 0xfe
           FRU Addr: 47, FRU ID: 0xfe
           FRU Addr: 48, FRU ID: 0xfe
           FRU Addr: 49, FRU ID: 0xfe
           FRU Addr: 4a, FRU ID: 0xfe
           FRU Addr: 4b, FRU ID: 0xfe
           FRU Addr: 4c, FRU ID: 0xfe
           FRU Addr: 4d, FRU ID: 0xfe
           FRU Addr: 4e, FRU ID: 0xfe
           FRU Addr: 4f, FRU ID: 0xfe
           FRU Addr: 50, FRU ID: 0xfe

#
```

3.46.3. Modifying Minimum Expected Operating Voltage

Syntax:

shelf minvoltage [<feed>] <voltage>

Purpose:

This command sets the Minimum Expected Operating Voltage for the specified feed number and updates all known Shelf FRU 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.

Example:

Changing the Minimum Expected Operating Voltage for the Feed 0

```
# clia shelf pd

Pigeon Point Shelf Manager Command Line Interpreter

    PICMG Shelf Power Distribution Record (ID=0x11)
        Version = 0
    Feed count: 1
    Feed 00:
        Maximum External Available Current: 99.0 Amps
        Maximum Internal Current: Not specified
        Minimum Expected Operating Voltage: -40.5 Volts
        Actual Power Available: 2025.000 Watts
        Currently Used Power: 200.000 Watts
        Feed-to-FRU Mapping entries count: 16
           FRU Addr: 41, FRU ID: 0xfe
           FRU Addr: 42, FRU ID: 0xfe
           FRU Addr: 43, FRU ID: 0xfe
           FRU Addr: 44, FRU ID: 0xfe
           FRU Addr: 45, FRU ID: 0xfe
           FRU Addr: 46, FRU ID: 0xfe
           FRU Addr: 47, FRU ID: 0xfe
           FRU Addr: 48, FRU ID: 0xfe
           FRU Addr: 49, FRU ID: 0xfe
           FRU Addr: 4a, FRU ID: 0xfe
           FRU Addr: 4b, FRU ID: 0xfe
           FRU Addr: 4c, FRU ID: 0xfe
           FRU Addr: 4d, FRU ID: 0xfe
           FRU Addr: 4e, FRU ID: 0xfe
           FRU Addr: 4f, FRU ID: 0xfe
           FRU Addr: 50, FRU ID: 0xfe

#

# clia shelf minvoltage 0 -59

Pigeon Point Shelf Manager Command Line Interpreter
```

```
    Updating Shelf FRU Info

    Cached information updated

#

# clia shelf pd

Pigeon Point Shelf Manager Command Line Interpreter

    PICMG Shelf Power Distribution Record (ID=0x11)
        Version = 0
    Feed count: 1
    Feed 00:
        Maximum External Available Current: 99.0 Amps
        Maximum Internal Current: Not specified
        Minimum Expected Operating Voltage: -59.0 Volts
        Actual Power Available: 2025.000 Watts
        Currently Used Power: 200.000 Watts
        Feed-to-FRU Mapping entries count: 16
           FRU Addr: 41, FRU ID: 0xfe
           FRU Addr: 42, FRU ID: 0xfe
           FRU Addr: 43, FRU ID: 0xfe
           FRU Addr: 44, FRU ID: 0xfe
           FRU Addr: 45, FRU ID: 0xfe
           FRU Addr: 46, FRU ID: 0xfe
           FRU Addr: 47, FRU ID: 0xfe
           FRU Addr: 48, FRU ID: 0xfe
           FRU Addr: 49, FRU ID: 0xfe
           FRU Addr: 4a, FRU ID: 0xfe
           FRU Addr: 4b, FRU ID: 0xfe
           FRU Addr: 4c, FRU ID: 0xfe
           FRU Addr: 4d, FRU ID: 0xfe
           FRU Addr: 4e, FRU ID: 0xfe
           FRU Addr: 4f, FRU ID: 0xfe
           FRU Addr: 50, FRU ID: 0xfe

#
```

3.46.4. Modifying Shelf Manager Controlled Activation Flag

Syntax[1]:

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]

[1] Note: the reference notation power_supply <N>, plus its abbreviation, is supported only in CompactPCI shelves.

shelf activation fan_tray <N> [1/0]

Purpose:

This command displays or changes the Shelf Manager Controlled Activation field for the specified FRU of the specified IPM controller. The command modifies the Shelf Manager Controlled Activation flag only for already existing entries in the Shelf Activation and Power Management record. This command also updates the cached version of the Shelf FRU Information used by the Shelf Manager. Thus the new value of the Shelf Manager Controlled Activation field becomes effective immediately without the need to restart the Shelf Manager.

The parameter <hardware addr> is the 7-bit hardware address in hexadecimal format.

The parameter <fru id> is a FRU ID in hexadecimal format; 0xFE means ALL FRUs at that hardware address.

Example:

Enabling Shelf Manager Controlled Activation on an IPM Controller with hardware address 0x42 (IPMB address 0x84).

```
# clia shelf pm

Pigeon Point Shelf Manager Command Line Interpreter

    PICMG Shelf Activation And Power Management Record (ID=0x12)
        Version = 0
    Allowance for FRU Activation Readiness: 10 seconds
    FRU Activation and Power Description Count: 16
    Hw Address: 41, FRU ID: 0xfe, Maximum FRU Power Capabilities: 150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 42, FRU ID: 0xfe, Maximum FRU Power Capabilities: 150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 43, FRU ID: 0xfe, Maximum FRU Power Capabilities: 150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 44, FRU ID: 0xfe, Maximum FRU Power Capabilities: 150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 45, FRU ID: 0xfe, Maximum FRU Power Capabilities: 150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 46, FRU ID: 0xfe, Maximum FRU Power Capabilities: 150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 47, FRU ID: 0xfe, Maximum FRU Power Capabilities: 150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 48, FRU ID: 0xfe, Maximum FRU Power Capabilities: 150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 49, FRU ID: 0xfe, Maximum FRU Power Capabilities: 150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 4a, FRU ID: 0xfe, Maximum FRU Power Capabilities: 150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 4b, FRU ID: 0xfe, Maximum FRU Power Capabilities: 150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 4c, FRU ID: 0xfe, Maximum FRU Power Capabilities: 150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 4d, FRU ID: 0xfe, Maximum FRU Power Capabilities: 150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 4e, FRU ID: 0xfe, Maximum FRU Power Capabilities: 150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 4f, FRU ID: 0xfe, Maximum FRU Power Capabilities: 150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 50, FRU ID: 0xfe, Maximum FRU Power Capabilities: 150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds


#

# clia shelf activation 42 0xfe 0

Pigeon Point Shelf Manager Command Line Interpreter

    Updating Shelf FRU Info, address: 0x42, FRU ID # 254
    Cached information updated
    Wrote Information to the Shelf FRU

#

# clia shelf pm

Pigeon Point Shelf Manager Command Line Interpreter
```

```
PICMG Shelf Activation And Power Management Record (ID=0x12)
    Version = 0
Allowance for FRU Activation Readiness: 10 seconds
FRU Activation and Power Description Count: 16
Hw Address: 41, FRU ID: 0xfe, Maximum FRU Power Capabilities: 150 Watts
    Shelf Manager Controlled Activation: Enabled
    Delay Before Next Power On: 0.0 seconds

Hw Address: 42, FRU ID: 0xfe, Maximum FRU Power Capabilities: 150 Watts
    Shelf Manager Controlled Activation: Disabled
    Delay Before Next Power On: 0.0 seconds

Hw Address: 43, FRU ID: 0xfe, Maximum FRU Power Capabilities: 150 Watts
    Shelf Manager Controlled Activation: Enabled
    Delay Before Next Power On: 0.0 seconds

Hw Address: 44, FRU ID: 0xfe, Maximum FRU Power Capabilities: 150 Watts
    Shelf Manager Controlled Activation: Enabled
    Delay Before Next Power On: 0.0 seconds

Hw Address: 45, FRU ID: 0xfe, Maximum FRU Power Capabilities: 150 Watts
    Shelf Manager Controlled Activation: Enabled
    Delay Before Next Power On: 0.0 seconds

Hw Address: 46, FRU ID: 0xfe, Maximum FRU Power Capabilities: 150 Watts
    Shelf Manager Controlled Activation: Enabled
    Delay Before Next Power On: 0.0 seconds

Hw Address: 47, FRU ID: 0xfe, Maximum FRU Power Capabilities: 150 Watts
    Shelf Manager Controlled Activation: Enabled
    Delay Before Next Power On: 0.0 seconds

Hw Address: 48, FRU ID: 0xfe, Maximum FRU Power Capabilities: 150 Watts
    Shelf Manager Controlled Activation: Enabled
    Delay Before Next Power On: 0.0 seconds

Hw Address: 49, FRU ID: 0xfe, Maximum FRU Power Capabilities: 150 Watts
    Shelf Manager Controlled Activation: Enabled
    Delay Before Next Power On: 0.0 seconds

Hw Address: 4a, FRU ID: 0xfe, Maximum FRU Power Capabilities: 150 Watts
    Shelf Manager Controlled Activation: Enabled
    Delay Before Next Power On: 0.0 seconds

Hw Address: 4b, FRU ID: 0xfe, Maximum FRU Power Capabilities: 150 Watts
    Shelf Manager Controlled Activation: Enabled
    Delay Before Next Power On: 0.0 seconds

Hw Address: 4c, FRU ID: 0xfe, Maximum FRU Power Capabilities: 150 Watts
    Shelf Manager Controlled Activation: Enabled
    Delay Before Next Power On: 0.0 seconds

Hw Address: 4d, FRU ID: 0xfe, Maximum FRU Power Capabilities: 150 Watts
    Shelf Manager Controlled Activation: Enabled
    Delay Before Next Power On: 0.0 seconds

Hw Address: 4e, FRU ID: 0xfe, Maximum FRU Power Capabilities: 150 Watts
```

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

    Hw Address: 4f, FRU ID: 0xfe, Maximum FRU Power Capabilities: 150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds

    Hw Address: 50, FRU ID: 0xfe, Maximum FRU Power Capabilities: 150 Watts
        Shelf Manager Controlled Activation: Enabled
        Delay Before Next Power On: 0.0 seconds


#
```

3.46.5. Modifying Shelf Manager Controlled DeActivation Flag[1]

Syntax[2]:

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 fan_tray <N> [1/0]

Purpose:

This command displays or changes the Shelf Manager Controlled Deactivation field for the specified FRU of the specified IPM controller. The command modifies the Shelf Manager Controlled Deactivation flag only for already existing entries in the Shelf 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.

Example:

Enabling Shelf Manager Controlled Deactivation on an IPM Controller with hardware address 0x42 (IPMB address 0x84).

[1] This command is relevant to CompactPCI shelves only.

[2] Note: the reference notation power_supply <N>, plus its abbreviation, is supported only in CompactPCI shelves.

```
# 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: Disabled
        Delay Before Next Power On: 2.2 seconds

#

# clia shelf activation 42 0xfe 1

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

#
```

3.46.6. *Modifying Shelf Manager BDSelGrounded Flag*[1]

Syntax:

shelf bdselgrounded <slot number>[1/0]

[1] This command is relevant to CompactPCI shelves only.

shelf bdselgrounded board <N> [1/0]

shelf bdselgrounded board all [1/0]

Purpose:

This command allows 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, and also it makes it impossible to turn on/off power for this slot from the Shelf Manager. 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 both decimal (11) and hexadecimal (0x0B) format.

Example:

Setting normal BD SEL# signal operation for the 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"

#
```

3.46.7. Modifying Maximum FRU Power Capability[1]

Syntax[2]:

shelf pwrcapability <hardware addr> <fru id> <value>

shelf pwrcapability board <N> <value>

shelf pwrcapability power_supply <N> <value>

shelf pwrcapability fan_tray <N> <value>

[1] This command is relevant to AdvancedTCA shelves only.

[2] Note: the reference notation power_supply <N>, plus its abbreviation, is supported only in CompactPCI shelves.

Purpose:

This command changes the Maximum FRU Power Capability field for the specified FRU of the specified IPM controller. Note: never set the Maximum FRU Power Capability field to a larger value than is safe for your shelf environment.

The command modifies this field only for already existing entries in the Shelf Activation and Power Management record. This command also updates the cached version of the Shelf FRU Information used by the Shelf Manager. Thus the new value of the Maximum FRU Power Capability field becomes effective immediately without the need to restart the Shelf Manager.

The parameter <hardware addr> is the 7-bit hardware address in hexadecimal format.

The parameter <fru id> is a FRU ID in hexadecimal format; 0xFE means ALL FRUs at that hardware address.

The parameter <value> is the new value for the field in Watts. The range of the possible values is 0..65535.

Example:

Setting Maximum FRU Power Capability on an IPM Controller with hardware address 0x42 (IPMB address 0x84) to 150 Watts.

```
# clia shelf pm

Pigeon Point Shelf Manager Command Line Interpreter
    Power Management:
    Allowance for FRU Activation Readiness: 10 seconds
    FRU Activation and Power Description Count: 2
    Hw Address: 41, FRU ID: fe, Maximum FRU Power Capabilities: 200 Watts
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 2.2 seconds

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

#

# clia shelf pwrcapability 42 0xfe 150

Pigeon Point Shelf Manager Command Line Interpreter

    Updating Shelf FRU Info
    Cached information updated
```

```
#

# clia shelf pm

Pigeon Point Shelf Manager Command Line Interpreter

    Power Management:
    Allowance for FRU Activation Readiness: 10 seconds
    FRU Activation and Power Description Count: 2
    Hw Address: 41, FRU ID: fe, Maximum FRU Power Capabilities: 200 Watts
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 2.2 seconds

    Hw Address: 42, FRU ID: fe, Maximum FRU Power Capabilities: 150 Watts
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 2.2 seconds



#
```

3.46.8. Modifying Delay Before Next Power On[1]

Syntax[2]:

shelf pwrdelay <hardware addr> <fru id> <value>

shelf pwrdelay board <N> <value>

shelf pwrdelay power_supply <N> <value>

shelf pwrdelay fan_tray <N> <value>

Purpose:

This command changes the Delay Before Next Power On field for the specified FRU of the specified IPM controller. The command modifies this field only for already existing entries in the Shelf Activation and Power Management record. This command also updates the cached version of the Shelf FRU Information used by the Shelf Manager. Thus the new value of the Delay Before Next Power On field becomes effective immediately without the need to restart the Shelf Manager.

The parameter <hardware addr> is the 7-bit hardware address in hexadecimal format.

The parameter <fru id> is a FRU ID in hexadecimal format; 0xFE means ALL FRUs at that hardware address.

[1] This command is relevant to AdvancedTCA shelves only.

[2] Note: the reference notation power_supply <N>, plus its abbreviation, is supported only in CompactPCI shelves.

The parameter <value> is the new value for the field in tenths of a second. The range of the possible values is 0..63.

Example:

Setting Delay Before Next Power On for an IPM Controller with hardware address 0x42 (IPMB address 0x84) to 5 seconds.

```
# clia shelf pm

Pigeon Point Shelf Manager Command Line Interpreter

    Power Management:
    Allowance for FRU Activation Readiness: 10 seconds
    FRU Activation and Power Description Count: 2
    Hw Address: 41, FRU ID: fe, Maximum FRU Power Capabilities: 200 Watts
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 2.2 seconds

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

#

# clia shelf pwrdelay 42 0xfe 50

Pigeon Point Shelf Manager Command Line Interpreter

    Updating Shelf FRU Info
    Cached information updated

#

# clia shelf pm

Pigeon Point Shelf Manager Command Line Interpreter

    Power Management:
    Allowance for FRU Activation Readiness: 10 seconds
    FRU Activation and Power Description Count: 2
    Hw Address: 41, FRU ID: fe, Maximum FRU Power Capabilities: 200 Watts
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 2.2 seconds

    Hw Address: 42, FRU ID: fe, Maximum FRU Power Capabilities: 200 Watts
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 5.0 seconds

#
```

3.46.9. *Modifying Allowance for FRU Activation Readiness*[1]

Syntax:

shelf allowance <value>

Purpose:

This command changes the Allowance for FRU Activation Readiness parameter.

The parameter <value> is the new value for the parameter in seconds. The range of the possible values is 0..255.

Example:

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
```

[1] This command is relevant to AdvancedTCA shelves only.

```
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


#
```

3.46.10. Reorder the FRU Activation and Power Descriptors[1]

Syntax[2]:

shelf pwrreorder <hardware addr 1> <fru id 1> before/after <hardware addr 2> <fru id 2>

As usual, <hw address> <fru id> can be replaced by the following:

board <N>

power_supply <N>

fan_tray <N>

Purpose:

This command changes the order of the FRU Activation and Power Descriptors in the Shelf FRU Information. The command can reorder only the already existing descriptors. The current implementation is also limited to reordering the descriptors only inside a single Shelf Activation and Power Management record. This command also updates the cached version of the Shelf FRU Information used by the Shelf Manager. Thus the new order of the descriptors becomes effective immediately without the need to restart the Shelf Manager.

The parameter <hardware addr 1> is the 7-bit hardware address in hexadecimal format of the descriptor that needs to be moved 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.

[1] This command is relevant to AdvancedTCA shelves only.

[2] Note: the reference notation power_supply <N>, plus its abbreviation, is supported only in CompactPCI shelves.

The parameter <hardware addr 2> is the 7-bit hardware address in hexadecimal format of the descriptor, before/after which the <hardware addr1>/<fru id 1> descriptor should be placed.

The parameter <fru id 2> is a FRU ID in hexadecimal format of the descriptor, before/after which the <hardware addr1>/<fru id 1> descriptor should be placed.

Example:

Placing the descriptor for an IPM Controller with hardware address 0x42 (IPMB address 0x84) before the descriptor for an IPM Controller with hardware address 0x41 (IPMB address 0x82).

```
# clia shelf pm

Pigeon Point Shelf Manager Command Line Interpreter

    Power Management:
    Allowance for FRU Activation Readiness: 10 seconds
    FRU Activation and Power Description Count: 2
    Hw Address: 41, FRU ID: fe, Maximum FRU Power Capabilities: 200 Watts
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 2.2 seconds

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

#

# clia shelf pwrreorder 42 0xfe before 41 0xfe

Pigeon Point Shelf Manager Command Line Interpreter

    Updating Shelf FRU Info
    Cached information updated

#

# clia shelf pm

Pigeon Point Shelf Manager Command Line Interpreter

    Power Management:
    Allowance for FRU Activation Readiness: 10 seconds
    FRU Activation and Power Description Count: 2
    Hw Address: 42, FRU ID: fe, Maximum FRU Power Capabilities: 200 Watts
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 2.2 seconds

    Hw Address: 41, FRU ID: fe, Maximum FRU Power Capabilities: 200 Watts
        Shelf Manager Controlled Activation: Disabled
        Delay Before Next Power On: 2.2 seconds

#
```

3.46.11. Refresh the Shelf FRU Info[1]

Syntax:

shelf info_refresh

Purpose:

This command causes the Shelf Manager to re-read the previously found sources of Shelf FRU Information in the shelf and reassess which of the sources contain valid Shelf FRU Information. Assuming that valid Shelf FRU Information is confirmed, all of the Shelf FRU Information storage devices and the cached master copy of the Shelf FRU Information are updated with the contents of the new Shelf FRU Information.

As specified by PICMG 3.0 (section 3.6.4), the Shelf Manager tries to find possible Shelf FRU Information storage devices during initialization. If the Shelf Manager finds at least two FRU Information devices that contain valid Shelf FRU Information, the Shelf Manager performs an "election" to determine which Shelf FRU Information sources to use. This election is based on validating the data the storage devices contain and comparing the contents. After a successful election, the Shelf Manager creates a cached master copy of the Shelf FRU 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 reboot of the Shelf Manager.

Example:

Successful refresh: two matching sources of the Shelf FRU Info.

```
# clia shelf info_refresh

Pigeon Point Shelf Manager Command Line Interpreter

Read 0x20 # 2, size = 1024
Read 0x20 # 1, size = 1024
Found 2 Matching Shelf FRU Info

0x20 # 2, size = 1024 (data size = 775), "Valid" Shelf FRU, "Matching"
0x20 # 1, size = 1024 (data size = 775), "Valid" Shelf FRU, "Matching"
Shelf FRU Info was not changed

#
```

[1] This command may have additional shelf-specific aspects that are important to understand. Please check the documentation provided by your shelf vendor for any such details.

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

#
```

3.46.12. Update the Shelf FRU Info Storage Devices[1]

Syntax:

shelf info_force_update

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.

Example:

```
# clia shelf info_force_update

Pigeon Point Shelf Manager Command Line Interpreter

Starting the Shelf FRU Info source device update

#
```

[1] This command may have additional shelf-specific aspects that are important to understand. Please check the documentation provided by your shelf vendor for any such details.

3.47. shelfaddress

Syntax:

shelfaddress [<up to 30 characters of the shelf address>]

Purpose:

This command gets or sets the Shelf Address field of the Address Table within Shelf FRU Info. This command uses 6-bit packed values, so CAPITAL letters and digits are allowed only.

Lower case letters are automatically capitalized.

Example:

```
# 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"

#
```

3.48. shmstatus

Syntax:

shmstatus

Purpose:

This command returns the Shelf Manager status in redundant configurations: Active or Backup. In verbose mode it reports a more detailed picture: status of the Shelf FRU 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 ftom the active Shelf Manager.

Example:

```
# clia shmstatus -v

Pigeon Point Shelf Manager Command Line Interpreter

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

#
```

3.49. showunhealthy

Syntax:

showunhealthy

Purpose:

This command shows the list of FRUs that appear to have a problem. In the PICMG 3.0 context, this list includes FRUs for which the cause of last hot swap state change is “Communication Lost”, “Communication lost due to local failure”, “Unexpected deactivation”. 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

Example:

Show the list of unhealthy components in the system.

```
# clia showunhealthy

Pigeon Point Shelf Manager Command Line Interpreter

There are no unhealthy components in the shelf.

#
```

3.50. switchover

Syntax:

switchover

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.

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.

#
```

3.51. terminate

Syntax:

terminate [-reboot]

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, if this variant is supported by the ShMM on which the command is executed. Currently, no-reboot variant of the command is supported on the ShMM-500 and not supported on the ShMM-300.

If the ShMM does not support terminating the Shelf Manager without reboot, the ShMM is rebooted.

Examples:

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

```
# clia terminate
    Terminating the Shelf Manager.

#
```

3.52. user

Syntax:

user [<subcommand>]
The following subcommands are supported:
add
delete
enable
name
passwd
channel

Purpose:

The "user" command shows information about the RMCP user accounts on the Shelf Manager and provides a simple way to add, delete and modify the user accounts.

The following subsections describe the syntax of the "user" command for different applications of this command.

3.52.1. Displaying User Information

Syntax:

user [-v] [<user id>]

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.

Example:

```
# 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"
```

3.52.2. *Adding a New User*

Syntax:

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

Purpose:

This command adds a new user to the system. It sets the same maximum privilege level and channel access flags for all channels, as specified in the command. The command will return an error if the specified user does not exist. Command parameters have the following meaning:

user_id - is a valid user id;
user_name - is the user name (it will be truncated to the 16 characters without any 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 the user password (it will be truncated to the 16 characters without any notice)

Example:

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 successfuly

# clia user

Pigeon Point Shelf Manager Command Line Interpreter

    1: ""
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"

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

3.52.3. Deleting a User

Syntax:

user delete <user id>

Purpose:

This command deletes the user specified by the user id.

Example:

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 successfuly


# clia user

Pigeon Point Shelf Manager Command Line Interpreter

    1: ""
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"

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

3.52.4. Enabling and Disabling a User

Syntax:

user enable <user id> 1 | 0

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;
non-zero	- enable the specified user

Example:

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 successfuly

# clia user –v

Pigeon Point Shelf Manager Command Line Interpreter

    1: ""
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"

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

# clia user enable 9 1

Pigeon Point Shelf Manager Command Line Interpreter

    User 9 enabled successfuly


# clia user

Pigeon Point Shelf Manager Command Line Interpreter

    1: ""
        Channels 0-15 Privilege level: "Administrator"
            Flags: "IPMI Messaging"

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

3.52.5. *Modifying a User Name*

Syntax:

user name <user id> <user name>

Purpose:

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

user_id - is the valid user id ;
user_name - is the user name (which will be truncated to 16 characters without any notice);

Example:

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 successfuly

# 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"
```

3.52.6. Modifying a User's Password

Syntax:

user passwd <user id> <password>

Purpose:

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

user_id - is the valid user id;

password - is the user password (which will be truncated to 16 characters without any notice);

Example:

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"
```

3.52.7. *Modify Channel Access Settings for a Specified User and a Specified Channel*

Syntax:

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

Purpose:

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

user_id - is the valid user id;
channel number - is the channel number;
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

Example:

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"
```

3.53. version

Syntax:

version

Purpose:

This command shows the version information for the Shelf Manager software.

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

Example:

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

Pigeon Point Shelf Manager ver. 2.3
Pigeon Point is a trademark of Pigeon Point Systems.
Copyright (c) 2002-2006 Pigeon Point Systems
Build date/time: May 31 2006 16:39:37
All rights reserved #
```

Chapter 4

4. Web Interface

4.1. Pigeon Point 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.

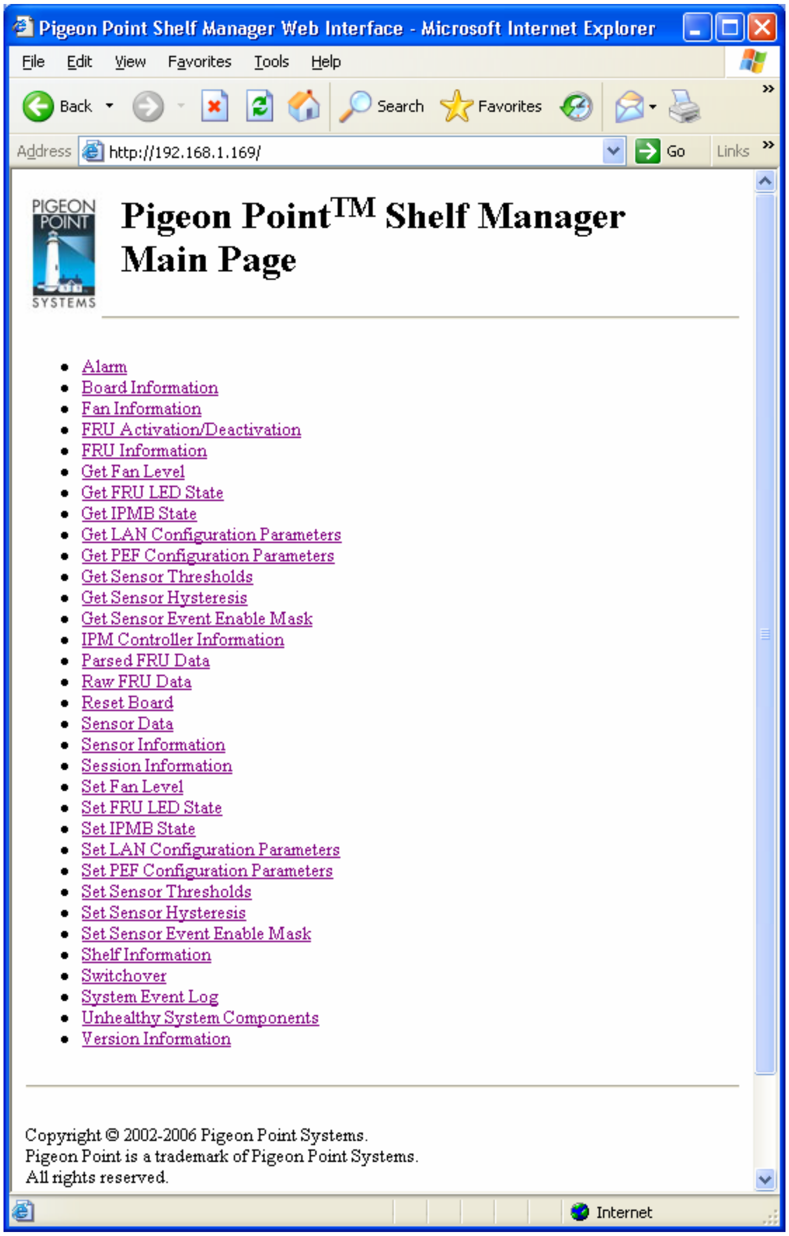
4.2. Starting the Web Interface

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, a user should 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”.


Pigeon Point Shelf Manager Web Interface - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://192.168.1.169/ Go Links
PIGEON POINT SYSTEMS
Internet

Pigeon Point^{TM} 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 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

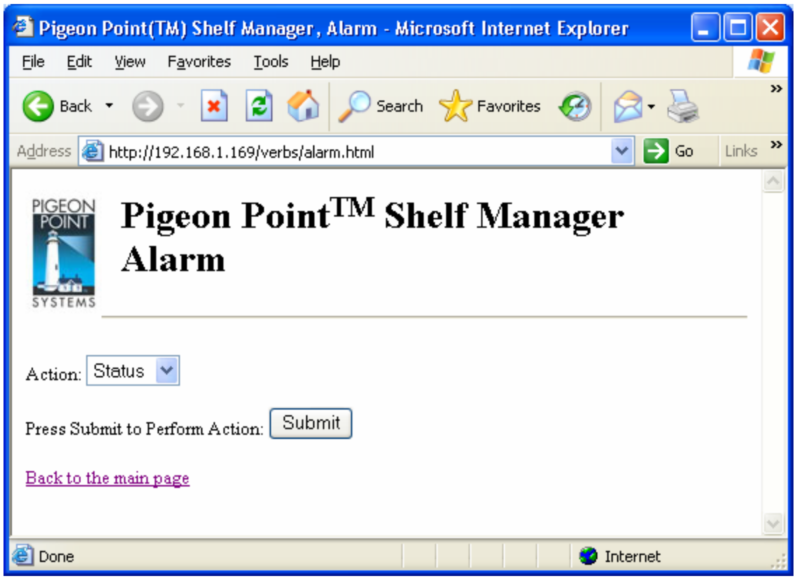
Copyright © 2002-2006 Pigeon Point Systems.
Pigeon Point is a trademark of Pigeon Point Systems.
All rights reserved.

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.

4.2.1. *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.



http://192.168.1.169/verbs/alarm.html

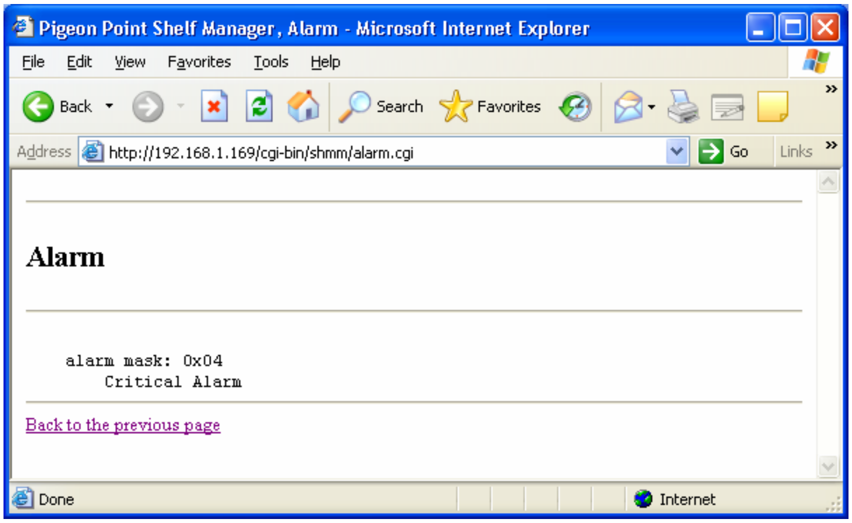
Pigeon Point^{TM} Shelf Manager
Alarm

Action: Status

Press Submit to Perform Action: Submit

Back to the main page

After the user selects one of the actions specified and click 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 correspondent parameter.


Pigeon Point Shelf Manager, Alarm - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://192.168.1.169/cgi-bin/shmm/alarm.cgi
Go Links
Done
Internet

Alarm

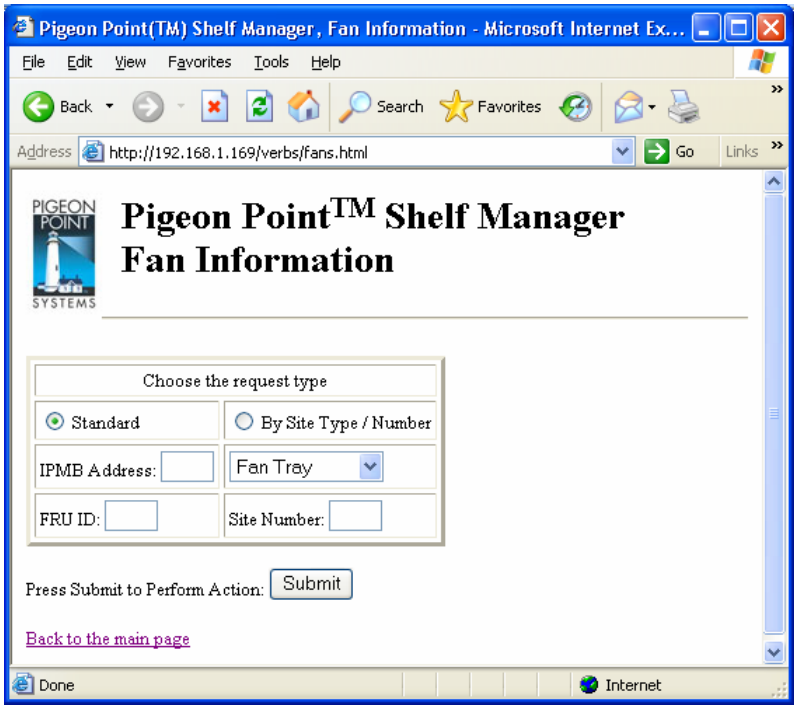
```
alarm mask: 0x04
    Critical Alarm
```

Back to the previous page

4.2.2. 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.


Pigeon Point(TM) Shelf Manager, Fan Information - Microsoft Internet Ex...
File Edit View Favorites Tools Help
Back Search Favorites
Address http://192.168.1.169/verbs/fans.html
Go Links
PIGEON POINT SYSTEMS
Done
Internet

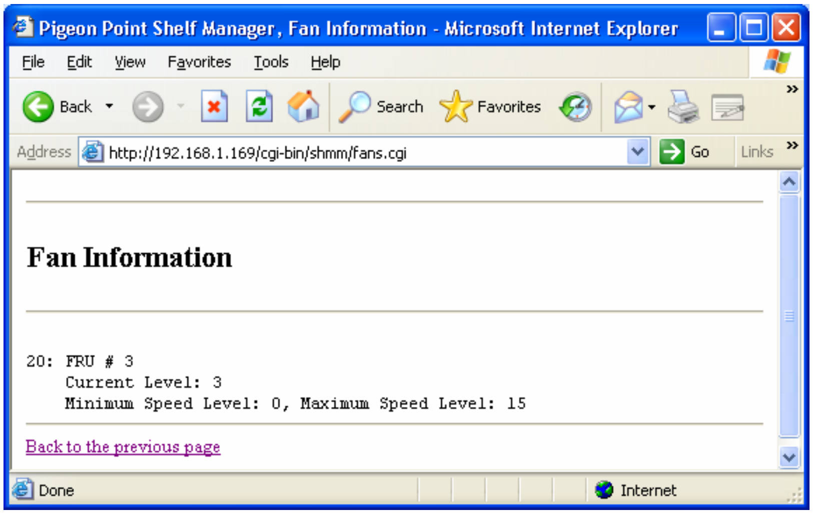
Pigeon Point^{TM} Shelf Manager
Fan Information

Choose the request type	
● Standard	○ By Site Type / Number
IPMB Address:	Fan Tray
FRU ID:	Site Number:

Press Submit to Perform Action: Submit

Back to the main page

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".


Pigeon Point Shelf Manager, Fan Information - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://192.168.1.169/cgi-bin/shmm/fans.cgi
Go
Links
Done
Internet

Fan Information

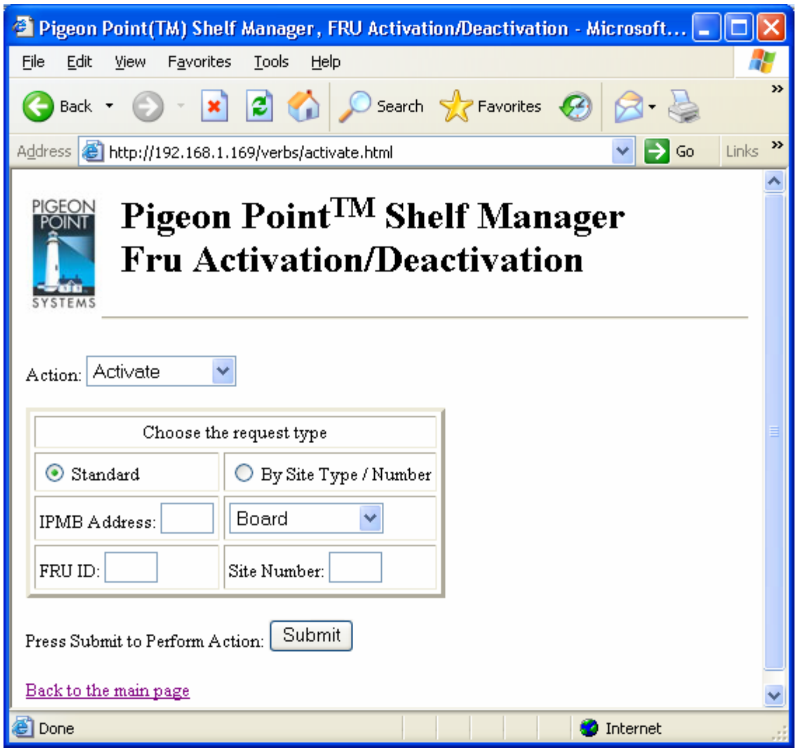
```
20: FRU # 3
    Current Level: 3
    Minimum Speed Level: 0, Maximum Speed Level: 15
```

Back to the previous page

4.2.3. 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.



Pigeon Point(TM) Shelf Manager, FRU Activation/Deactivation - Microsoft...

Address: http://192.168.1.169/verbs/activate.html

Pigeon Point™ Shelf Manager
Fru Activation/Deactivation

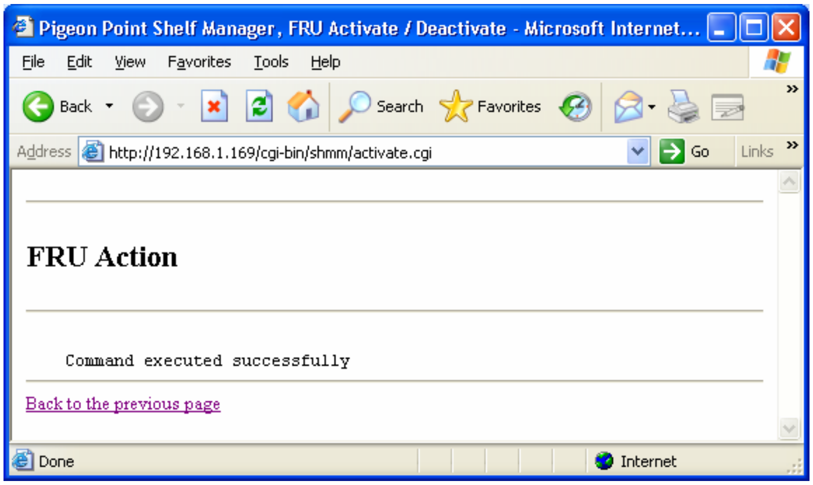
Action: Activate

Choose the request type	
● Standard	○ By Site Type / Number
IPMB Address:	Board
FRU ID:	Site Number:

Press Submit to Perform Action: Submit

Back to the main page

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.



Pigeon Point Shelf Manager, FRU Activate / Deactivate - Microsoft Internet...

Address: http://192.168.1.169/cgi-bin/shmm/activate.cgi

FRU Action

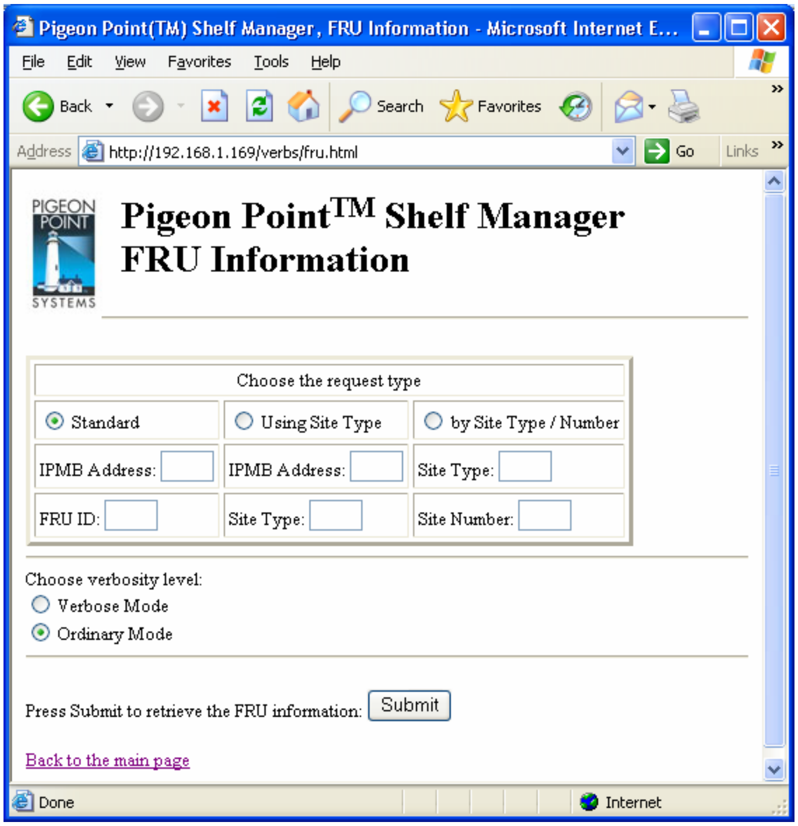
```
Command executed successfully
```

Back to the previous page

4.2.4. 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.



Pigeon Point(TM) Shelf Manager, FRU Information - Microsoft Internet E...
File Edit View Favorites Tools Help
Back Search Favorites
Address http://192.168.1.169/verbs/fru.html Go Links
PIGEON POINT SYSTEMS
Done Internet

Pigeon Point^{TM} Shelf Manager
FRU Information

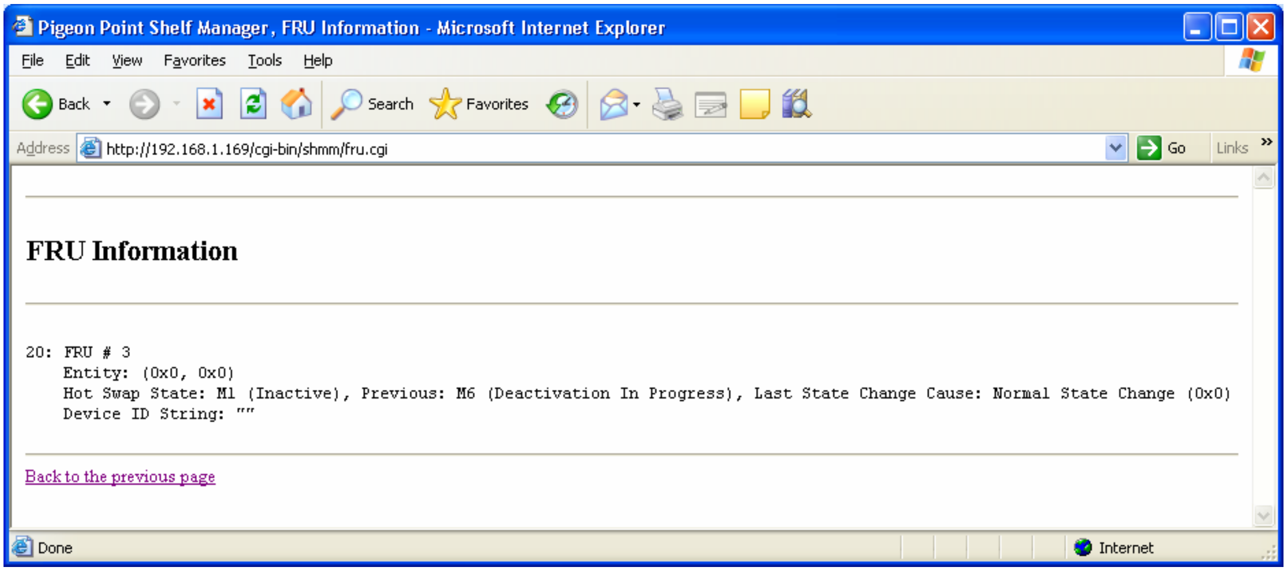
Choose the request type		
● Standard	○ Using Site Type	○ by Site Type / Number
IPMB Address:	IPMB Address:	Site Type:
FRU ID:	Site Type:	Site Number:

Choose verbosity level:
○ Verbose Mode
● Ordinary Mode

Press Submit to retrieve the FRU information: Submit

Back to the main page

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".


Pigeon Point Shelf Manager, FRU Information - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://192.168.1.169/cgi-bin/shmm/fru.cgi Go Links
Done
Internet

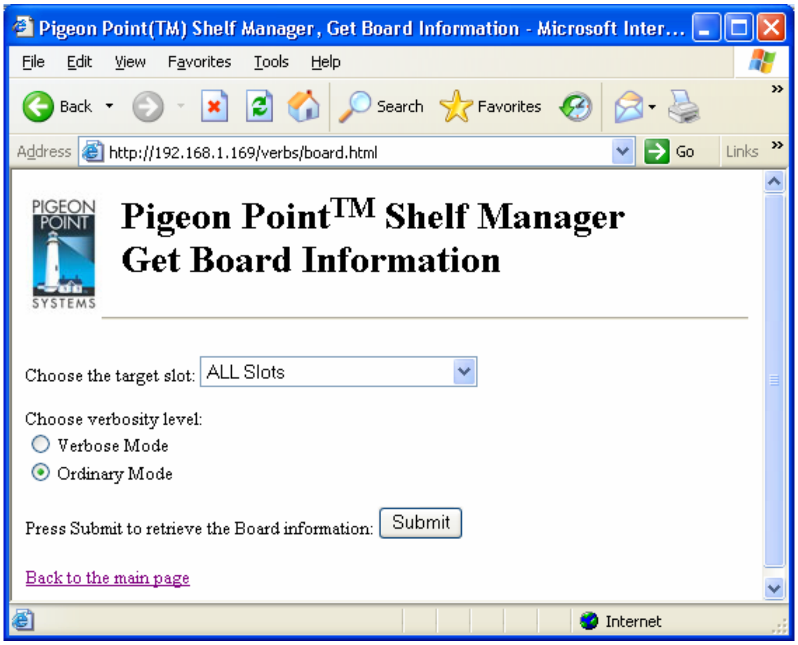
FRU Information

```
20: FRU # 3
    Entity: (0x0, 0x0)
    Hot Swap State: M1 (Inactive), Previous: M6 (Deactivation In Progress), Last State Change Cause: Normal State Change (0x0)
    Device ID String: ""
```

Back to the previous page

4.2.5. 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”.


Pigeon Point(TM) Shelf Manager, Get Board Information - Microsoft Inter...
File Edit View Favorites Tools Help
Back Search Favorites
Address http://192.168.1.169/verbs/board.html Go Links
PIGEON POINT SYSTEMS
Internet

Pigeon Point™ Shelf Manager
Get Board Information

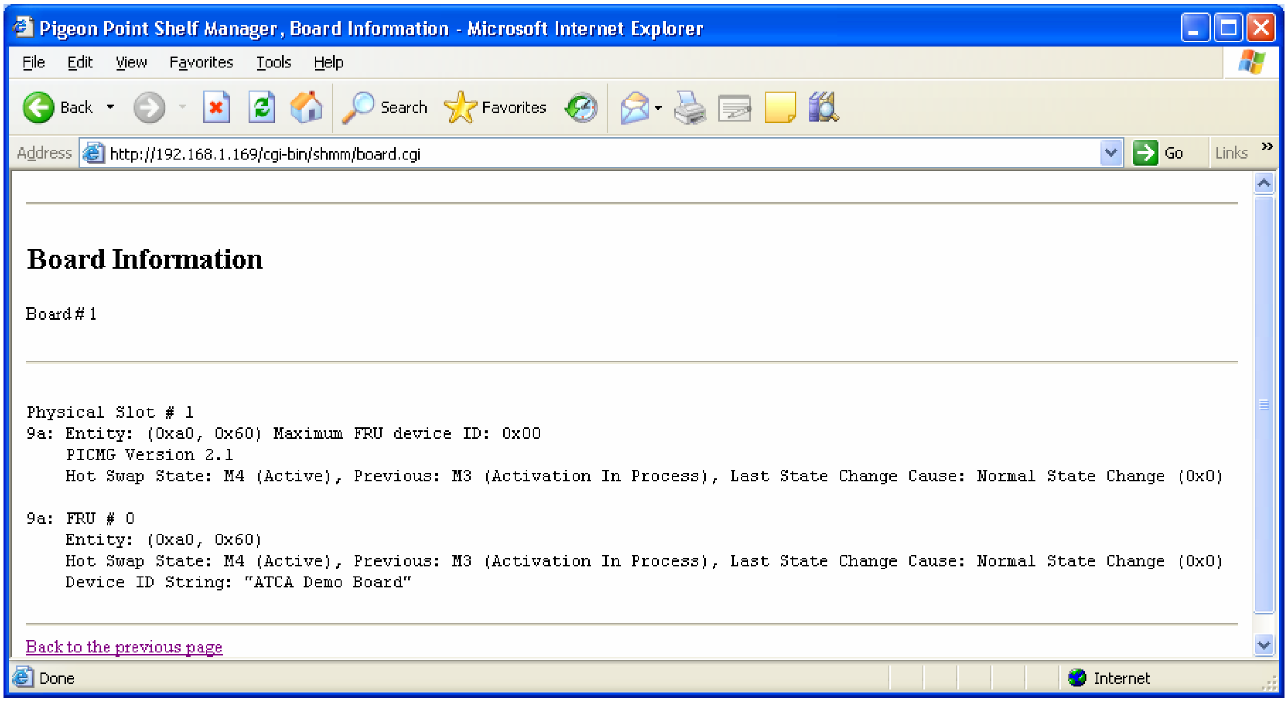
Choose the target slot: ALL Slots

Choose verbosity level:
○ Verbose Mode
● Ordinary Mode

Press Submit to retrieve the Board information: Submit

Back to the main page

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”.


Pigeon Point Shelf Manager, Board Information - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://192.168.1.169/cgi-bin/shmm/board.cgi
Go Links
Done
Internet

Board Information

Board # 1

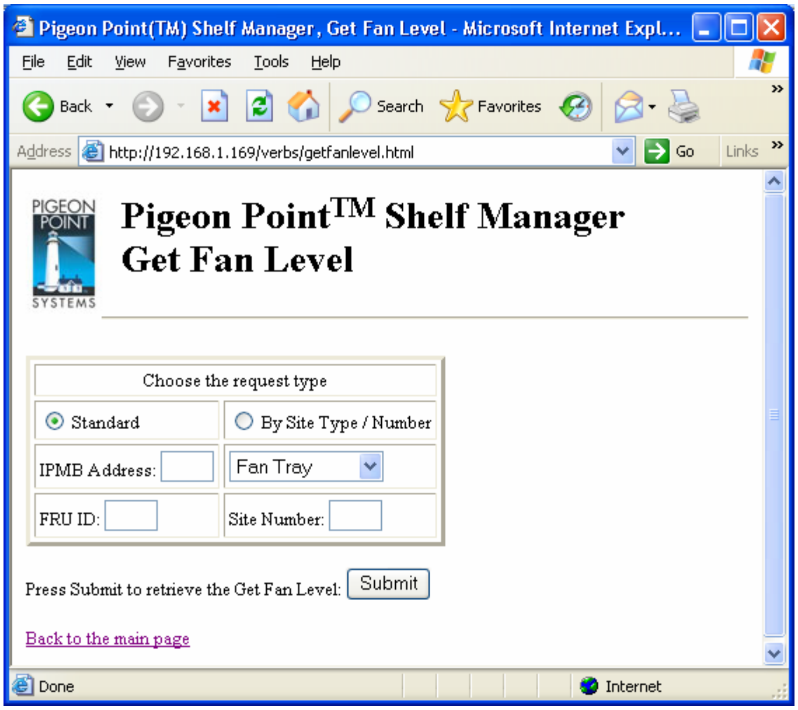
```
Physical Slot # 1
9a: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x00
    PICMG Version 2.1
    Hot Swap State: M4 (Active), Previous: M3 (Activation In Process), Last State Change Cause: Normal State Change (0x0)

9a: FRU # 0
    Entity: (0xa0, 0x60)
    Hot Swap State: M4 (Active), Previous: M3 (Activation In Process), Last State Change Cause: Normal State Change (0x0)
    Device ID String: "ATCA Demo Board"
```

Back to the previous page

4.2.6. *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.


Pigeon Point(TM) Shelf Manager, Get Fan Level - Microsoft Internet Expl...
File Edit View Favorites Tools Help
Back Search Favorites
Address http://192.168.1.169/verbs/getfanlevel.html
Go Links
PIGEON POINT SYSTEMS
Done
Internet

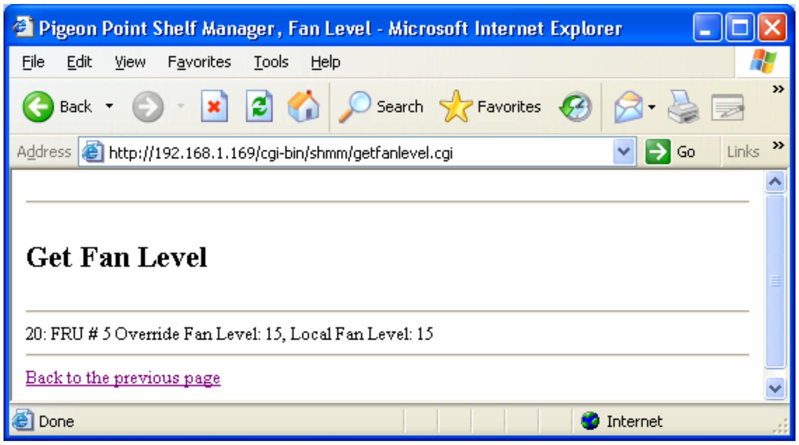
Pigeon Point™ Shelf Manager
Get Fan Level

Choose the request type	
● Standard	○ By Site Type / Number
IPMB Address:	Fan Tray
FRU ID:	Site Number:

Press Submit to retrieve the Get Fan Level: Submit

Back to the main page

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".


Pigeon Point Shelf Manager, Fan Level - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://192.168.1.169/cgi-bin/shmm/getfanlevel.cgi
Go
Links
Done
Internet

Get Fan Level

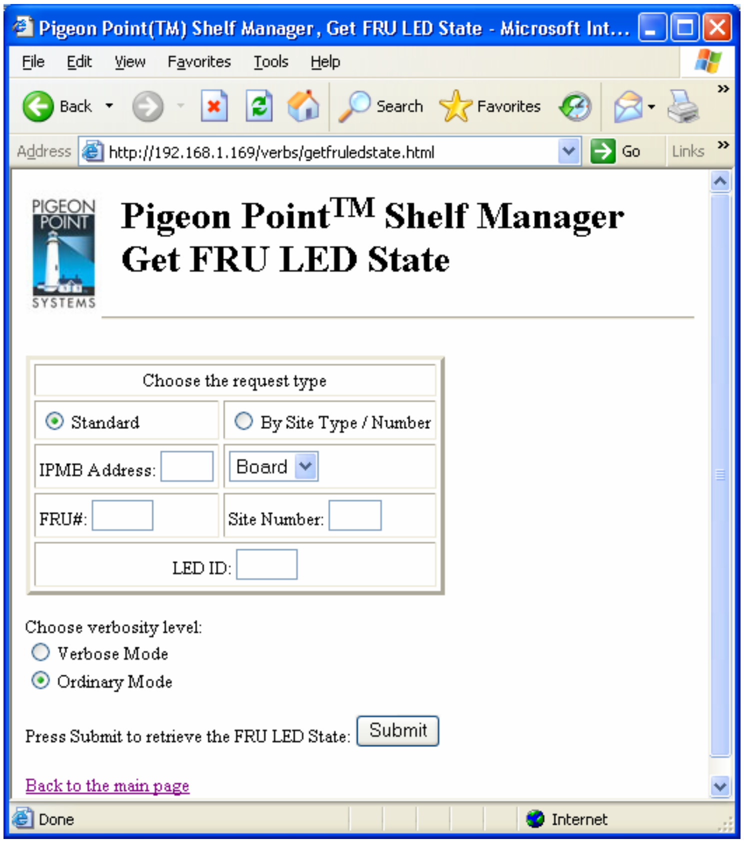
20: FRU # 5 Override Fan Level: 15, Local Fan Level: 15

Back to the previous page

4.2.7. 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.


Pigeon Point(TM) Shelf Manager, Get FRU LED State - Microsoft Int...
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
Go
Links
PIGEON POINT SYSTEMS
Submit
Done
Internet

http://192.168.1.169/verbs/getfruledstate.html

Pigeon Point^{TM} Shelf Manager
Get FRU LED State

Choose the request type

● Standard ○ By Site Type / Number

IPMB Address: Board

FRU#: Site Number:

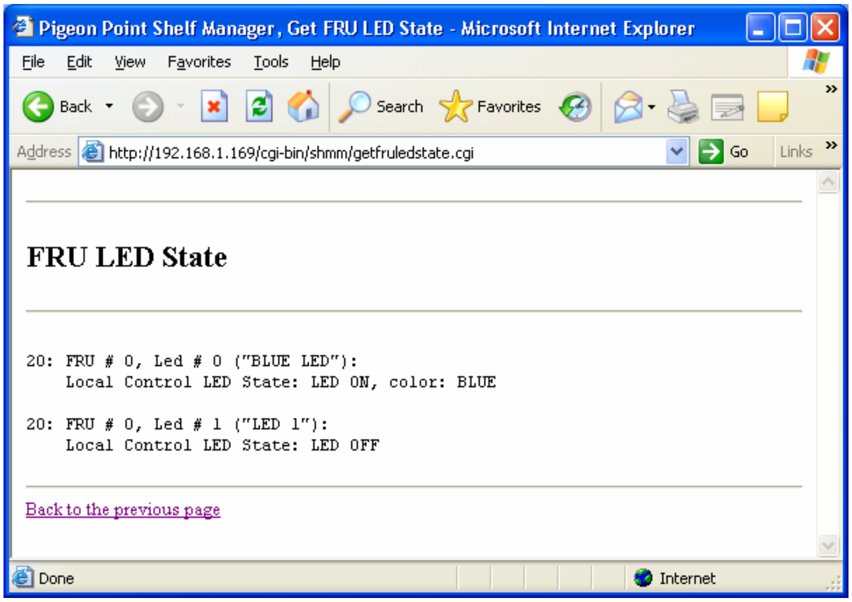
LED ID:

Choose verbosity level:
○ Verbose Mode
● Ordinary Mode

Press Submit to retrieve the FRU LED State:

Back to the main page

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
Go
Links
Done
Internet

http://192.168.1.169/cgi-bin/shmm/getfruledstate.cgi

FRU LED State

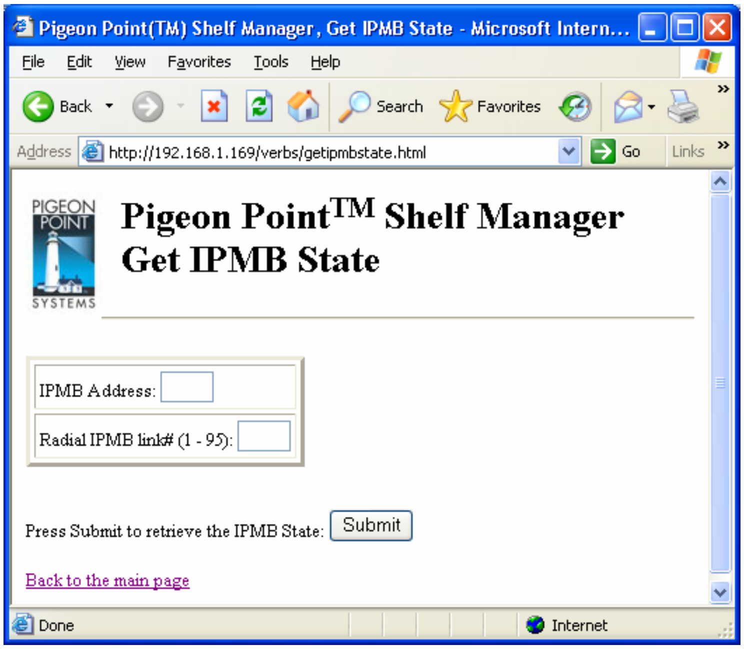
```
20: FRU # 0, Led # 0 ("BLUE LED"):
    Local Control LED State: LED ON, color: BLUE

20: FRU # 0, Led # 1 ("LED 1"):
    Local Control LED State: LED OFF
```

Back to the previous page

4.2.8. 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.



Pigeon Point(TM) Shelf Manager, Get IPMB State - Microsoft Intern...

Address: http://192.168.1.169/verbs/getipmbstate.html

Pigeon Point^{TM} Shelf Manager
Get IPMB State

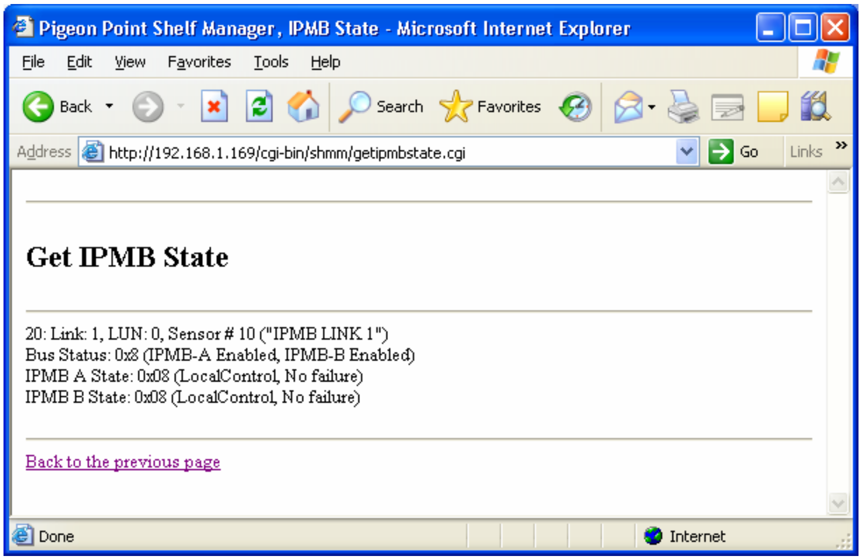
IPMB Address:

Radial IPMB link# (1 - 95):

Press Submit to retrieve the IPMB State: Submit

Back to the main page

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”.


Pigeon Point Shelf Manager, IPMB State - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://192.168.1.169/cgi-bin/shmm/getipmbstate.cgi
Go Links
Done
Internet

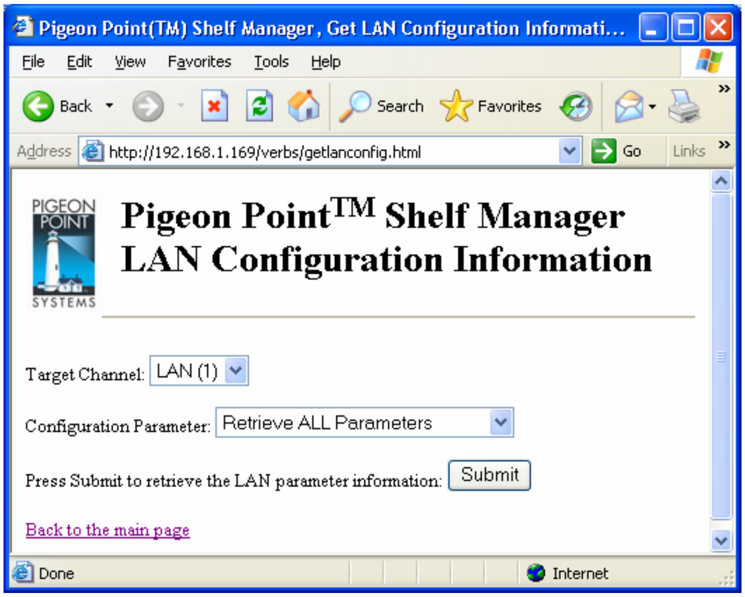
Get IPMB State

20: Link: 1, LUN: 0, Sensor # 10 ("IPMB LINK 1")
Bus Status: 0x8 (IPMB-A Enabled, IPMB-B Enabled)
IPMB A State: 0x08 (LocalControl, No failure)
IPMB B State: 0x08 (LocalControl, No failure)

Back to the previous page

4.2.9. 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.


Pigeon Point(TM) Shelf Manager, Get LAN Configuration Informati...
File Edit View Favorites Tools Help
Back Search Favorites
Address http://192.168.1.169/verbs/getlanconfig.html
Go Links
PIGEON POINT SYSTEMS
Submit
Done
Internet

Pigeon Point^{TM} Shelf Manager
LAN Configuration Information

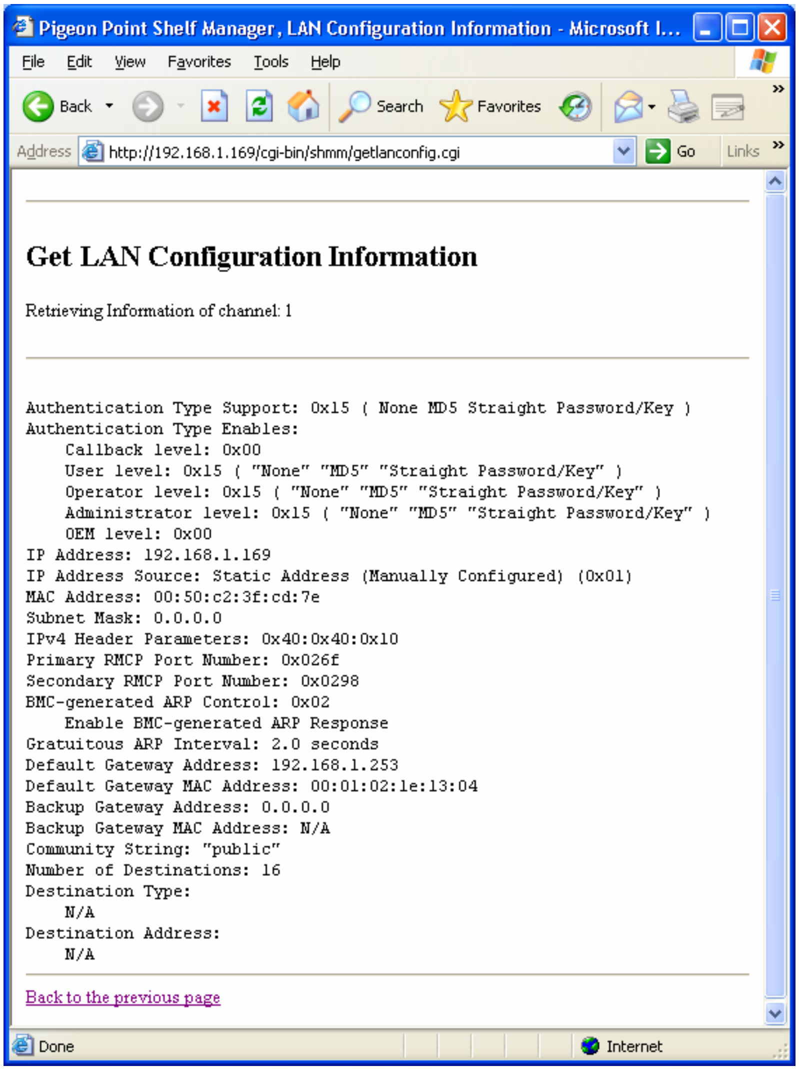
Target Channel: LAN (1)

Configuration Parameter: Retrieve ALL Parameters

Press Submit to retrieve the LAN parameter information: Submit

Back to the main page

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".


Pigeon Point Shelf Manager, LAN Configuration Information - Microsoft I...
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://192.168.1.169/cgi-bin/shmm/getlanconfig.cgi
Go
Links
Done
Internet

Get LAN Configuration Information

Retrieving Information of channel: 1

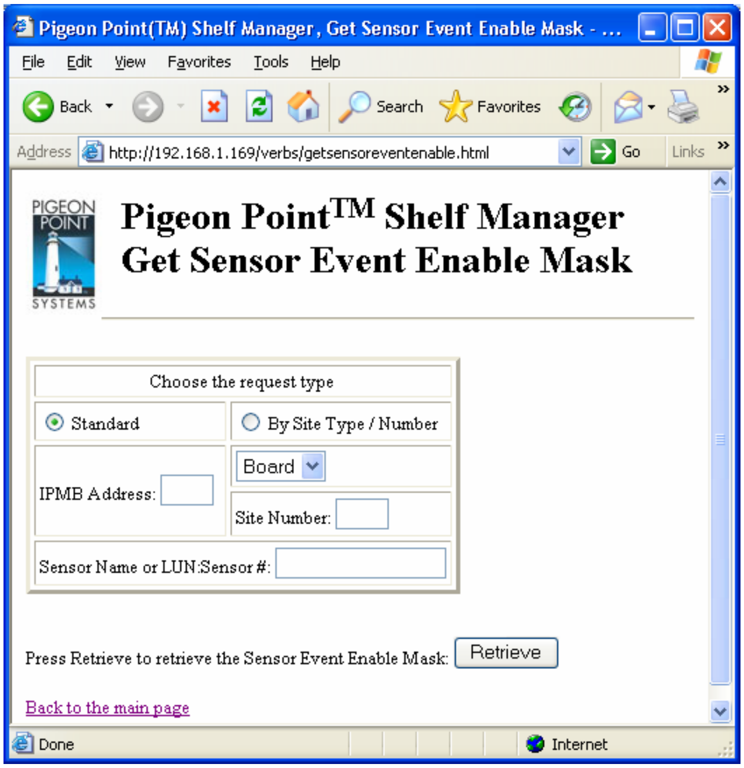
```
Authentication Type Support: 0x15 ( None MD5 Straight Password/Key )
Authentication Type Enables:
    Callback level: 0x00
    User level: 0x15 ( "None" "MD5" "Straight Password/Key" )
    Operator level: 0x15 ( "None" "MD5" "Straight Password/Key" )
    Administrator level: 0x15 ( "None" "MD5" "Straight Password/Key" )
    OEM level: 0x00
IP Address: 192.168.1.169
IP Address Source: Static Address (Manually Configured) (0x01)
MAC Address: 00:50:c2:3f:cd:7e
Subnet Mask: 0.0.0.0
IPv4 Header Parameters: 0x40:0x40:0x10
Primary RMCP Port Number: 0x026f
Secondary RMCP Port Number: 0x0298
BMC-generated ARP Control: 0x02
    Enable BMC-generated ARP Response
Gratuitous ARP Interval: 2.0 seconds
Default Gateway Address: 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
```

Back to the previous page

4.2.10. 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.


Pigeon Point(TM) Shelf Manager, Get Sensor Event Enable Mask - ...
File Edit View Favorites Tools Help
Back
Search
Favorites
Address http://192.168.1.169/verbs/getsensoreventenable.html
Go
Links
PIGEON POINT SYSTEMS
Done
Internet

Pigeon Point^{TM} Shelf Manager
Get Sensor Event Enable Mask

Choose the request type

● Standard ○ By Site Type / Number

IPMB Address:

Board

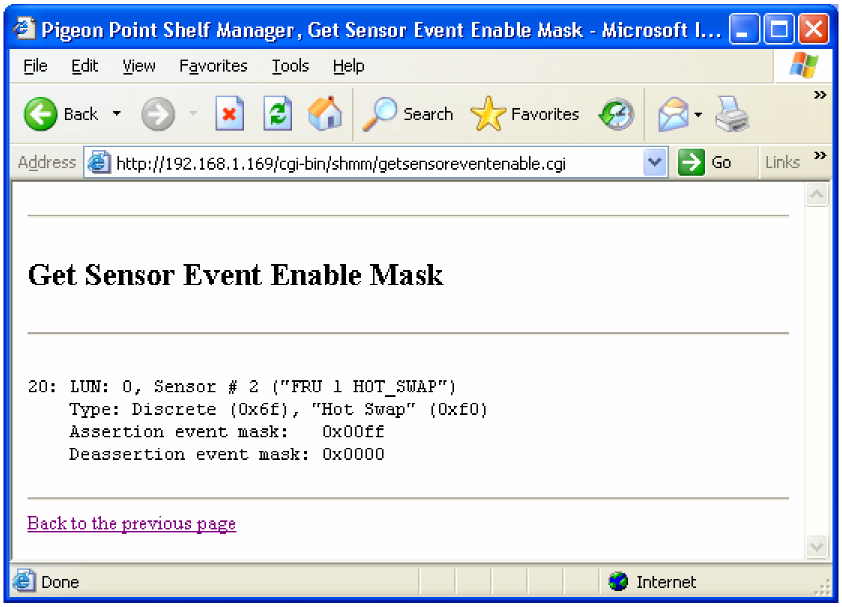
Site Number:

Sensor Name or LUN:Sensor #:

Press Retrieve to retrieve the Sensor Event Enable Mask: Retrieve

Back to the main page

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 I...
File Edit View Favorites Tools Help
Back
Search
Favorites
Address http://192.168.1.169/cgi-bin/shmm/getsensoreventenable.cgi
Go
Links
Done
Internet

Get Sensor Event Enable Mask

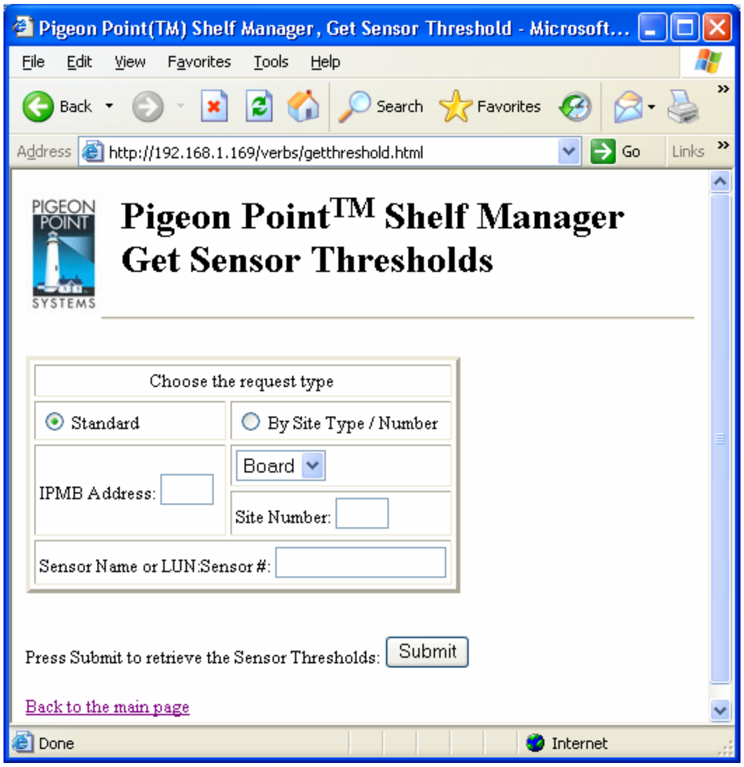
```
20: LUN: 0, Sensor # 2 ("FRU 1 HOT_SWAP")
    Type: Discrete (0x6f), "Hot Swap" (0xf0)
    Assertion event mask:   0x00ff
    Deassertion event mask: 0x0000
```

Back to the previous page

4.2.11. 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.


Pigeon Point(TM) Shelf Manager, Get Sensor Threshold - Microsoft...
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
Go
Links
PIGEON POINT SYSTEMS
Submit
Done
Internet

Address: http://192.168.1.169/verbs/getthreshold.html

Pigeon Point^{TM} Shelf Manager
Get Sensor Thresholds

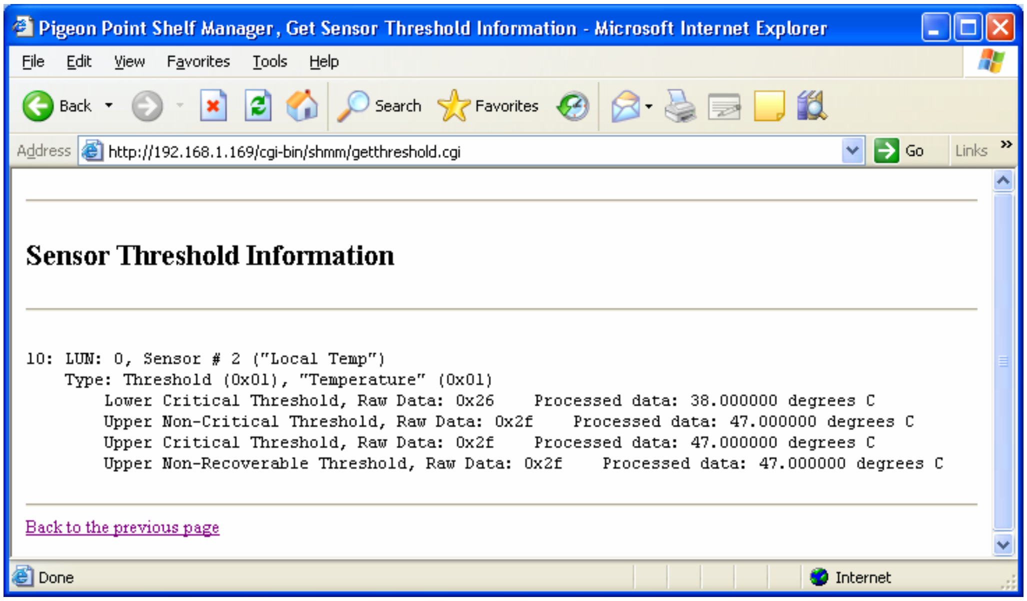
Choose the request type	
● Standard	○ By Site Type / Number
IPMB Address:	Board
	Site Number:
Sensor Name or LUN:Sensor #:	

Press Submit to retrieve the Sensor Thresholds: Submit

Back to the main page

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://192.168.1.169/cgi-bin/shmm/getthreshold.cgi
Go
Links
Done
Internet

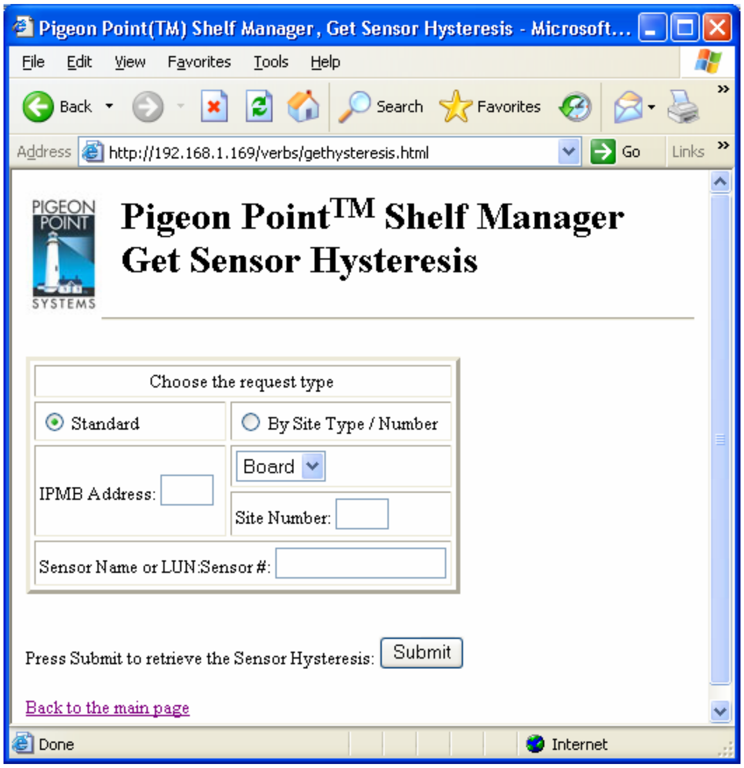
Sensor Threshold Information

```
10: LUN: 0, Sensor # 2 ("Local Temp")
    Type: Threshold (0x01), "Temperature" (0x01)
        Lower Critical Threshold, Raw Data: 0x26    Processed data: 38.000000 degrees C
        Upper Non-Critical Threshold, Raw Data: 0x2f    Processed data: 47.000000 degrees C
        Upper Critical Threshold, Raw Data: 0x2f    Processed data: 47.000000 degrees C
        Upper Non-Recoverable Threshold, Raw Data: 0x2f    Processed data: 47.000000 degrees C
```

Back to the previous page

4.2.12. Get Sensor Hysteresis

The page “Get Sensor Hysteresis” allows the user to retrieve the positive-going and negative-going hystereses 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.).


Pigeon Point(TM) Shelf Manager, Get Sensor Hysteresis - Microsoft...
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
Go
Links
Done
Internet

http://192.168.1.169/verbs/gethysteresis.html

Pigeon Point^{TM} Shelf Manager
Get Sensor Hysteresis

Choose the request type

● Standard ○ By Site Type / Number

IPMB Address:

Board

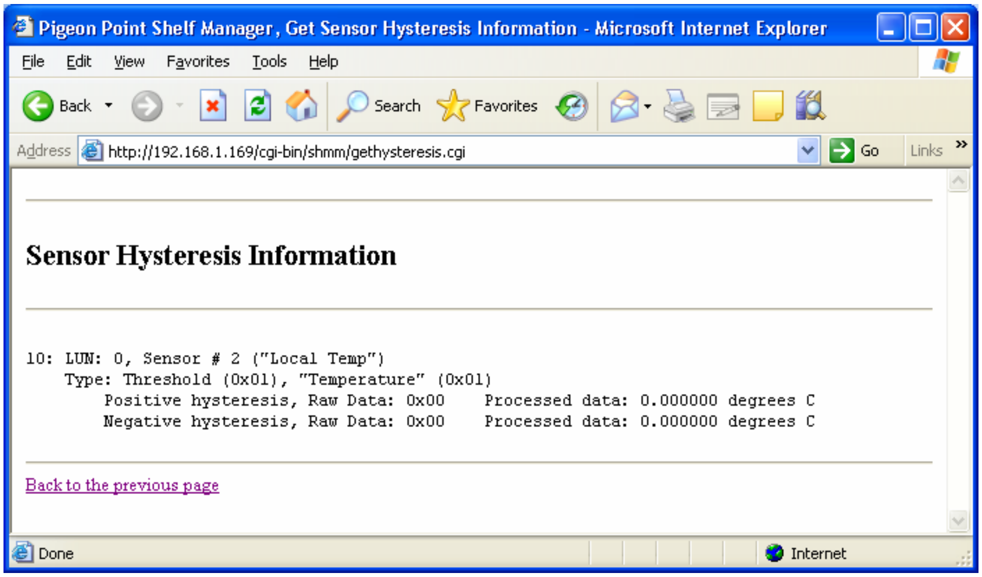
Site Number:

Sensor Name or LUN:Sensor #:

Press Submit to retrieve the Sensor Hysteresis: Submit

Back to the main page

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”.


Pigeon Point Shelf Manager, Get Sensor Hysteresis Information - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
Go
Links
Done
Internet

http://192.168.1.169/cgi-bin/shmm/gethysteresis.cgi

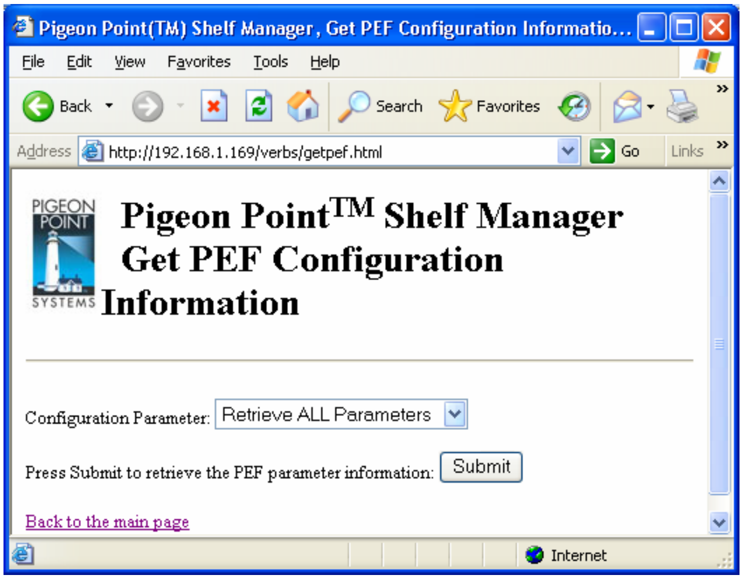
Sensor Hysteresis Information

```
10: LUN: 0, Sensor # 2 ("Local Temp")
    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
```

Back to the previous page

4.2.13. 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.



Pigeon Point(TM) Shelf Manager, Get PEF Configuration Informatio...
File Edit View Favorites Tools Help
Back
Search
Favorites
Address http://192.168.1.169/verbs/getpef.html
Go
Links
PIGEON POINT SYSTEMS
Internet

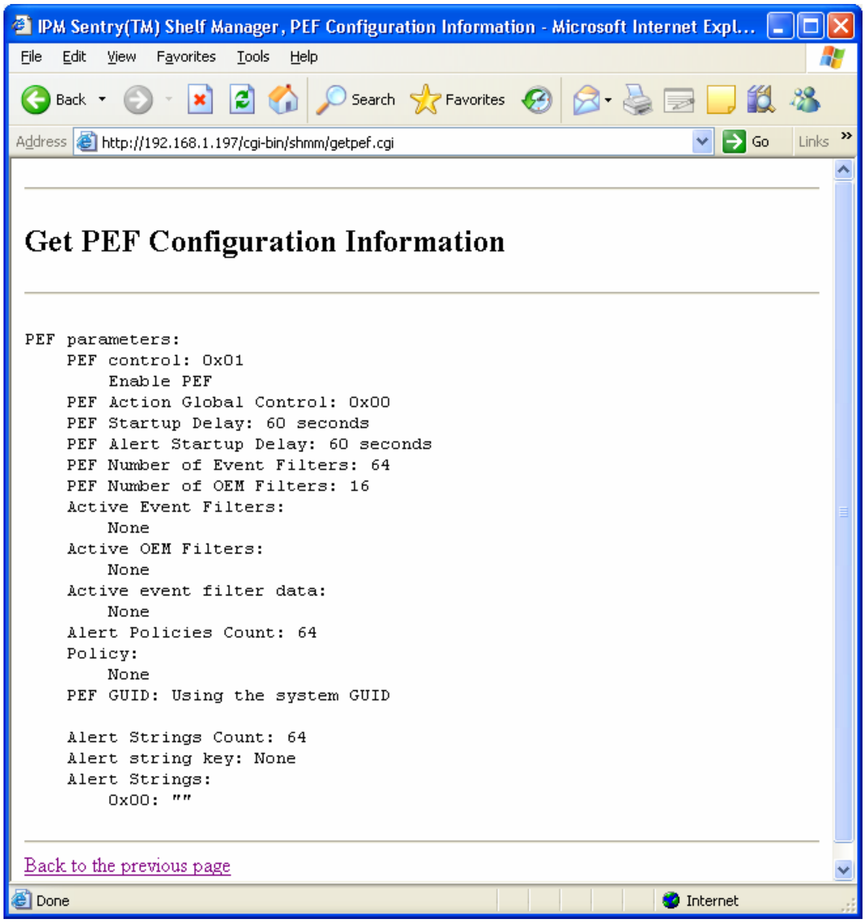
Pigeon Point^{TM} Shelf Manager Get PEF Configuration Information

Configuration Parameter: Retrieve ALL Parameters

Press Submit to retrieve the PEF parameter information: Submit

Back to the main page

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".


IPM Sentry(TM) Shelf Manager, PEF Configuration Information - Microsoft Internet Expl...
File Edit View Favorites Tools Help
Back
Search
Favorites
Address http://192.168.1.197/cgi-bin/shmm/getpef.cgi
Go
Links
Done
Internet

Get PEF Configuration Information

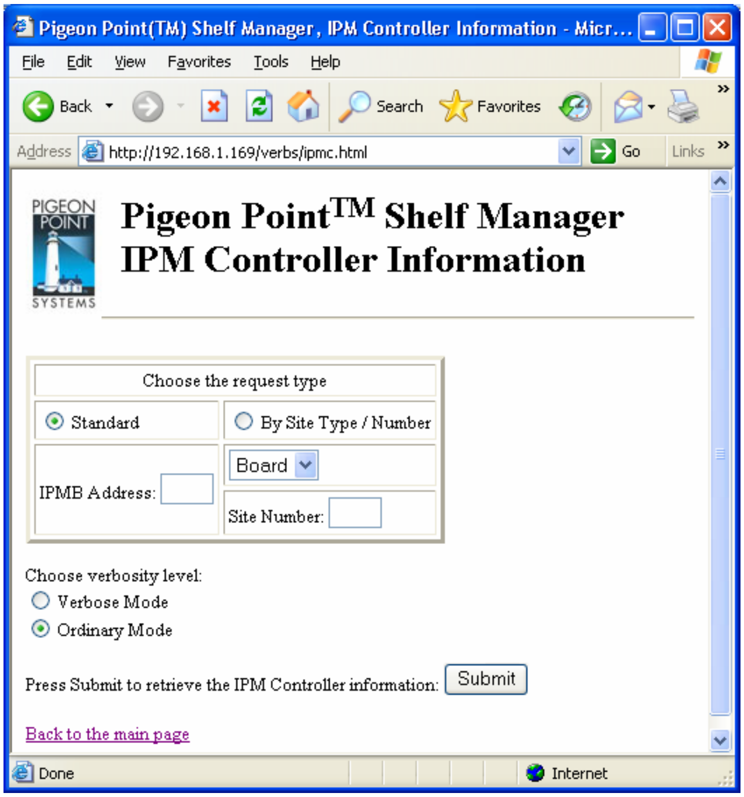
```
PEF parameters:
    PEF control: 0x01
        Enable PEF
    PEF Action Global Control: 0x00
    PEF Startup Delay: 60 seconds
    PEF Alert Startup Delay: 60 seconds
    PEF Number of Event Filters: 64
    PEF Number of OEM Filters: 16
    Active Event Filters:
        None
    Active OEM Filters:
        None
    Active event filter data:
        None
    Alert Policies Count: 64
    Policy:
        None
    PEF GUID: Using the system GUID

    Alert Strings Count: 64
    Alert string key: None
    Alert Strings:
        0x00: ""
```

Back to the previous page

4.2.14. 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.


Pigeon Point(TM) Shelf Manager, IPM Controller Information - Micr...
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
Go
Links
Done
Internet

http://192.168.1.169/verbs/ipmc.html

Pigeon Point™ Shelf Manager
IPM Controller Information

Choose the request type	
● Standard	○ By Site Type / Number
IPMB Address:	Board
	Site Number:

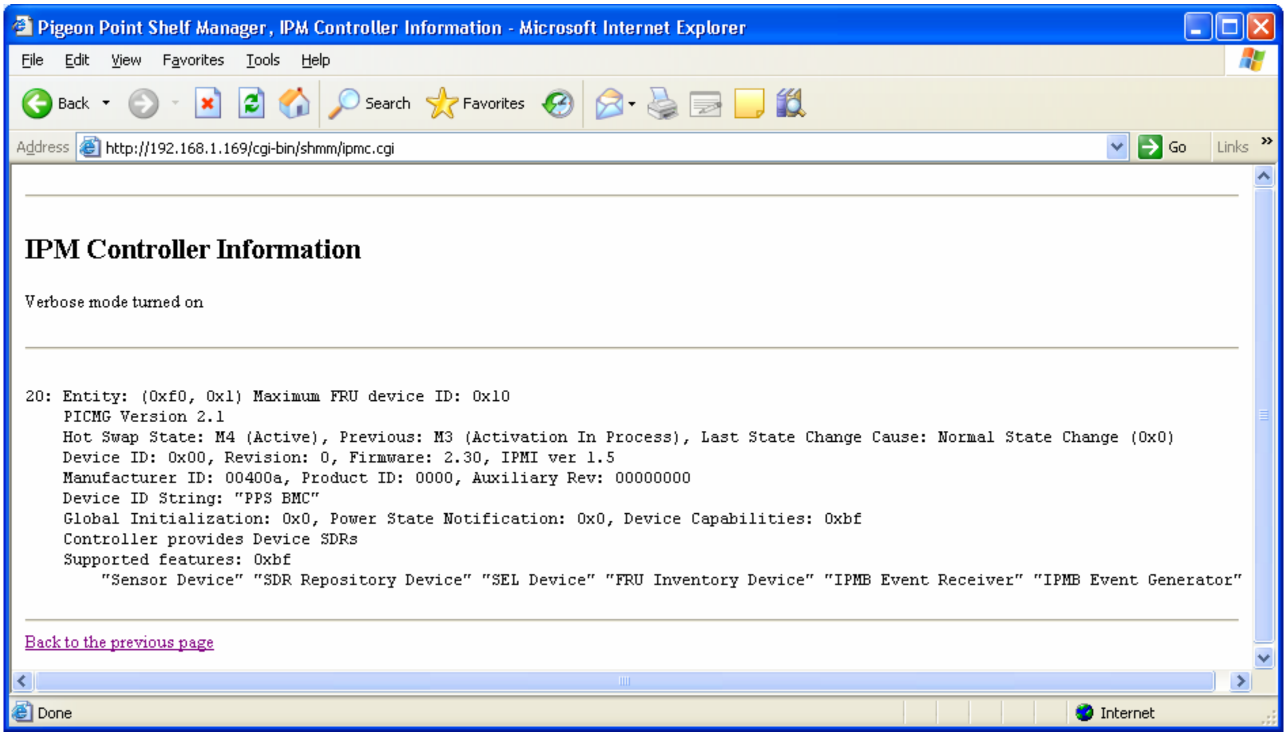
Choose verbosity level:
○ Verbose Mode
● Ordinary Mode

Press Submit to retrieve the IPM Controller information: Submit

Back to the main page

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.


Pigeon Point Shelf Manager, IPM Controller Information - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://192.168.1.169/cgi-bin/shmm/ipmc.cgi
Go
Links
Done
Internet

IPM Controller Information

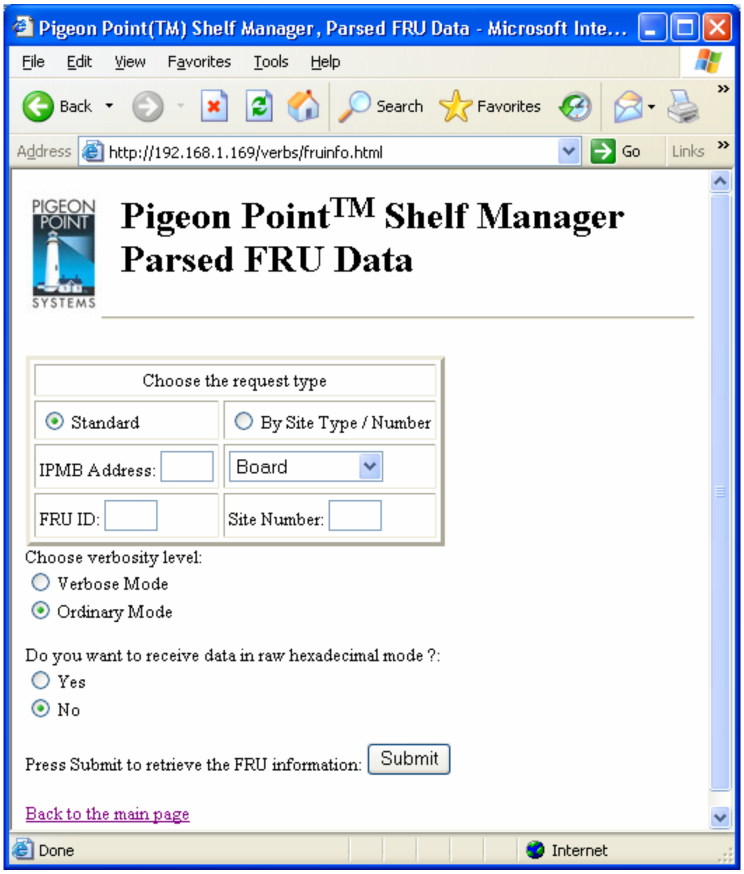
Verbose mode turned on

```
20: Entity: (0xf0, 0x1) Maximum FRU device ID: 0x10
    PICMG Version 2.1
    Hot Swap State: M4 (Active), Previous: M3 (Activation In Process), Last State Change Cause: Normal State Change (0x0)
    Device ID: 0x00, Revision: 0, Firmware: 2.30, IPMI ver 1.5
    Manufacturer ID: 00400a, Product ID: 0000, Auxiliary Rev: 00000000
    Device ID String: "PPS BMC"
    Global Initialization: 0x0, Power State Notification: 0x0, Device Capabilities: 0xbf
    Controller provides Device SDRs
    Supported features: 0xbf
        "Sensor Device" "SDR Repository Device" "SEL Device" "FRU Inventory Device" "IPMB Event Receiver" "IPMB Event Generator"
```

Back to the previous page

4.2.15. 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).


Pigeon Point(TM) Shelf Manager, Parsed FRU Data - Microsoft Inte...
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
Go
Links
PIGEON POINT SYSTEMS
Done
Internet

http://192.168.1.169/verbs/fruinfo.html

Pigeon Point^{TM} Shelf Manager
Parsed FRU Data

Choose the request type	
● Standard	○ By Site Type / Number
IPMB Address:	Board
FRU ID:	Site Number:

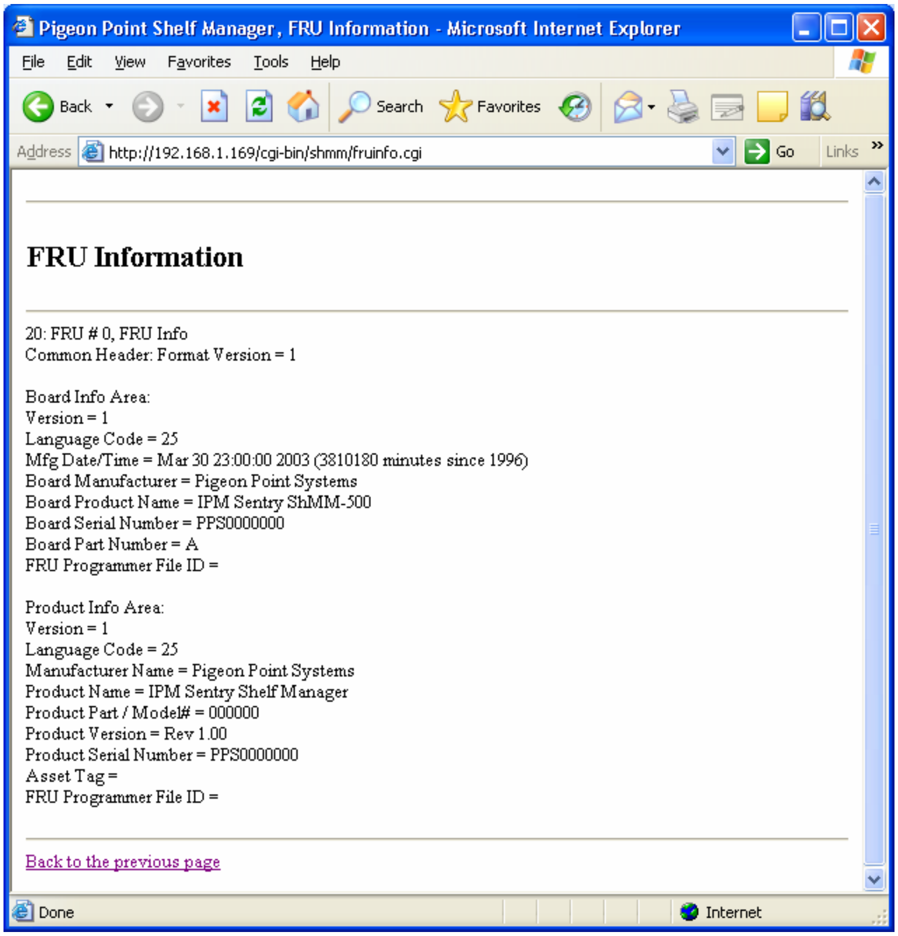
Choose verbosity level:
○ Verbose Mode
● Ordinary Mode

Do you want to receive data in raw hexadecimal mode ?:
○ Yes
● No

Press Submit to retrieve the FRU information: Submit

Back to the main page

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".


Pigeon Point Shelf Manager, FRU Information - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://192.168.1.169/cgi-bin/shmm/fruinfo.cgi
Go
Links
Done
Internet

FRU Information

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

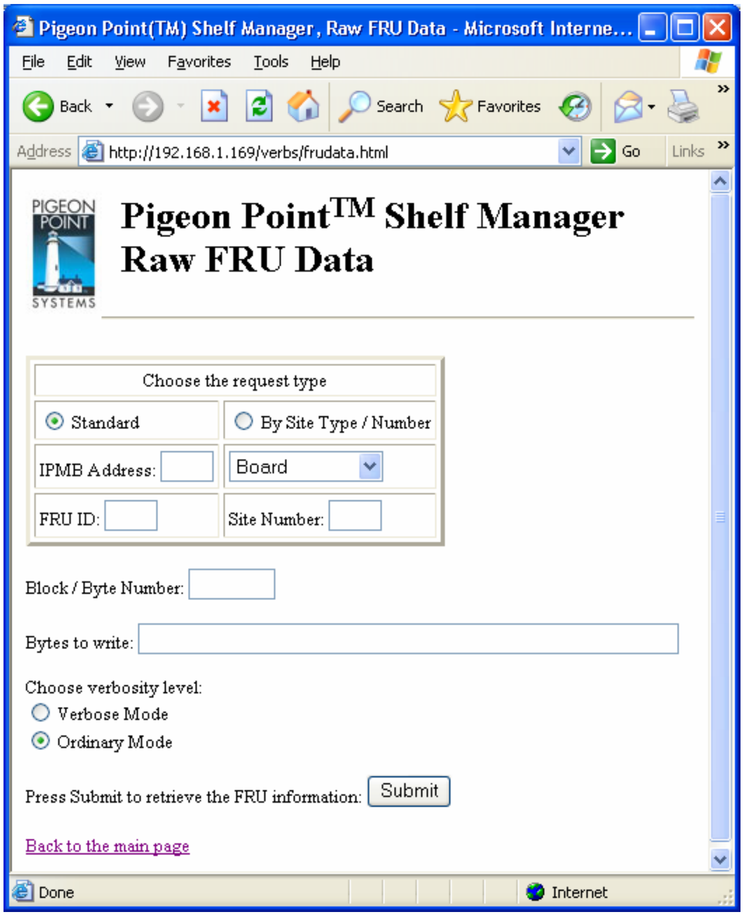
Board Info Area:
Version = 1
Language Code = 25
Mfg Date/Time = Mar 30 23:00:00 2003 (3810180 minutes since 1996)
Board Manufacturer = Pigeon Point Systems
Board Product Name = IPM Sentry ShMM-500
Board Serial Number = PPS0000000
Board Part Number = A
FRU Programmer File ID =

Product Info Area:
Version = 1
Language Code = 25
Manufacturer Name = Pigeon Point Systems
Product Name = IPM Sentry 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

4.2.16. 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.



Pigeon Point(TM) Shelf Manager, Raw FRU Data - Microsoft Interne...
File Edit View Favorites Tools Help
Back
Search
Favorites
Address http://192.168.1.169/verbs/frudata.html
Go
Links
PIGEON POINT SYSTEMS
Done
Internet

Pigeon Point^{TM} Shelf Manager
Raw FRU Data

Choose the request type

● Standard ○ By Site Type / Number

IPMB Address: ____ Board

FRU ID: ____ Site Number: ____

Block / Byte Number: ____

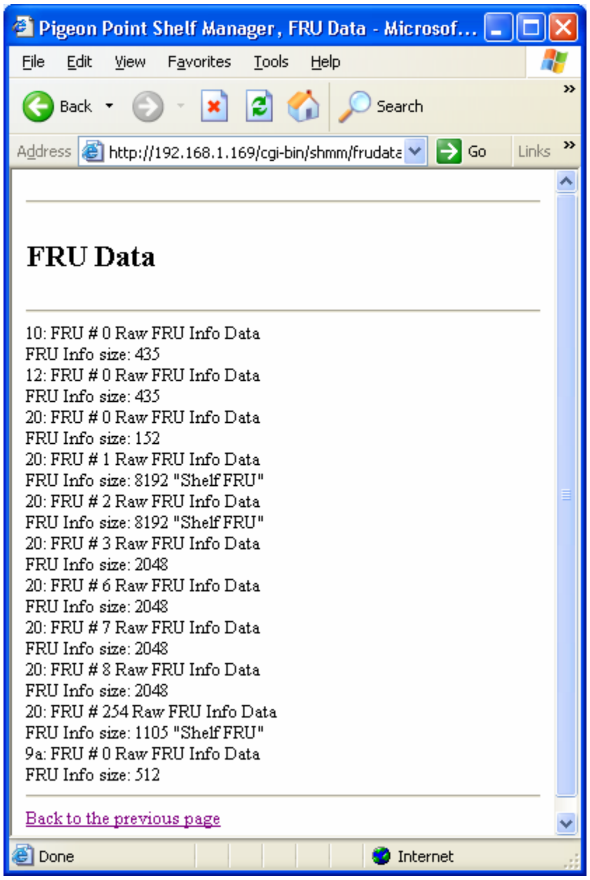
Bytes to write: ____

Choose verbosity level:
○ Verbose Mode
● Ordinary Mode

Press Submit to retrieve the FRU information: Submit

Back to the main page

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 - Microsof...
File Edit View Favorites Tools Help
Back
Search
Address http://192.168.1.169/cgi-bin/shmm/frudata
Go
Links
Done
Internet

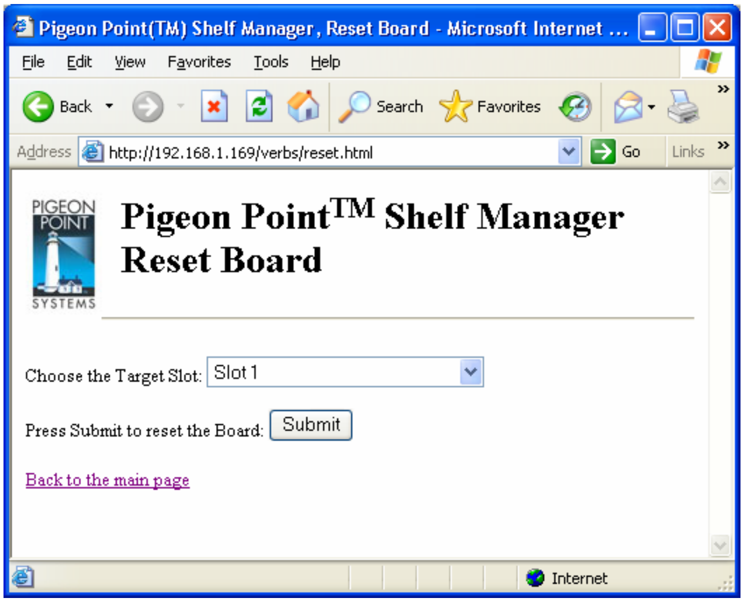
FRU Data

10: FRU # 0 Raw FRU Info Data
FRU Info size: 435
12: FRU # 0 Raw FRU Info Data
FRU Info size: 435
20: FRU # 0 Raw FRU Info Data
FRU Info size: 152
20: FRU # 1 Raw FRU Info Data
FRU Info size: 8192 "Shelf FRU"
20: FRU # 2 Raw FRU Info Data
FRU Info size: 8192 "Shelf FRU"
20: FRU # 3 Raw FRU Info Data
FRU Info size: 2048
20: FRU # 6 Raw FRU Info Data
FRU Info size: 2048
20: FRU # 7 Raw FRU Info Data
FRU Info size: 2048
20: FRU # 8 Raw FRU Info Data
FRU Info size: 2048
20: FRU # 254 Raw FRU Info Data
FRU Info size: 1105 "Shelf FRU"
9a: FRU # 0 Raw FRU Info Data
FRU Info size: 512

Back to the previous page

4.2.17. 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.


Pigeon Point(TM) Shelf Manager, Reset Board - Microsoft Internet ...
File Edit View Favorites Tools Help
Back
Search
Favorites
Address http://192.168.1.169/verbs/reset.html
Go
Links
PIGEON POINT SYSTEMS
Internet

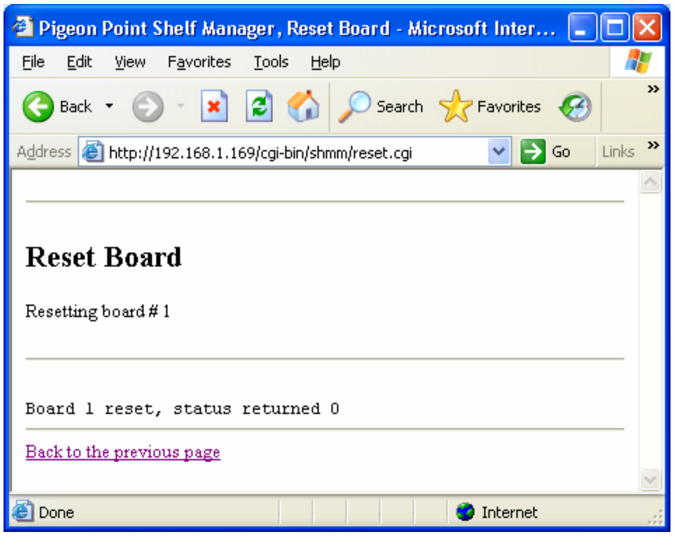
Pigeon Point^{TM} Shelf Manager Reset Board

Choose the Target Slot: Slot 1

Press Submit to reset the Board: Submit

Back to the main page

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".



Pigeon Point Shelf Manager, Reset Board - Microsoft Inter...
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
Go
Links
Done
Internet

http://192.168.1.169/cgi-bin/shmm/reset.cgi

Reset Board

Resetting board # 1

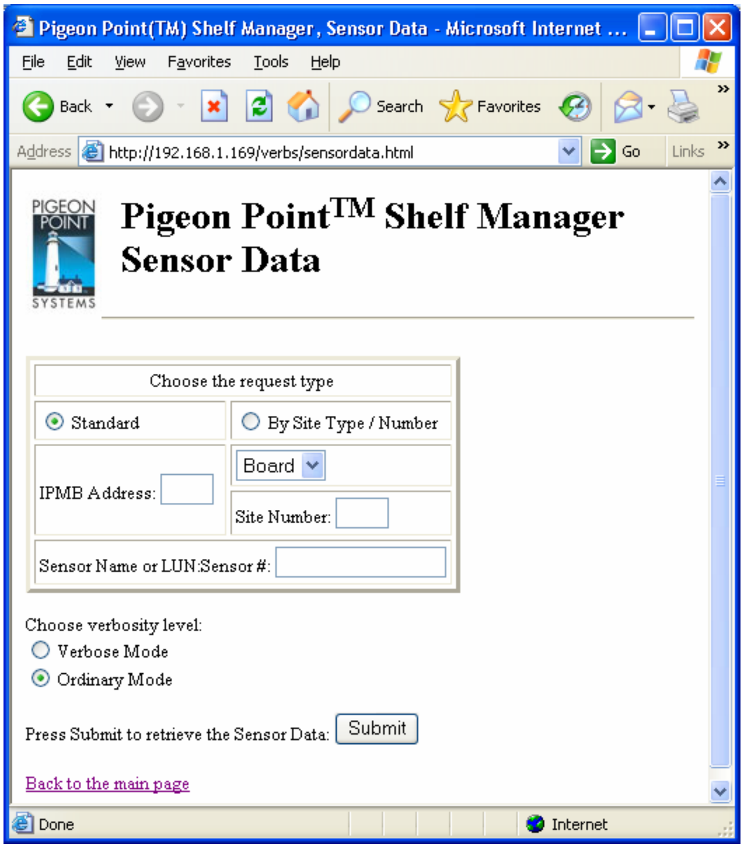
```
Board 1 reset, status returned 0
```

Back to the previous page

4.2.18. 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.


Pigeon Point(TM) Shelf Manager, Sensor Data - Microsoft Internet ...
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://192.168.1.169/verbs/sensordata.html
Go
Links
PIGEON POINT SYSTEMS
Done
Internet

Pigeon Point^{TM} Shelf Manager
Sensor Data

Choose the request type

● Standard ○ By Site Type / Number

IPMB Address:

Board

Site Number:

Sensor Name or LUN:Sensor #:

Choose verbosity level:

○ Verbose Mode

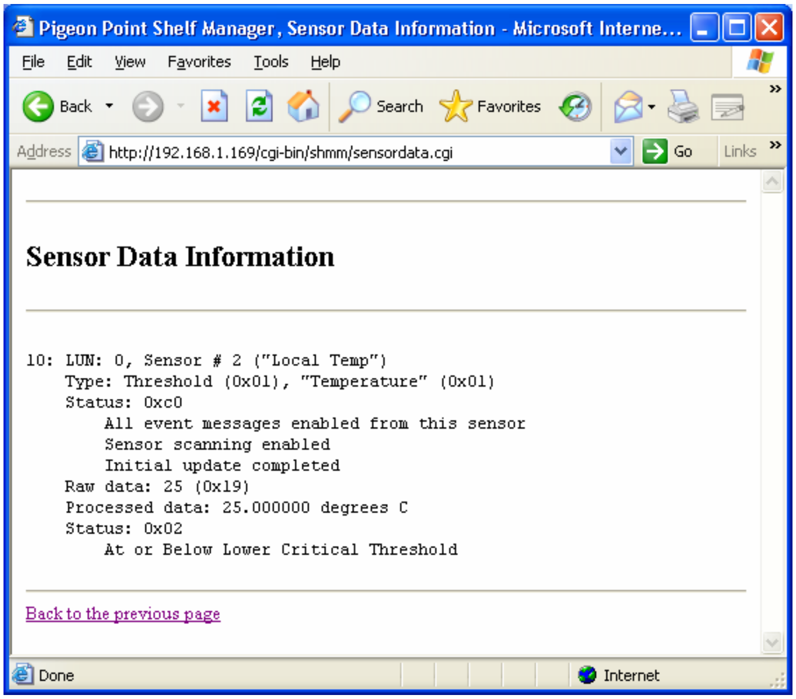
● Ordinary Mode

Press Submit to retrieve the Sensor Data: Submit

Back to the main page

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://192.168.1.169/cgi-bin/shmm/sensordata.cgi Go Links
Done Internet

Sensor Data Information

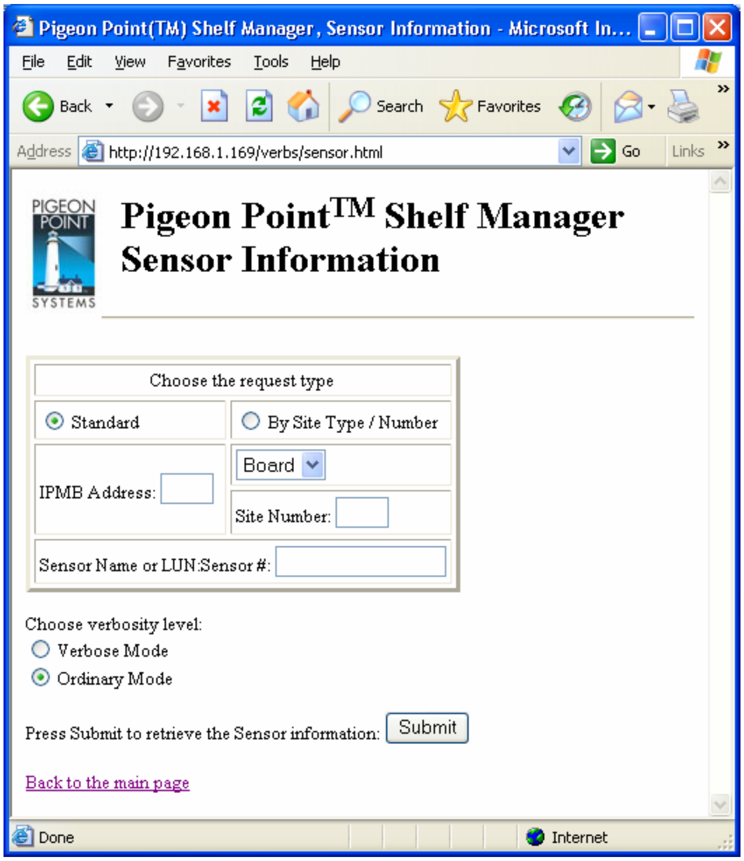
```
10: LUN: 0, Sensor # 2 ("Local Temp")
    Type: Threshold (0x01), "Temperature" (0x01)
    Status: 0xc0
        All event messages enabled from this sensor
        Sensor scanning enabled
        Initial update completed
    Raw data: 25 (0x19)
    Processed data: 25.000000 degrees C
    Status: 0x02
        At or Below Lower Critical Threshold
```

Back to the previous page

4.2.19. 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.



http://192.168.1.169/verbs/sensor.html

Pigeon Point^{TM} Shelf Manager
Sensor Information

Choose the request type

● Standard ○ By Site Type / Number

IPMB Address: Board

Site Number:

Sensor Name or LUN:Sensor #:

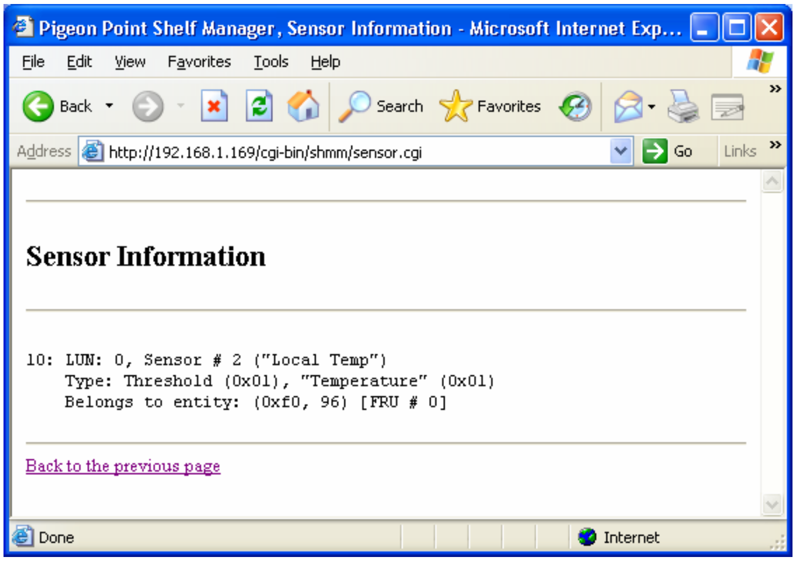
Choose verbosity level:
○ Verbose Mode
● Ordinary Mode

Press Submit to retrieve the Sensor information: Submit

Back to the main page

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 Exp...
File Edit View Favorites Tools Help
Back Search Favorites
Address http://192.168.1.169/cgi-bin/shmm/sensor.cgi
Go Links
Done
Internet

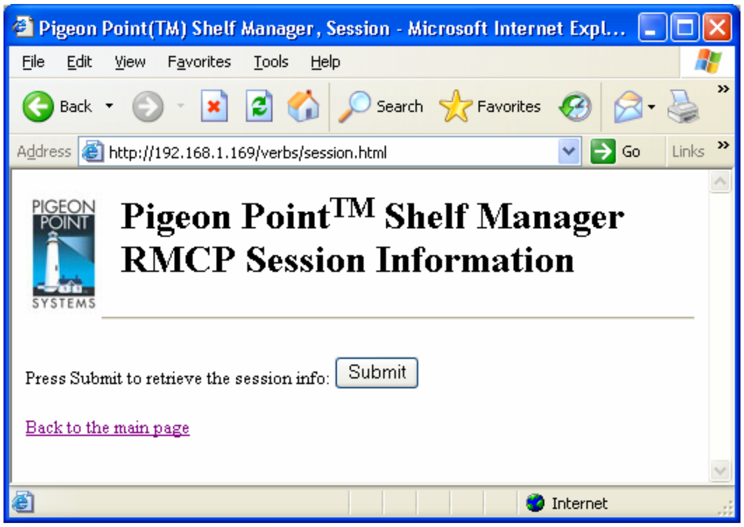
Sensor Information

```
10: LUN: 0, Sensor # 2 ("Local Temp")
    Type: Threshold (0x01), "Temperature" (0x01)
    Belongs to entity: (0xf0, 96) [FRU # 0]
```

Back to the previous page

4.2.20. Session Information

The page "Session Information" allows the user to obtaion information about active RMCP sessions.

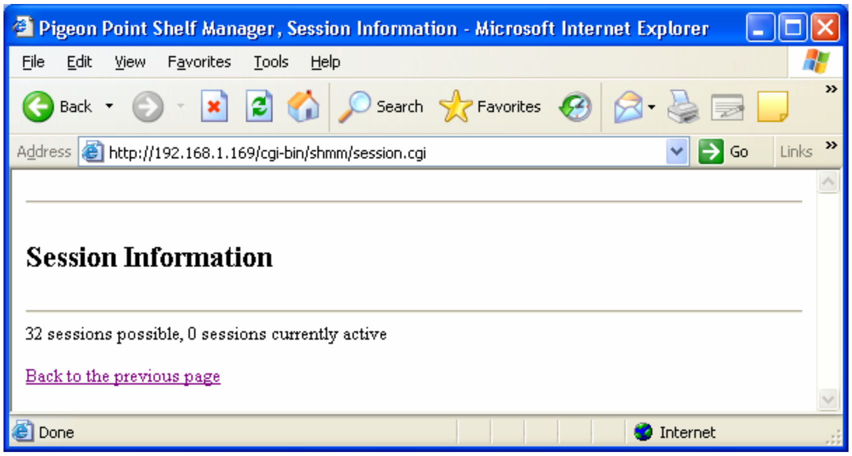

Pigeon Point(TM) Shelf Manager, Session - Microsoft Internet Expl...
File Edit View Favorites Tools Help
Back Search Favorites
Address http://192.168.1.169/verbs/session.html
Go Links
PIGEON POINT SYSTEMS
Internet

Pigeon Point^{TM} Shelf Manager
RMCP Session Information

Press Submit to retrieve the session info: Submit

Back to the main page

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".


Pigeon Point Shelf Manager, Session Information - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
Go
Links
Done
Internet

http://192.168.1.169/cgi-bin/shmm/session.cgi

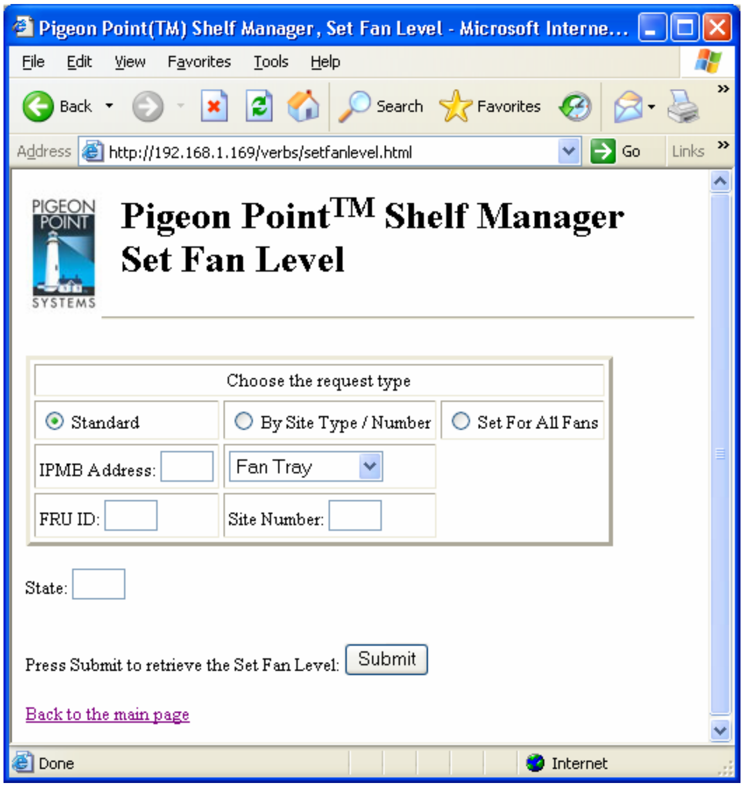
Session Information

32 sessions possible, 0 sessions currently active

Back to the previous page

4.2.21. *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.


Pigeon Point(TM) Shelf Manager, Set Fan Level - Microsoft Interne...
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
Go
Links
PIGEON POINT SYSTEMS
Done
Internet

http://192.168.1.169/verbs/setfanlevel.html

Pigeon Point^{TM} Shelf Manager
Set Fan Level

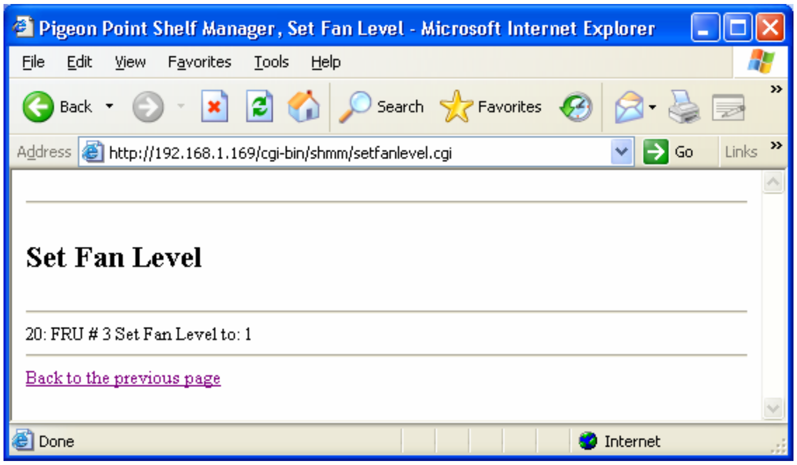
Choose the request type		
● Standard	○ By Site Type / Number	○ Set For All Fans
IPMB Address:	Fan Tray	
FRU ID:	Site Number:	

State:

Press Submit to retrieve the Set Fan Level: Submit

Back to the main page

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".


Pigeon Point Shelf Manager, Set Fan Level - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://192.168.1.169/cgi-bin/shmm/setfanlevel.cgi
Go Links
Done
Internet

Set Fan Level

20: FRU # 3 Set Fan Level to: 1

Back to the previous page

4.2.22. 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:

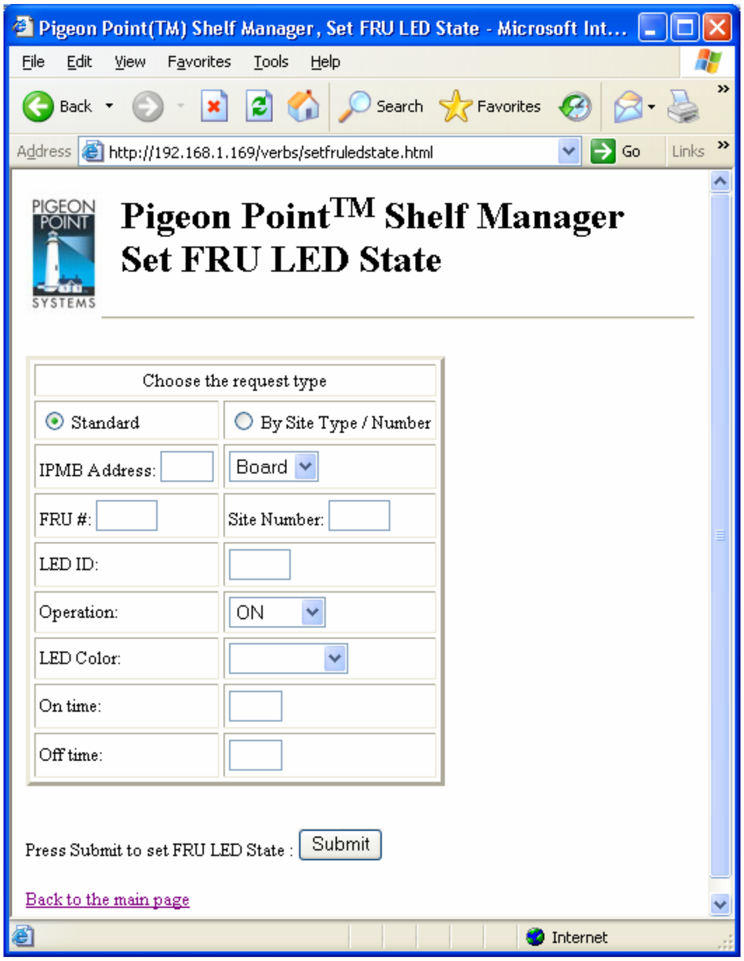
- ON – turn on the LED
- OFF – turn off the LED
- LOCAL – revert to local control of the LED
- BLINK – cause the LED to blink, repeatedly turning it on for a period of time specified in "On Time" field (in milliseconds) and then turning it off for a period of time specified in "Off Time" field (in milliseconds)
- 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


Pigeon Point(TM) Shelf Manager, Set FRU LED State - Microsoft Int...
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://192.168.1.169/verbs/setfruledstate.html
Go
Links
PIGEON POINT SYSTEMS
Internet

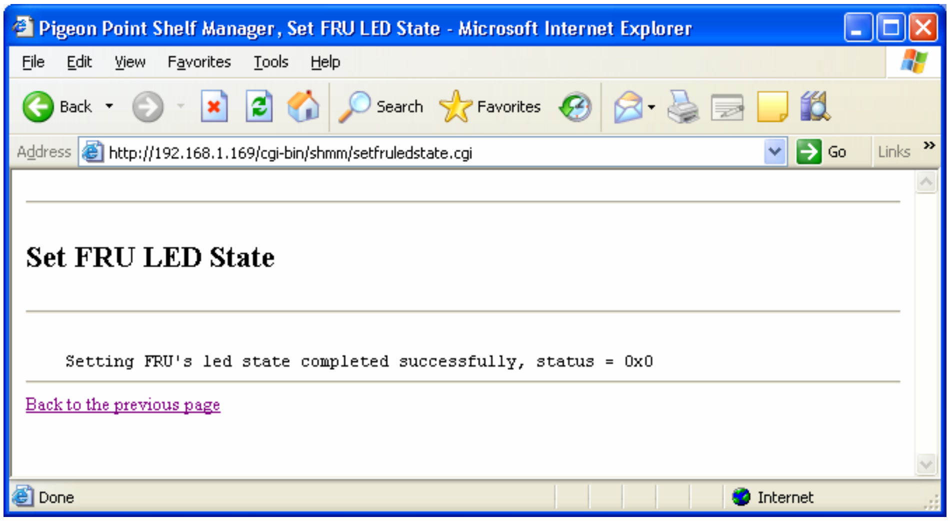
Pigeon Point™ Shelf Manager
Set FRU LED State

Choose the request type	
◉ Standard	○ By Site Type / Number
IPMB Address:	Board
FRU #:	Site Number:
LED ID:	
Operation:	ON
LED Color:	
On time:	
Off time:	

Press Submit to set FRU LED State : Submit

Back to the main page

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".


Pigeon Point Shelf Manager, Set FRU LED State - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://192.168.1.169/cgi-bin/shmm/setfruledstate.cgi
Go
Links
Done
Internet

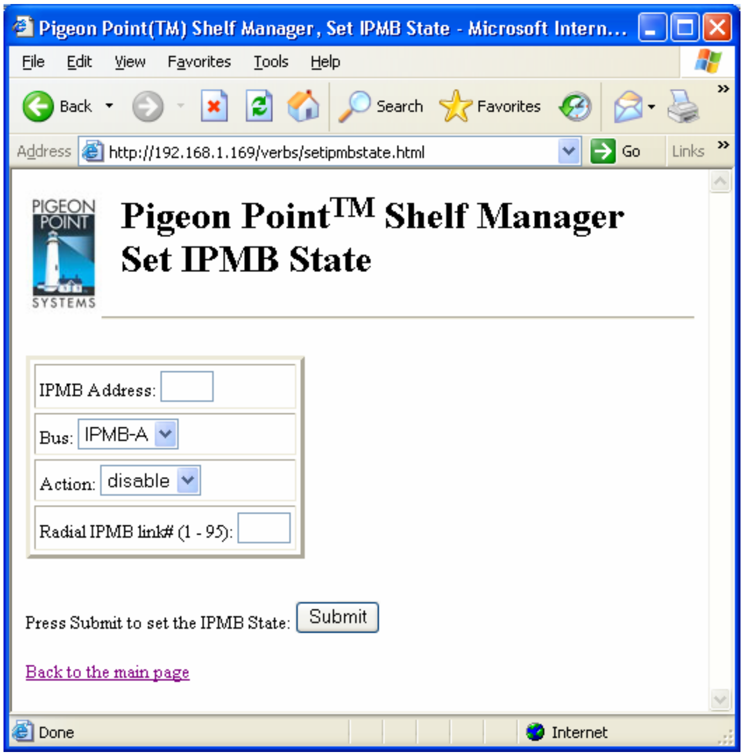
Set FRU LED State

```
    Setting FRU's led state completed successfully, status = 0x0
```

Back to the previous page

4.2.23. 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.


Pigeon Point(TM) Shelf Manager, Set IPMB State - Microsoft Intern...
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://192.168.1.169/verbs/setipmbstate.html
Go
Links
PIGEON POINT SYSTEMS
Done
Internet

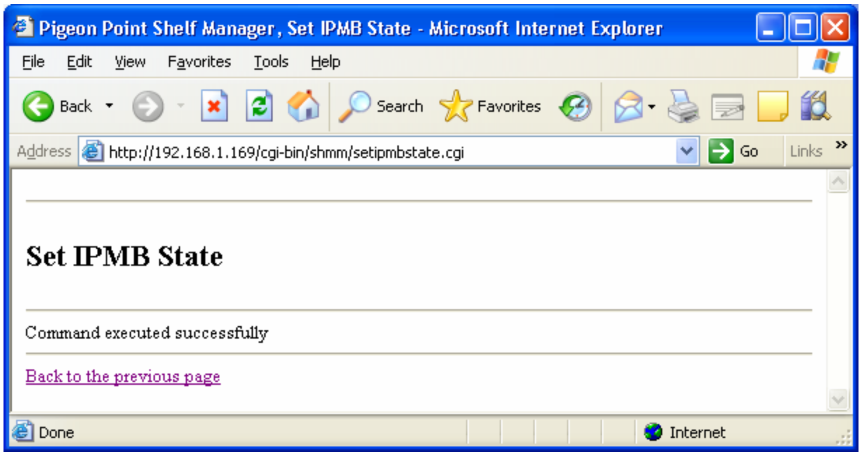
Pigeon Point^{TM} Shelf Manager
Set IPMB State

IPMB Address:
Bus: IPMB-A
Action: disable
Radial IPMB link# (1 - 95):

Press Submit to set the IPMB State: Submit

Back to the main page

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".


Pigeon Point Shelf Manager, Set IPMB State - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://192.168.1.169/cgi-bin/shmm/setipmbstate.cgi
Go
Links
Done
Internet

Set IPMB State

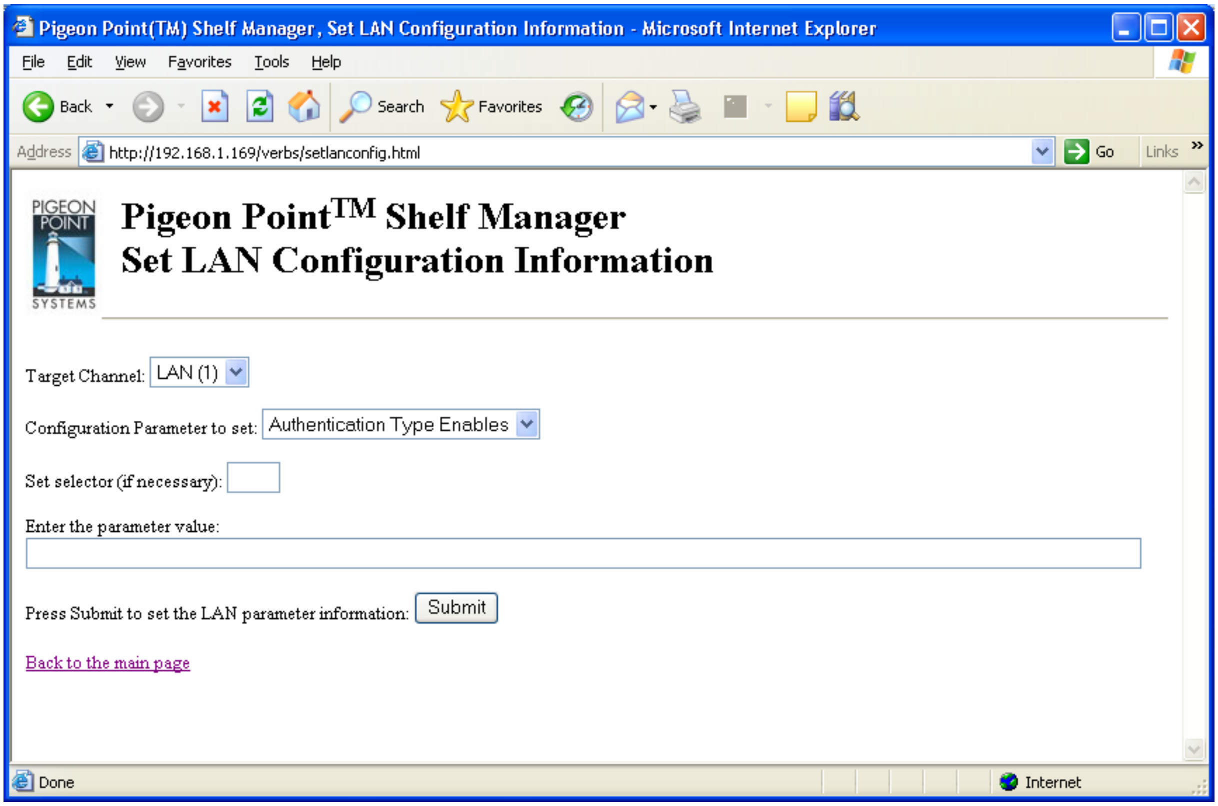
Command executed successfully

Back to the previous page

4.2.24. 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. The formats are described in section 3.37 of this document, which describes the CLI command "setlanconfig".



Pigeon Point(TM) Shelf Manager, Set LAN Configuration Information - Microsoft Internet Explorer

Address: http://192.168.1.169/verbs/setlanconfig.html

Pigeon Point^{TM} Shelf Manager
Set LAN Configuration Information

Target Channel: LAN (1)

Configuration Parameter to set: Authentication Type Enables

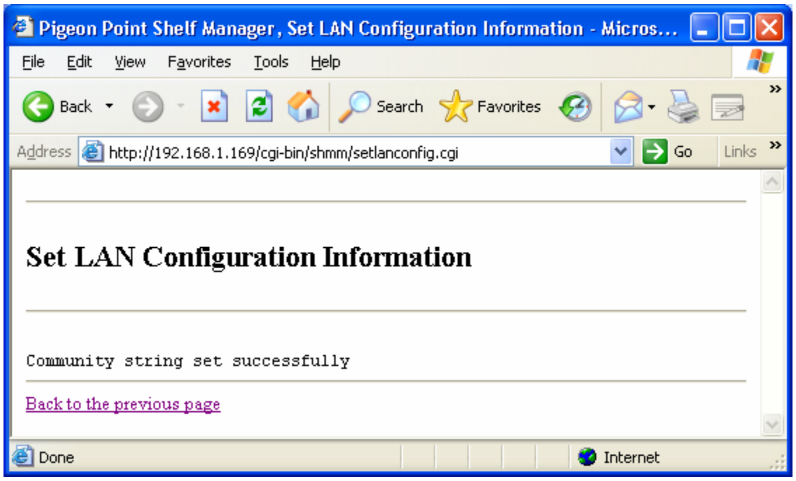
Set selector (if necessary):

Enter the parameter value:

Press Submit to set the LAN parameter information: Submit

Back to the main page

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".



Pigeon Point Shelf Manager, Set LAN Configuration Information - Micros...

Address: http://192.168.1.169/cgi-bin/shmm/setlanconfig.cgi

Set LAN Configuration Information

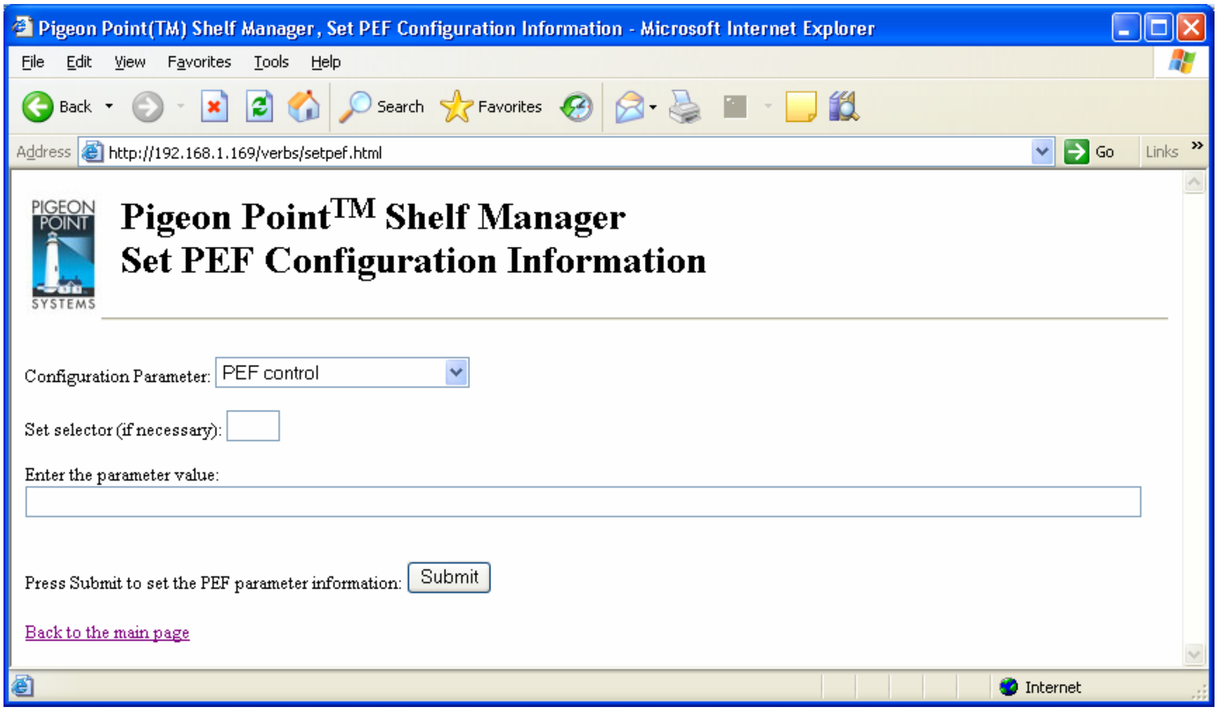
```
Community string set successfully
```

Back to the previous page

4.2.25. 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 section 3.42 of this document, which describes the CLI command "setpefconfig".


Pigeon Point(TM) Shelf Manager, Set PEF Configuration Information - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
Go
Links
PIGEON POINT SYSTEMS
Submit
Internet

http://192.168.1.169/verbs/setpef.html

Pigeon Point^{TM} Shelf Manager
Set PEF Configuration Information

Configuration Parameter: PEF control

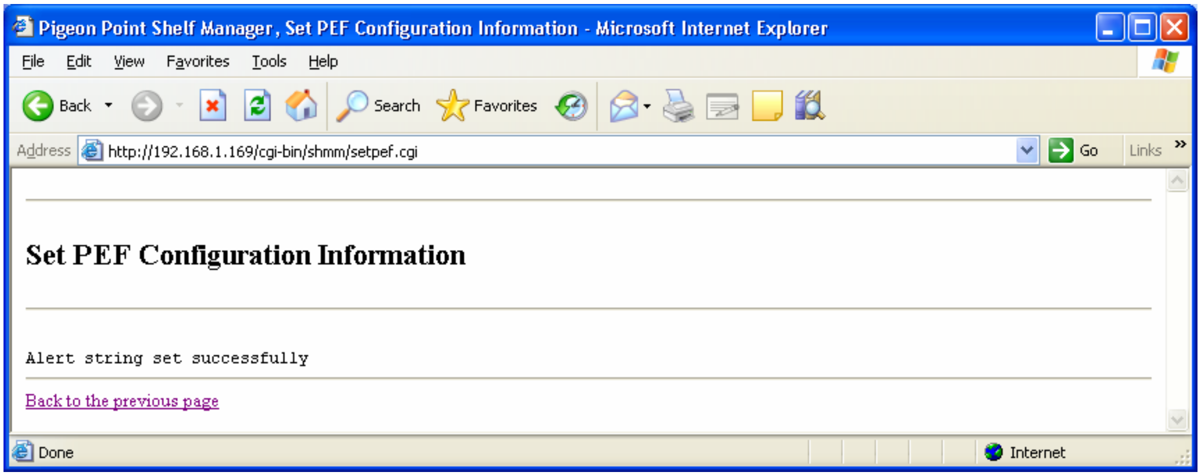
Set selector (if necessary):

Enter the parameter value:

Press Submit to set the PEF parameter information:

Back to the main page

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".


Pigeon Point Shelf Manager, Set PEF Configuration Information - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://192.168.1.169/cgi-bin/shmm/setpef.cgi
Go Links
Done
Internet

Set PEF Configuration Information

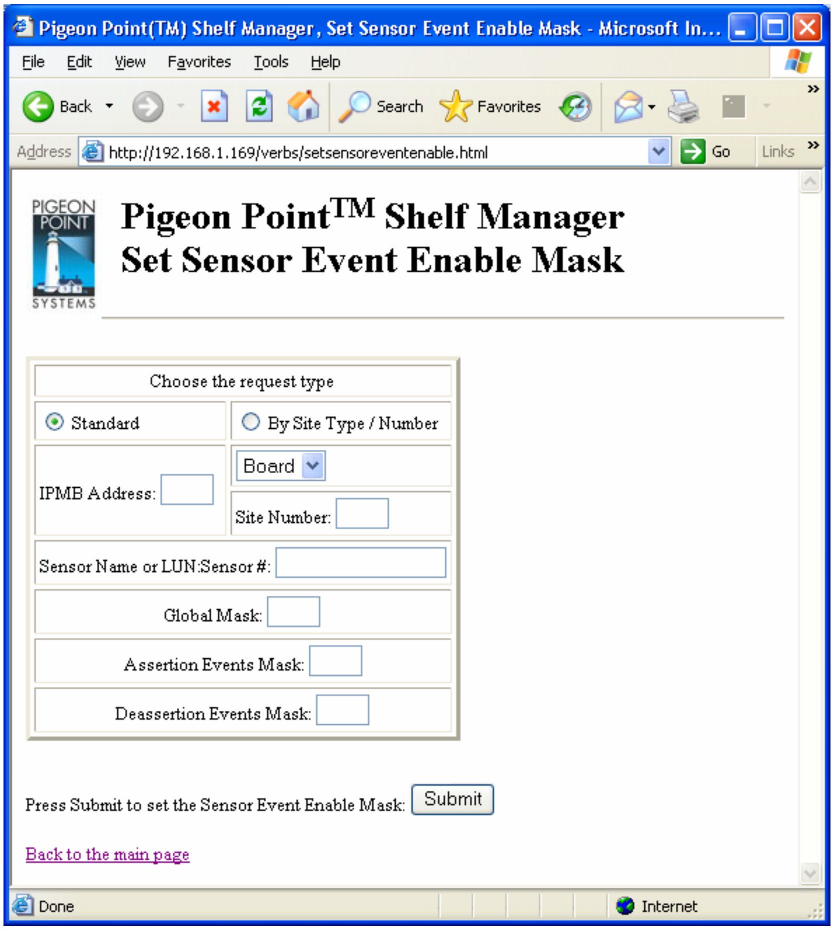
Alert string set successfully

Back to the previous page

4.2.26. Set Sensor Event Enable Mask

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.



Pigeon Point(TM) Shelf Manager, Set Sensor Event Enable Mask - Microsoft In...
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://192.168.1.169/verbs/setsensoreventenable.html
Go
Links
Done
Internet

Pigeon Point^{TM} Shelf Manager
Set Sensor Event Enable Mask

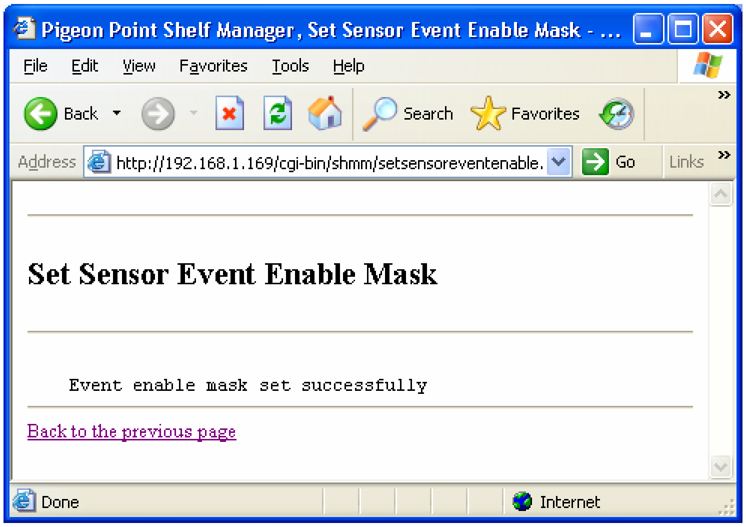
Choose the request type	
● Standard	○ By Site Type / Number
IPMB Address:	Board
	Site Number:
Sensor Name or LUN:Sensor #:	
Global Mask:	
Assertion Events Mask:	
Deassertion Events Mask:	

Press Submit to set the Sensor Event Enable Mask: Submit

Back to the main page

The field "Deassertion Events Mask" field may be left empty. The fields "Assertion Event Mask" and "Assertion Events Mask" may be left empty simultaneously. The other fields must be filled.

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".


Pigeon Point Shelf Manager, Set Sensor Event Enable Mask - ...
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://192.168.1.169/cgi-bin/shmm/setsensoreventenable.
Go
Links
Done
Internet

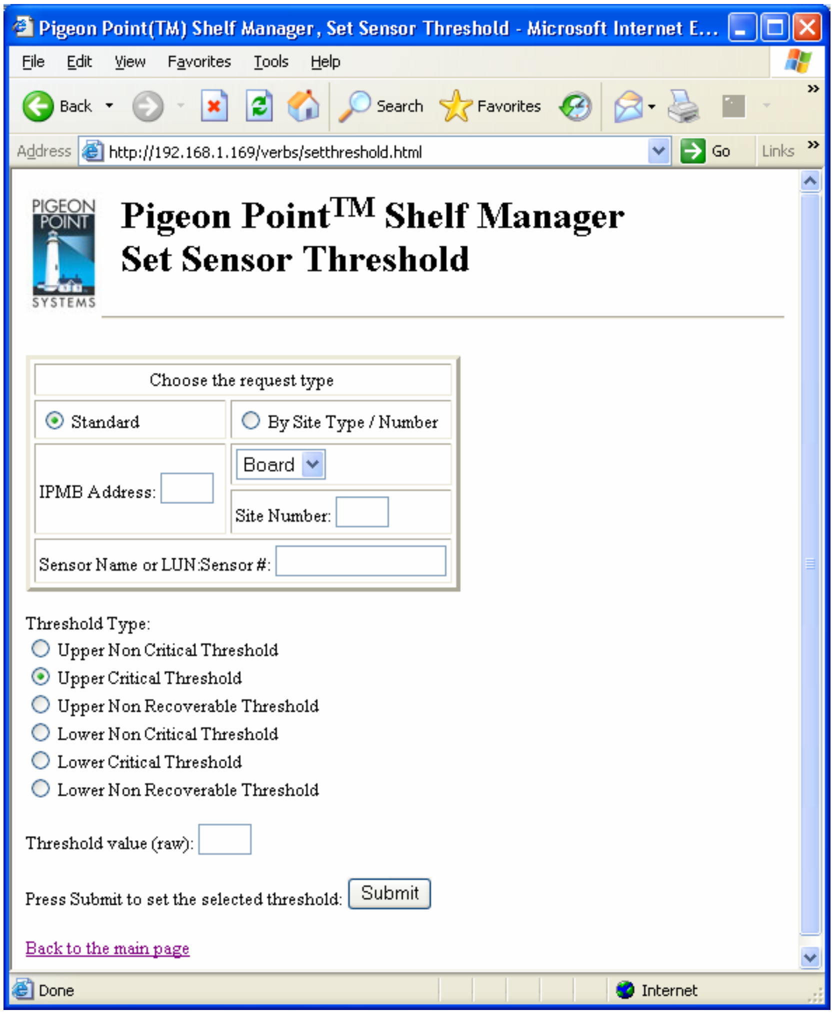
Set Sensor Event Enable Mask

```
Event enable mask set successfully
```

Back to the previous page

4.2.27. 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.



Address: http://192.168.1.169/verbs/setthreshold.html

Pigeon Point^{TM} Shelf Manager
Set Sensor Threshold

Choose the request type

● Standard ○ By Site Type / Number

IPMB Address:

Board

Site Number:

Sensor Name or LUN:Sensor #:

Threshold Type:

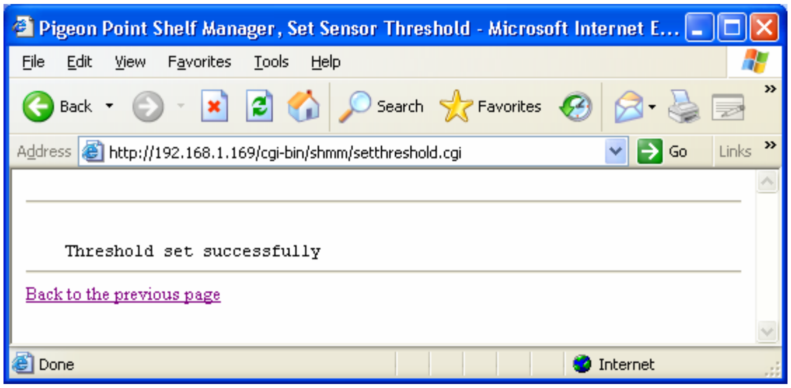
○ Upper Non Critical Threshold
● Upper Critical Threshold
○ Upper Non Recoverable Threshold
○ Lower Non Critical Threshold
○ Lower Critical Threshold
○ Lower Non Recoverable Threshold

Threshold value (raw):

Press Submit to set the selected threshold:

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”.

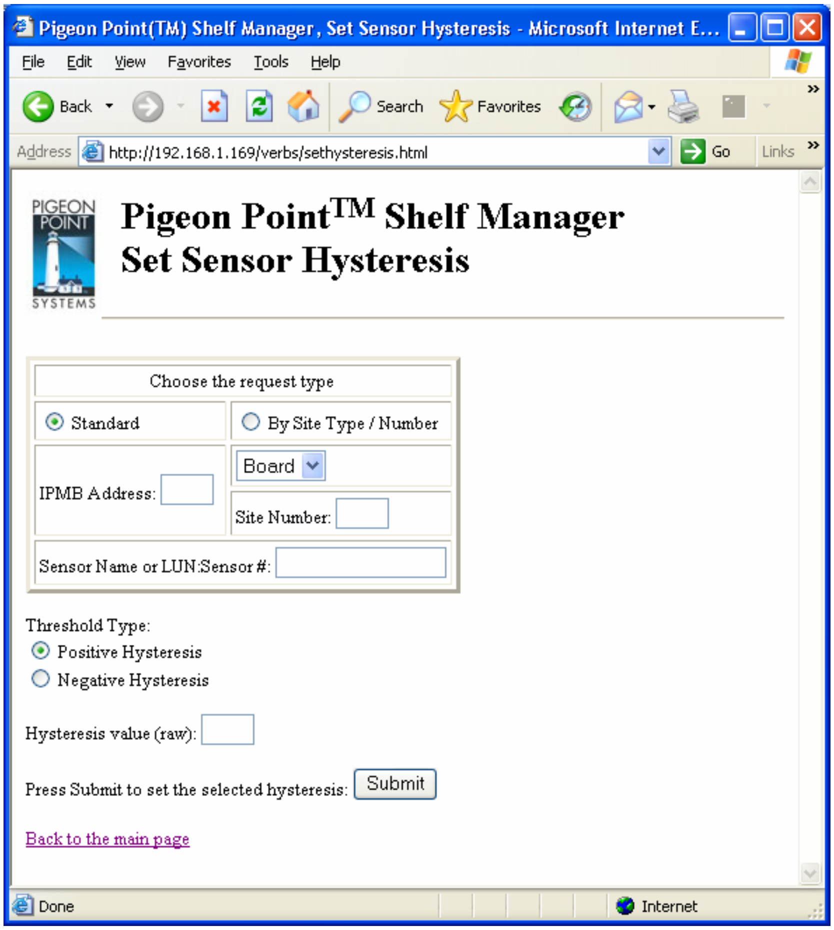

Pigeon Point Shelf Manager, Set Sensor Threshold - Microsoft Internet E...
File Edit View Favorites Tools Help
Back Search Favorites
Address http://192.168.1.169/cgi-bin/shmm/setthreshold.cgi Go Links
Done
Internet

```
Threshold set successfully
```

Back to the previous page

4.2.28. Set Sensor Hysteresis

The page "Set Sensor Hysteresis" allows the user to set value for the positive-going and negative-going hystereses 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.


Pigeon Point(TM) Shelf Manager, Set Sensor Hysteresis - Microsoft Internet E...
File Edit View Favorites Tools Help
Back Search Favorites
Address http://192.168.1.169/verbs/sethysteresis.html Go Links
PIGEON POINT SYSTEMS
Done
Internet

Pigeon Point^{TM} Shelf Manager
Set Sensor Hysteresis

Choose the request type	
● Standard	○ By Site Type / Number
IPMB Address: ▭	Board
	Site Number: ▭
Sensor Name or LUN:Sensor #: ▭	

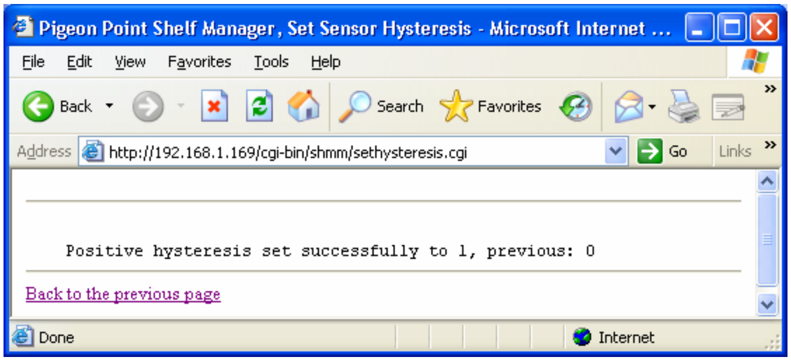
Threshold Type:
● Positive Hysteresis
○ Negative Hysteresis

Hysteresis value (raw): ▭

Press Submit to set the selected hysteresis: Submit

Back to the main page

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".

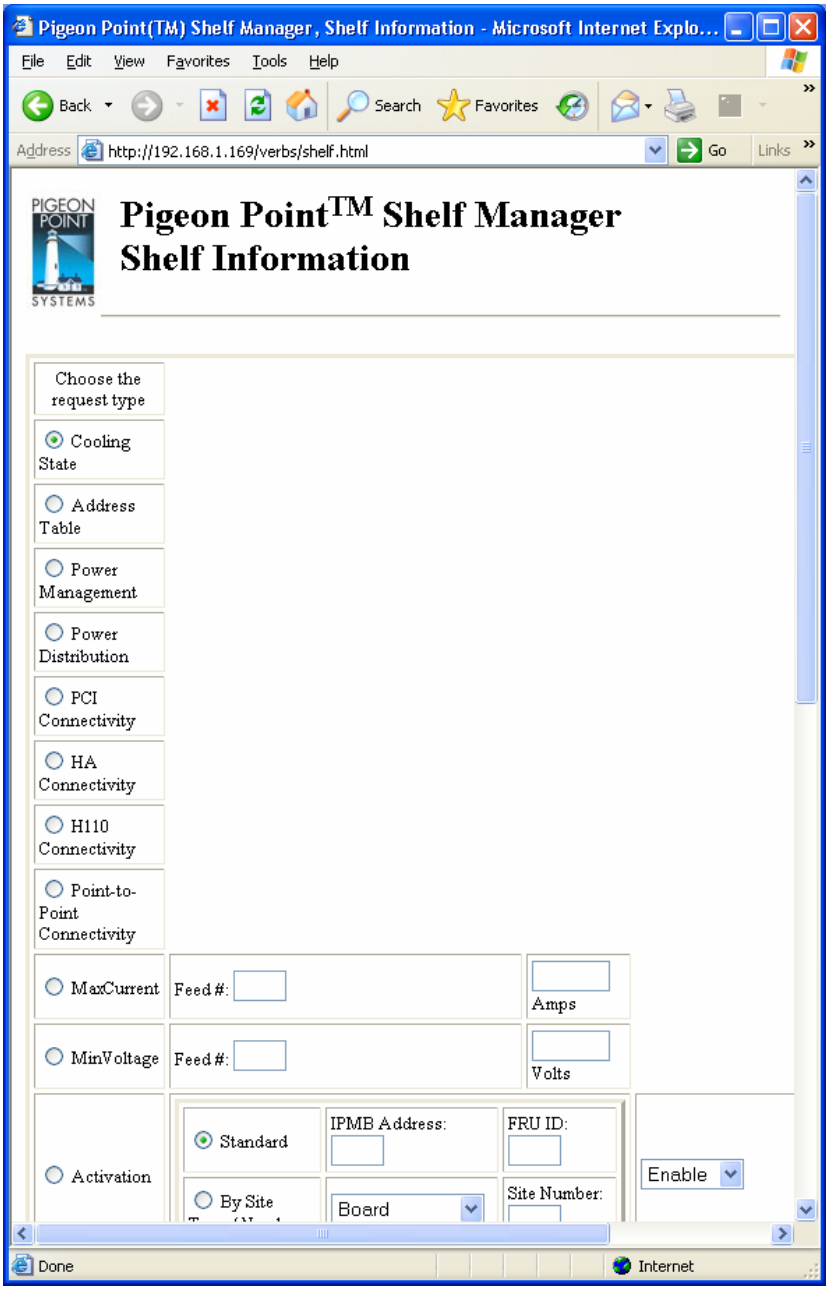

Pigeon Point Shelf Manager, Set Sensor Hysteresis - Microsoft Internet ...
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://192.168.1.169/cgi-bin/shmm/sethysteresis.cgi
Go
Links
Back to the previous page
Done
Internet

```
Positive hysteresis set successfully to 1, previous: 0
```

4.2.29. 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.


Pigeon Point(TM) Shelf Manager, Shelf Information - Microsoft Internet Explo...
File Edit View Favorites Tools Help
Back
Search
Favorites
Address http://192.168.1.169/verbs/shelf.html
Go
Links
PIGEON POINT SYSTEMS
Done
Internet

Pigeon Point™ Shelf Manager
Shelf Information

Choose the request type

● Cooling State

○ Address Table

○ Power Management

○ Power Distribution

○ PCI Connectivity

○ HA Connectivity

○ H110 Connectivity

○ Point-to-Point Connectivity

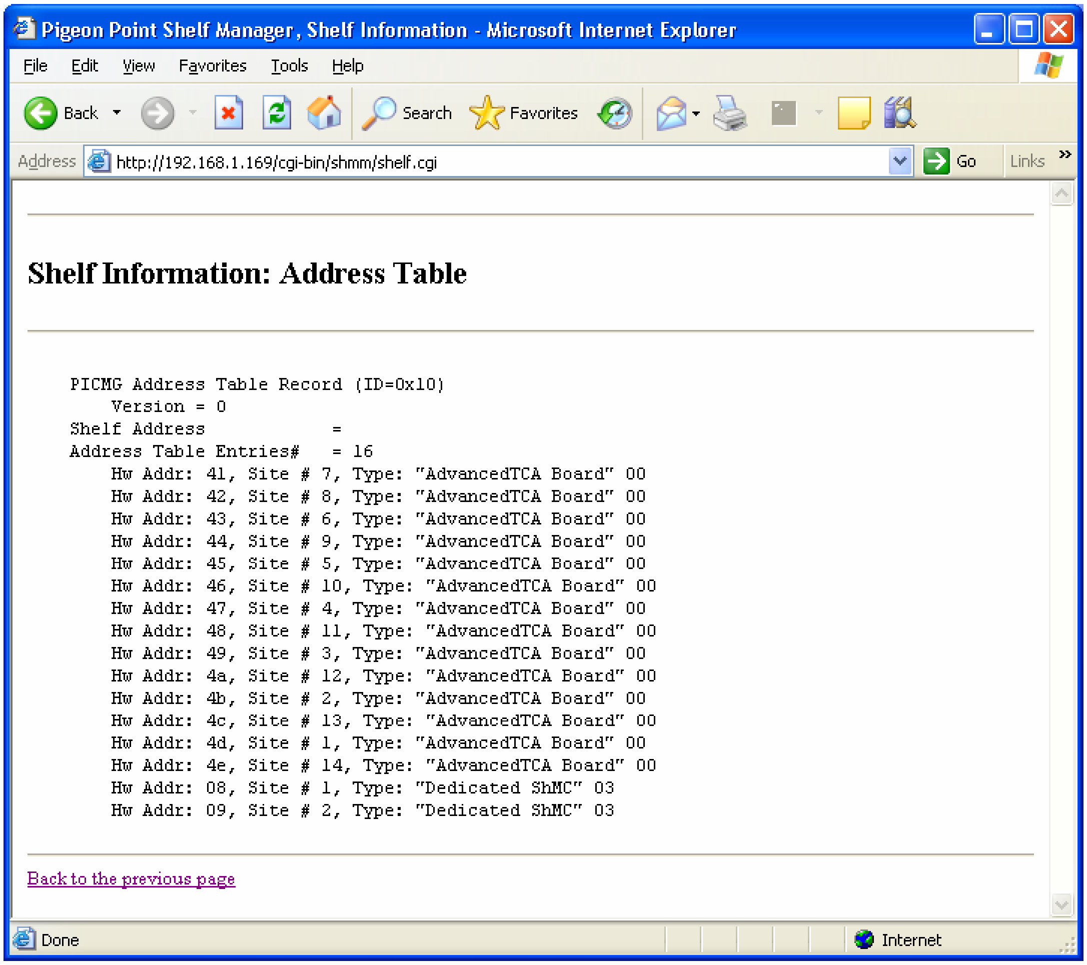
○ MaxCurrent Feed #: Amps

○ MinVoltage Feed #: Volts

○ Activation ● Standard IPMB Address: FRU ID: Enable

○ By Site Board Site Number:

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 <parameter>". The output page for the information type "Address Table" is shown below.


Pigeon Point Shelf Manager, Shelf Information - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://192.168.1.169/cgi-bin/shmm/shelf.cgi Go Links
Done Internet

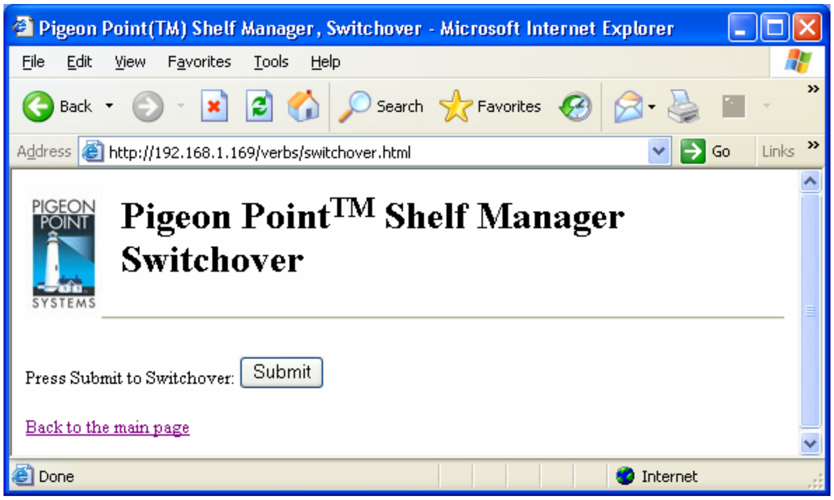
Shelf Information: Address Table

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

Back to the previous page

4.2.30. Switchover

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

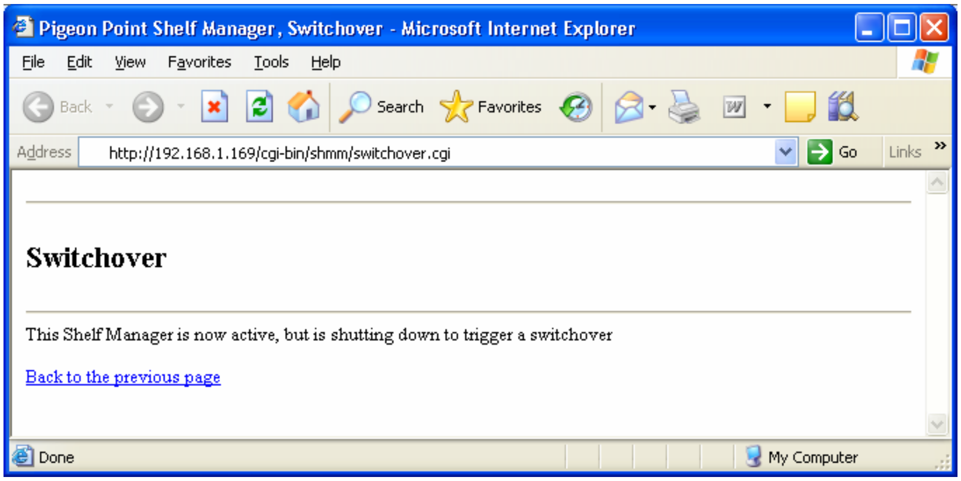

Pigeon Point(TM) Shelf Manager, Switchover - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://192.168.1.169/verbs/switchover.html Go Links
PIGEON POINT SYSTEMS
Done Internet

Pigeon Point^{TM} Shelf Manager Switchover

Press Submit to Switchover: Submit

Back to the main page

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.


Pigeon Point Shelf Manager, Switchover - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://192.168.1.169/cgi-bin/shmm/switchover.cgi
Go
Links
Done
My Computer

Switchover

This Shelf Manager is now active, but is shutting down to trigger a switchover

Back to the previous page

4.2.31. System Event Log

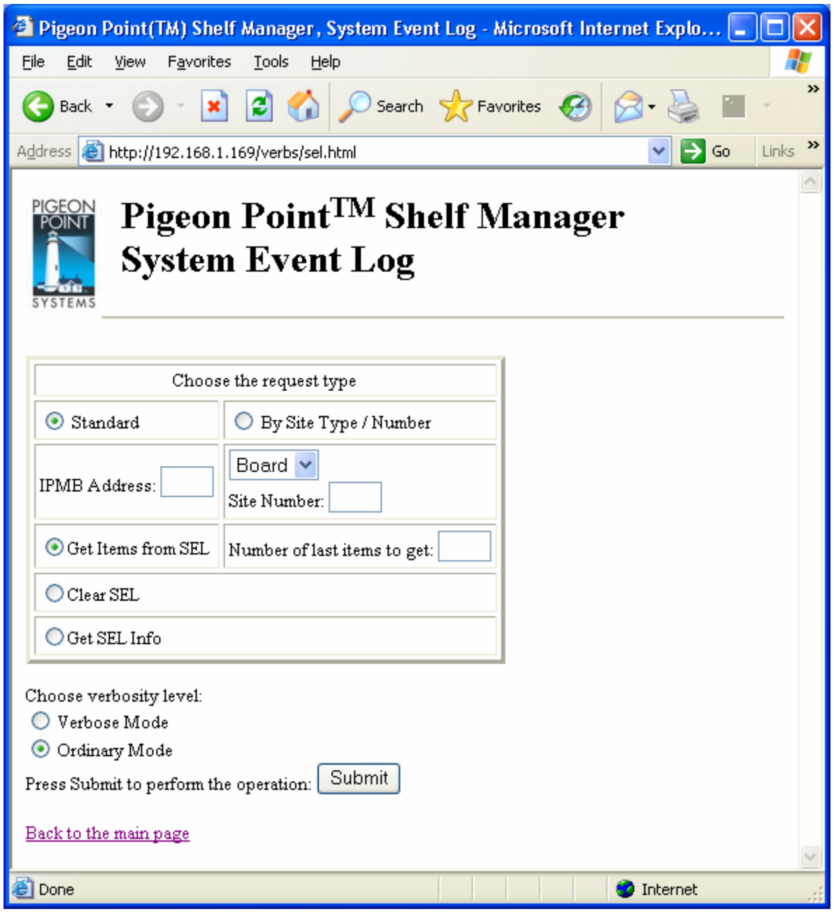
The page "System Event Log" allows the user to specify parameters for the System Event Log (SEL) information request or clear the event log.

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.


Pigeon Point(TM) Shelf Manager, System Event Log - Microsoft Internet Explo...
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://192.168.1.169/verbs/sel.html
Go
Links
PIGEON POINT SYSTEMS
Done
Internet

Pigeon Point™ Shelf Manager
System Event Log

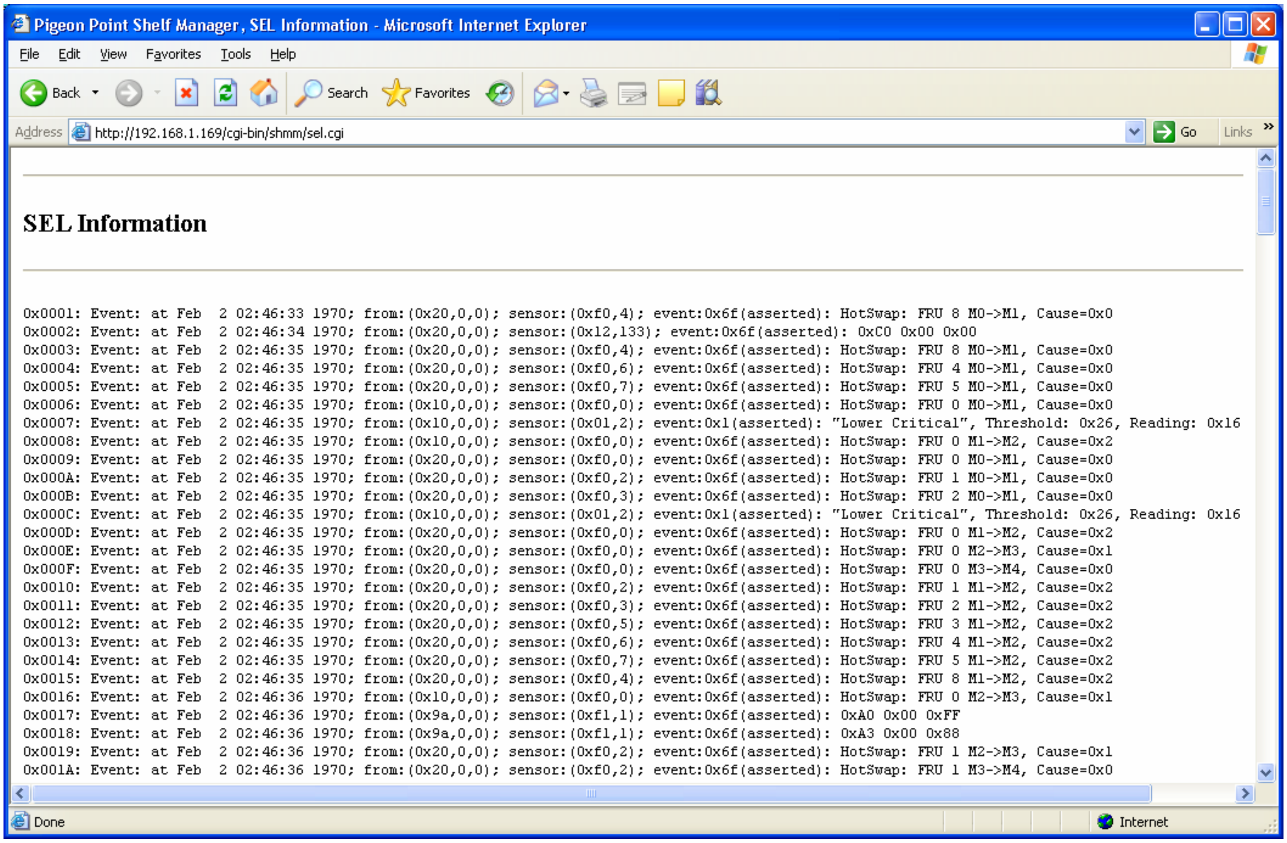
Choose the request type	
● Standard	○ By Site Type / Number
IPMB Address:	Board Site Number:
● Get Items from SEL	Number of last items to get:
○ Clear SEL	
○ Get SEL Info	

Choose verbosity level:
○ Verbose Mode
● Ordinary Mode

Press Submit to perform the operation: Submit

Back to the main page

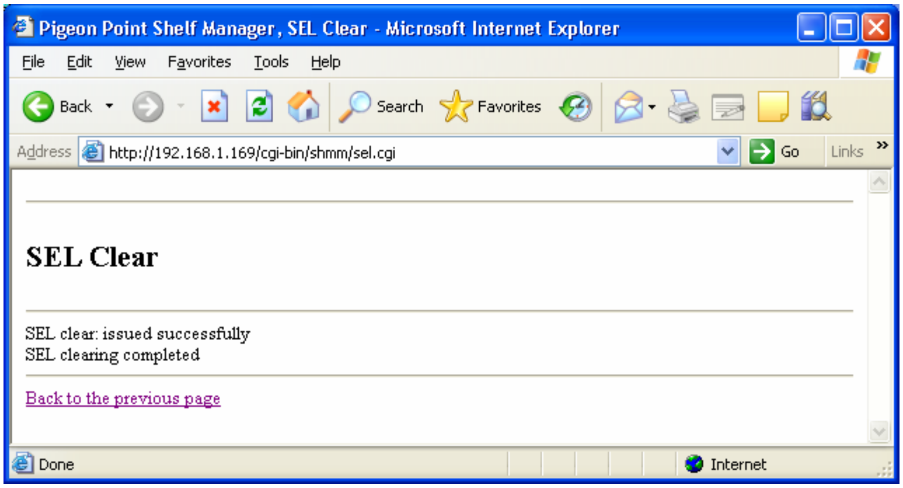
After the user chooses the upper radio option and fills in desired fields and clicks the "Submit" button, the request is executed and the results page is shown, similar to the one below. The output is essentially equal to the output produced by the CLI command "sel".


Pigeon Point Shelf Manager, SEL Information - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://192.168.1.169/cgi-bin/shmm/sel.cgi
Go Links
Done
Internet

SEL Information

```
0x0001: Event: at Feb  2 02:46:33 1970; from:(0x20,0,0); sensor:(0xf0,4); event:0x6f(asserted): HotSwap: FRU 8 M0->M1, Cause=0x0
0x0002: Event: at Feb  2 02:46:34 1970; from:(0x20,0,0); sensor:(0x12,133); event:0x6f(asserted): 0xC0 0x00 0x00
0x0003: Event: at Feb  2 02:46:35 1970; from:(0x20,0,0); sensor:(0xf0,4); event:0x6f(asserted): HotSwap: FRU 8 M0->M1, Cause=0x0
0x0004: Event: at Feb  2 02:46:35 1970; from:(0x20,0,0); sensor:(0xf0,6); event:0x6f(asserted): HotSwap: FRU 4 M0->M1, Cause=0x0
0x0005: Event: at Feb  2 02:46:35 1970; from:(0x20,0,0); sensor:(0xf0,7); event:0x6f(asserted): HotSwap: FRU 5 M0->M1, Cause=0x0
0x0006: Event: at Feb  2 02:46:35 1970; from:(0x10,0,0); sensor:(0xf0,0); event:0x6f(asserted): HotSwap: FRU 0 M0->M1, Cause=0x0
0x0007: Event: at Feb  2 02:46:35 1970; from:(0x10,0,0); sensor:(0x01,2); event:0x1(asserted): "Lower Critical", Threshold: 0x26, Reading: 0x16
0x0008: Event: at Feb  2 02:46:35 1970; from:(0x10,0,0); sensor:(0xf0,0); event:0x6f(asserted): HotSwap: FRU 0 M1->M2, Cause=0x2
0x0009: Event: at Feb  2 02:46:35 1970; from:(0x20,0,0); sensor:(0xf0,0); event:0x6f(asserted): HotSwap: FRU 0 M0->M1, Cause=0x0
0x000A: Event: at Feb  2 02:46:35 1970; from:(0x20,0,0); sensor:(0xf0,2); event:0x6f(asserted): HotSwap: FRU 1 M0->M1, Cause=0x0
0x000B: Event: at Feb  2 02:46:35 1970; from:(0x20,0,0); sensor:(0xf0,3); event:0x6f(asserted): HotSwap: FRU 2 M0->M1, Cause=0x0
0x000C: Event: at Feb  2 02:46:35 1970; from:(0x10,0,0); sensor:(0x01,2); event:0x1(asserted): "Lower Critical", Threshold: 0x26, Reading: 0x16
0x000D: Event: at Feb  2 02:46:35 1970; from:(0x20,0,0); sensor:(0xf0,0); event:0x6f(asserted): HotSwap: FRU 0 M1->M2, Cause=0x2
0x000E: Event: at Feb  2 02:46:35 1970; from:(0x20,0,0); sensor:(0xf0,0); event:0x6f(asserted): HotSwap: FRU 0 M2->M3, Cause=0x1
0x000F: Event: at Feb  2 02:46:35 1970; from:(0x20,0,0); sensor:(0xf0,0); event:0x6f(asserted): HotSwap: FRU 0 M3->M4, Cause=0x0
0x0010: Event: at Feb  2 02:46:35 1970; from:(0x20,0,0); sensor:(0xf0,2); event:0x6f(asserted): HotSwap: FRU 1 M1->M2, Cause=0x2
0x0011: Event: at Feb  2 02:46:35 1970; from:(0x20,0,0); sensor:(0xf0,3); event:0x6f(asserted): HotSwap: FRU 2 M1->M2, Cause=0x2
0x0012: Event: at Feb  2 02:46:35 1970; from:(0x20,0,0); sensor:(0xf0,5); event:0x6f(asserted): HotSwap: FRU 3 M1->M2, Cause=0x2
0x0013: Event: at Feb  2 02:46:35 1970; from:(0x20,0,0); sensor:(0xf0,6); event:0x6f(asserted): HotSwap: FRU 4 M1->M2, Cause=0x2
0x0014: Event: at Feb  2 02:46:35 1970; from:(0x20,0,0); sensor:(0xf0,7); event:0x6f(asserted): HotSwap: FRU 5 M1->M2, Cause=0x2
0x0015: Event: at Feb  2 02:46:35 1970; from:(0x20,0,0); sensor:(0xf0,4); event:0x6f(asserted): HotSwap: FRU 8 M1->M2, Cause=0x2
0x0016: Event: at Feb  2 02:46:36 1970; from:(0x10,0,0); sensor:(0xf0,0); event:0x6f(asserted): HotSwap: FRU 0 M2->M3, Cause=0x1
0x0017: Event: at Feb  2 02:46:36 1970; from:(0x9a,0,0); sensor:(0xf1,1); event:0x6f(asserted): 0xA0 0x00 0xFF
0x0018: Event: at Feb  2 02:46:36 1970; from:(0x9a,0,0); sensor:(0xf1,1); event:0x6f(asserted): 0xA3 0x00 0x88
0x0019: Event: at Feb  2 02:46:36 1970; from:(0x20,0,0); sensor:(0xf0,2); event:0x6f(asserted): HotSwap: FRU 1 M2->M3, Cause=0x1
0x001A: Event: at Feb  2 02:46:36 1970; from:(0x20,0,0); sensor:(0xf0,2); event:0x6f(asserted): HotSwap: FRU 1 M3->M4, Cause=0x0
```

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".

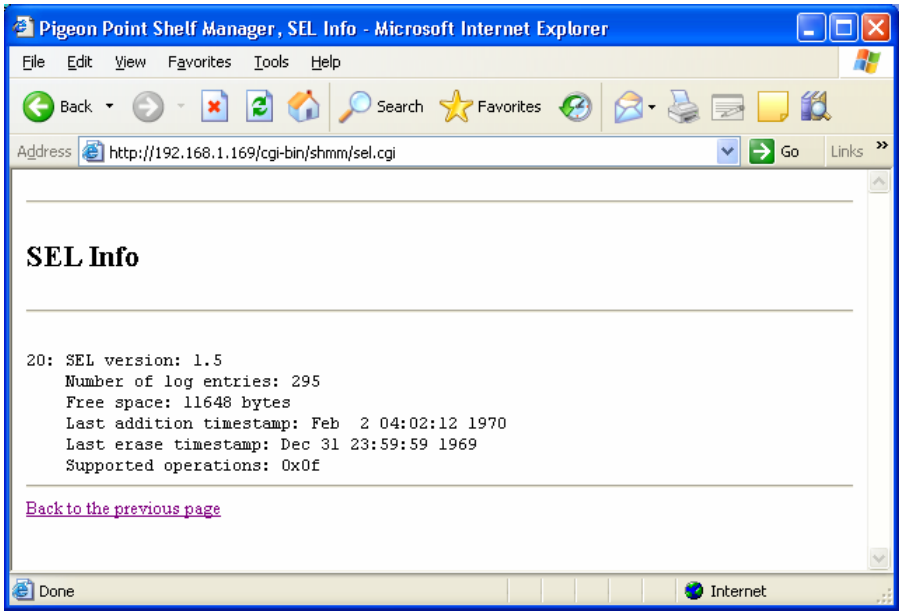

Pigeon Point Shelf Manager, SEL Clear - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://192.168.1.169/cgi-bin/shmm/sel.cgi
Go Links
Done
Internet

SEL Clear

SEL clear: issued successfully
SEL clearing completed

Back to the previous page

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://192.168.1.169/cgi-bin/shmm/sel.cgi
Go Links
Done
Internet

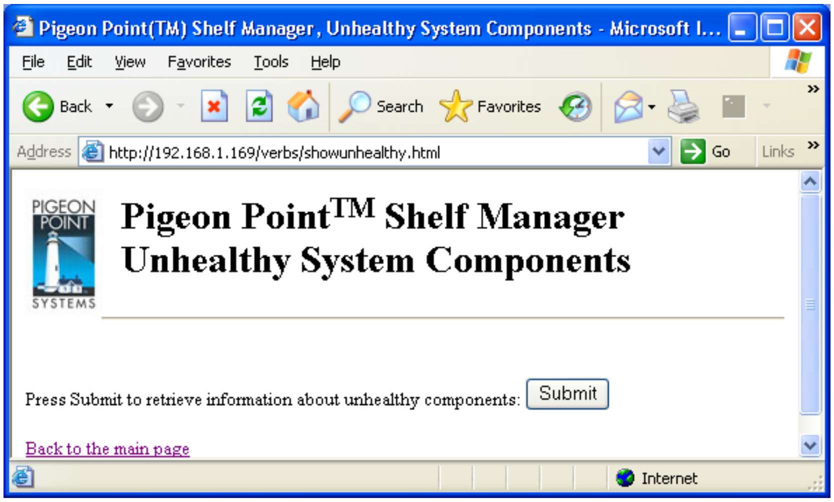
SEL Info

```
20: SEL version: 1.5
    Number of log entries: 295
    Free space: 11648 bytes
    Last addition timestamp: Feb  2 04:02:12 1970
    Last erase timestamp: Dec 31 23:59:59 1969
    Supported operations: 0x0f
```

Back to the previous page

4.2.32. 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.



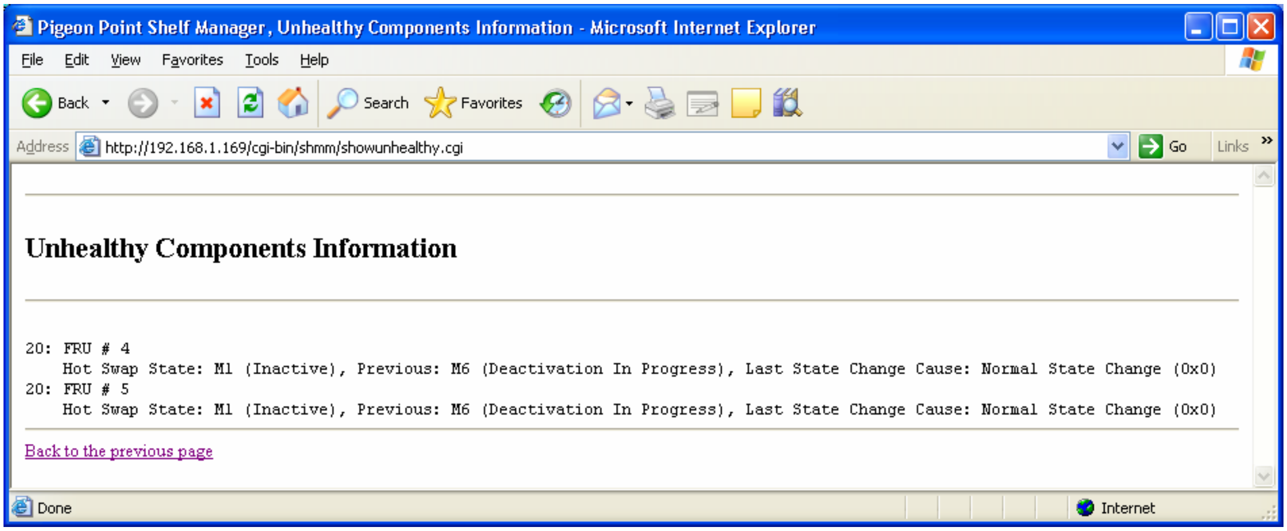
Pigeon Point(TM) Shelf Manager, Unhealthy System Components - Microsoft I...
File Edit View Favorites Tools Help
Back Search Favorites
Address http://192.168.1.169/verbs/showunhealthy.html
Go Links
PIGEON POINT SYSTEMS
Submit
Internet

Pigeon Point^{TM} Shelf Manager
Unhealthy System Components

Press Submit to retrieve information about unhealthy components:

Back to the main page

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”.


Pigeon Point Shelf Manager, Unhealthy Components Information - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://192.168.1.169/cgi-bin/shmm/showunhealthy.cgi
Go Links
Done
Internet

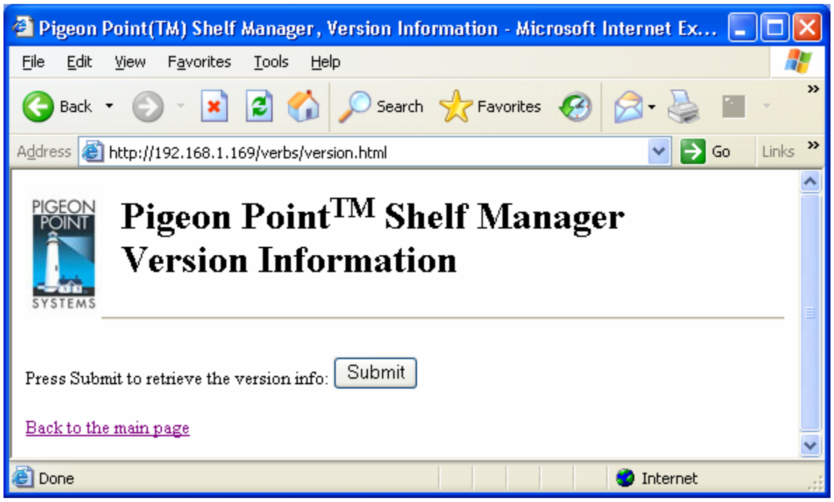
Unhealthy Components Information

```
20: FRU # 4
    Hot Swap State: M1 (Inactive), Previous: M6 (Deactivation In Progress), Last State Change Cause: Normal State Change (0x0)
20: FRU # 5
    Hot Swap State: M1 (Inactive), Previous: M6 (Deactivation In Progress), Last State Change Cause: Normal State Change (0x0)
```

Back to the previous page

4.2.33. Version

The page "Version" allows the user to request information about the software version of the Pigeon Point Shelf Manager.

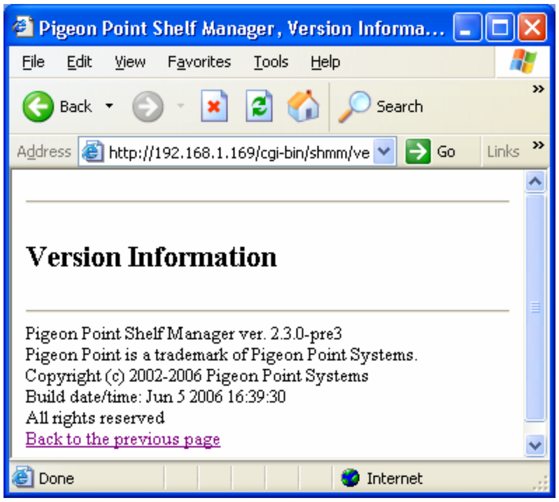

Pigeon Point(TM) Shelf Manager, Version Information - Microsoft Internet Ex...
File Edit View Favorites Tools Help
Back Search Favorites
Address http://192.168.1.169/verbs/version.html
Go Links
PIGEON POINT SYSTEMS
Submit
Done
Internet

Pigeon Point^{TM} Shelf Manager
Version Information

Press Submit to retrieve the version info:

Back to the main page

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".


Pigeon Point Shelf Manager, Version Informa...
File Edit View Favorites Tools Help
Back
Search
Address http://192.168.1.169/cgi-bin/shmm/ve
Go
Links
Done
Internet

Version Information

Pigeon Point Shelf Manager ver. 2.3.0-pre3
Pigeon Point is a trademark of Pigeon Point Systems.
Copyright (c) 2002-2006 Pigeon Point Systems
Build date/time: Jun 5 2006 16:39:30
All rights reserved
Back to the previous page

Chapter

5

5. SNMP Interface

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).pro
ducts(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 an 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 5.2.6) are supported.

Using the existing ARMBoot 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[1]. 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.

CompactPCI 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.

The following variables are defined for each board slot:

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.

[1] Note: only boards that are described in the Address Table are accessible. In the case of CompactPCI shelves, only the CompactPCI Site boards (0xC4) and in the case of ATCA systems, only the ATCA Site boards (0x00) are accessible.

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.
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 (0x20). 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 (0x20), 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”
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	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 board FRU information

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[1]. 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.

The following variables are defined for each fan tray slot:

Variable	**Index**	**Type**	**Access Mode**	**Description**
fantray-slot-number	1	INTEGER	Read-only	Table entry index, equal to `<fantraynum>`

[1] Note: only fan trays (0x04) that are described in the Address Table are accessible.

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.
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 (0x20). 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 (0x20), 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”
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

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 the system[1]. 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.

The following variables are defined for each power supply slot:

Variable	Index	Type	Access Mode	Description
powersupply-slot-number	1	INTEGER	Read-only	Table entry index, equal to `<powersupplynum>` This variable is available in 2.x systems only.
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.
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.

[1] Note: only Power Supplies (0xC5) that described in the Address Table are accessible

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 (0x20). If the unit is installed and has an IPM controller, return the IPM controller's slave 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 (0x20), 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-area-present	14	INTEGER	Read-only	1 – if the board area is 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

For example, to check the degrade state of the Power supply # 3, use the following OID:

```
<ROOT_OID>.34.1.2.3
```

5.1.4. Shelf Manager Variables

The variables defined in this section contain information about the Shelf Managers in the system[1]. 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.

The following variables are defined for each shelf manager slot:

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

[1] Note: only dedicated shelf managers (0x03) that described in the Address Table are accessible.

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)
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"

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

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 table. 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.

The following variables are defined for each chassis slot:

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
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

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.

The following variables are defined for each shelf manager slot:

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	Event class other (0), temperature (1), voltage (2), current (3), fan (4), HotSwap ('F0'H), PowerState ('E1'H)

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.

5.2. 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.

The following variables are defined for each IPM Controller:

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
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.

The following variables are defined for each FRU device:

Variable	**Index**	**Type**	**Access Mode**	**Description**
fru-device-index	1	INTEGER	Read-only	Table entry index, equal to (<ipmb_addr> << 16) |<fru_id>)
fru-device-sdr-version	2	INTEGER	Read-only	*SDR Version* of the FRU Device or Management Controller Device Locator Record for this FRU

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
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).
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).
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
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
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

fru-device-hot-swap-state	11	INTEGER	Read-only	Current PICMG 3.0 FRU state (M0...M7) for this FRU. If this variable is equal to *n*, that means that the FRU is in state M*n*.
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 0x20 = 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) or Compact Sensor Records (SDR Type 02h) exist in the shelf.

FRU device information 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 the 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.

The following variables are defined for each sensor:

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
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
sensor-initialization	9	INTEGER	Read-only	*Sensor Initialization* as defined in the Sensor Record
sensor-capabilities	10	INTEGER	Read-only	*Sensor Capabilities* as defined in the Sensor Record
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
sensor-deassertion-event-mask	14	INTEGER	Read-only	*Deassertion Event Mask / Upper Threshold Reading Mask* as defined in the Sensor Record
sensor-mask	15	INTEGER	Read-only	*Discrete Reading Mask / Settable Threshold Mask, Readable Threshold Mask* as defined in the Sensor Record
sensor-unit1	16	INTEGER	Read-only	*Sensor Units 1* as defined in the Sensor Record

sensor-unit2	17	INTEGER	Read-only	*Sensor Units 2 – Base Unit* as defined in the Sensor Record
sensor-unit3	18	INTEGER	Read-only	*Sensor Units 3 – Modifier Unit* as defined in the Sensor Record
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.
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.
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.
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.
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 1 for Compact 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.
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.
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.
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.

sensor-reading	28	INTEGER	Read-only	Current sensor reading in raw form.
sensor-processed-reading	29	DisplayString (SIZE(0..255))	Read-only	Current sensor reading processed according to reading conversion formula for this sensor.
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.
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.
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.
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.
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.
sensor-upper-non-recoverable-threshold	35	INTEGER	Read-write	Current value of the *Upper non-recoverable Threshold* for the specified sensor
sensor-upper-critical-threshold	36	INTEGER	Read-write	Current value of the *Upper critical Threshold* for the specified sensor.
sensor-upper-non-critical-threshold	37	INTEGER	Read-write	Current value of the *Upper non-critical Threshold* for the specified sensor.
sensor-lower-non-recoverable-threshold	38	INTEGER	Read-write	Current value of the *Lower non-recoverable Threshold* for the specified sensor.
sensor-lower-critical-threshold	39	INTEGER	Read-write	Current value of the *Lower critical Threshold* for the specified sensor..
sensor-lower-non-critical-threshold	40	INTEGER	Read-write	Current value of the *Lower non-critical Threshold* for the specified sensor.
sensor-positive-going-threshold-hysteresis	41	INTEGER	Read-write	Current value of the *Positive-going Threshold Hysteresis* for the specified sensor.
sensor-negative-going-threshold-hysteresis	42	INTEGER	Read-write	Current value of the *Negative-going Threshold Hysteresis* for the specified sensor.

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

For example, to get the ID String of the second sensor on the IPM controller at IPMB address 0x20 = 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.

AdvancedTCA Board slot 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.

The following variables are defined for each board slot:

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
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. 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.

The following variables are defined for each SEL record:

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
```

5.2.6. 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 created 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.

The following variables are defined for each shelf:

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
```

5.2.7. 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 LAN channel. Table entries are indexed by IPMI channel numbers. The current release of the Shelf Manager software supports only one LAN channel – IPMI channel #1.

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 only one LAN channel with number 1 is supported. The current release also has a fixed number of supported destinations – 16. Thus the SNMP variables for *Destination Type* and *Destination Addresses* parameters are implemented as fixed-sized arrays.

The following variables are defined for each LAN channel:

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
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
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

For example, to get the IP address of channel #1, use the following OID:

```
<ROOT_OID>.7.1.5.1
```

5.2.8. 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.

5.2.9. Scalar Variables

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 LAN channel configuration. The variable indices are defined in the table below.

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
```

5.2.10. Event Filter Table

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.

The following variables are defined for each PEF Event Filter:

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
```

5.2.11. Alert Policy Table

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.

The following variables are defined for each PEF Alert Policy:

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
```

5.2.12. Alert Strings Table

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.

The following variables are defined for each PEF Alert String:

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
```

5.2.13. FRU Information Table

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.

The following variables are defined for each FRU Info block:

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 the PEF Configuration Alert String Key for string #8, use the following OID:

```
<ROOT_OID>.19.1.2.9
```

5.2.14. 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.

The following variables are defined for each FRU device:

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
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
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

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).
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).
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
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
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
fru-device-by-site -hot-swap-state	11	INTEGER	Read-only	Current PICMG 3.0 FRU state (M0...M7) for this FRU. If this variable is equal to *n*, that means that the FRU is in state M*n*.

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>.2.1.10.2.1
```

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 / 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 like this:

For example, to retrieve the variable controller-sdr-version for the controller 0x20 (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
```

It will produce output similar to the following:

```
 PPS-SENTRY-MIB::ipm-controller-sdr-version.32 = INTEGER: 81
```

To retrieve the whole 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
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 0x20 (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

###############################################################################
#
# 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 Delaty
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
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 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.

Chapter 6

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 authentification 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, and 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 following table shows the IPMI commands implemented by the Shelf Manager. Due to sectuirty considerations, the treatment of a given command may be different, depending on whether it is received over the RMCP interface or on IPMB-0.

Command	NetFn	CMD	Arriving from RMCP Interface	Arriving from IPM Controllers
GetDeviceID	App	0x01	Supported	Supported
ColdReset	App	0x02	Supported	Supported
WarmReset	App	0x03	Not supported	Not supported
GetSelfTestResults	App	0x04	Supported	Supported
Manufacturing Test On	App	0x05	Not supported	Not supported
Set ACPI Power State	App	0x06	Supported	Supported
Get ACPI Power State	App	0x07	Supported	Supported
Get Device GUID	App	0x08	Supported	Supported

Reset Watchdog Timer	App	0x22	Supported	Supported
Set Watchdog Timer	App	0x24	Supported	Supported
Get Watchdog Timer	App	0x25	Supported	Supported
Set BMC Global Enables	App	0x2E	Supported	Supported
Get BMC Global Enables	App	0x2F	Supported	Supported
Clear Message Flags	App	0x30	Supported	Supported
Get Message Flags	App	0x31	Supported	Supported
Enable Message Channel Receive	App	0x32	Not supported	Not supported
Get Message	App	0x33	Not supported	Not supported
Send Message	App	0x34	Supported	Supported
Read Event Message Buffer	App	0x35	Not supported	Not supported
Get BT Interface Capabilities	App	0x36	Not supported	Not supported
Get System GUID	App	0x37	Supported	Supported
Get Channel Authentication Capabilities	App	0x38	Supported	Supported(*)
Get Session Challenge	App	0x39	Supported	Not supported
Activate Session	App	0x3A	Supported	Not supported
Set Session Privilege Level	App	0x3B	Supported	Not supported
Close Session	App	0x3C	Supported	Not supported
Get Session Info	App	0x3D	Supported	Supported(*)
Get AuthCode	App	0x3F	Supported	Supported(*)
Set Channel Access	App	0x40	Supported	Supported(*)
Get Channel Access	App	0x41	Supported	Supported(*)
Get Channel Info	App	0x42	Supported	Supported(*)
Set User Access	App	0x43	Supported	Supported(*)
Get User Access	App	0x44	Supported	Supported(*)
Set User Name	App	0x45	Supported	Supported(*)
Get User Name	App	0x46	Supported	Supported(*)
Set User Password	App	0x47	Supported	Supported(*)
Activate Payload	App	0x48	Not supported	Not supported
Deactivate Payload	App	0x49	Not supported	Not supported
Get Payload Activation Status	App	0x4A	Not supported	Not supported
Get Payload Instance Info	App	0x4B	Not supported	Not supported
Set User Payload Access	App	0x4C	Not supported	Not supported
Get User Payload Access	App	0x4D	Not supported	Not supported
Get Channel Payload Support	App	0x4E	Not supported	Not supported
Get Channel Payload Version	App	0x4F	Not supported	Not supported
Get Channel OEM Payload Info	App	0x50	Not supported	Not supported
Master Write-Read	App	0x52	Not supported	Not supported
Get Channel Cipher Suites	App	0x54	Not supported	Not supported
Suspend/Resume Payload Encryption	App	0x55	Not supported	Not supported
Set Channel Security Keys	App	0x56	Not supported	Not supported

Get System Interface Capabilities	App	0x57	Not supported	Not supported
Get Chassis Capabilities	Chassis	0x00	Supported	Supported
Get Chassis Status	Chassis	0x01	Supported	Supported
Chassis Control	Chassis	0x02	Supported	Supported
Chassis Reset	Chassis	0x03	Not supported	Not supported
Chassis Identify	Chassis	0x04	Not supported	Not supported
Set Chassis Capabilities	Chassis	0x05	Supported	Supported
Set Power Restore Policy	Chassis	0x06	Not supported	Not supported
Get System Restart Cause	Chassis	0x07	Not supported	Not supported
Set System Boot Options	Chassis	0x08	Not supported	Not supported
Get System Boot Options	Chassis	0x09	Not supported	Not supported
Set Front Panel Button Enables	Chassis	0x0A	Not supported	Not supported
Set Power Cycle Interval	Chassis	0x0B	Not supported	Not supported
Get POH Counter	Chassis	0x0F	Not supported	Not supported
Set LAN Configuration Parameters	Transport	0x01	Supported	Supported(*)
Get LAN Configuration Parameters	Transport	0x02	Supported	Supported
Suspend BMC ARPs	Transport	0x03	Supported	Supported(*)
Get IP/UDP/RMCP statistics	Transport	0x04	Not supported	Not supported
Set Serial/Modem Configuration	Transport	0x10	Not supported	Not supported
Get Serial/Modem Configuration	Transport	0x11	Not supported	Not supported
Set Serial/Modem Mux	Transport	0x12	Not supported	Not supported
Get TAP Response Codes	Transport	0x13	Not supported	Not supported
Set PPP UDP Proxy Transmit Data	Transport	0x14	Not supported	Not supported
Get PPP UDP Proxy Transmit Data	Transport	0x15	Not supported	Not supported
Send PPP UDP Proxy Packet	Transport	0x16	Not supported	Not supported
Get PPP UDP Proxy Receive Data	Transport	0x17	Not supported	Not supported
Serial/Modem Connection Active	Transport	0x18	Not supported	Not supported
Callback	Transport	0x19	Not supported	Not supported
Set User Callback Options	Transport	0x1A	Supported	Supported(*)
Get User Callback Options	Transport	0x1B	Supported	Supported(*)
SOL Activating	Transport	0x20	Not supported	Not supported
Set SOL Configuration Parameters	Transport	0x21	Not supported	Not supported
Get SOL Configuration Parameters	Transport	0x22	Not supported	Not supported
Get FRU Inventory Area Info	Storage	0x10	Supported	Supported

Read FRU Data	Storage	0x11	Supported	Supported
Write FRU Data	Storage	0x12	Supported	Supported
Get SDR Repository Info	Storage	0x20	Supported	Supported
Get SDR Repository Allocation Info	Storage	0x21	Not supported	Not supported
Reserve SDR Repository	Storage	0x22	Supported	Supported
Get SDR	Storage	0x23	Supported	Supported
Add SDR	Storage	0x24	Supported	Supported
Partial Add SDR	Storage	0x25	Supported	Supported
Delete SDR	Storage	0x26	Supported	Supported
Clear SDR Repository	Storage	0x27	Supported	Supported
Get SDR Repository Time	Storage	0x28	Supported	Supported
Set SDR Repository Time	Storage	0x29	Supported	Supported
Enter SDR Repository Update Mode	Storage	0x2A	Not supported	Not supported
Exit SDR Repository Update Mode	Storage	0x2B	Not supported	Not supported
Run Initialization Agent	Storage	0x2C	Not supported	Not supported
Get SEL Info	Storage	0x40	Supported	Supported
Get SEL Allocation Info	Storage	0x41	Supported	Supported
Reserve SEL	Storage	0x42	Supported	Supported
Get SEL Entry	Storage	0x43	Supported	Supported
Add SEL Entry	Storage	0x44	Supported	Supported
Partial Add SEL Entry	Storage	0x45	Supported	Supported
Delete SEL Entry	Storage	0x46	Supported	Supported
Clear SEL	Storage	0x47	Supported	Supported
Get SEL Time	Storage	0x48	Supported	Supported
Set SEL Time	Storage	0x49	Supported	Supported
Get Auxiliary Log Status	Storage	0x5A	Not supported	Not supported
Set Auxiliary Log Status	Storage	0x5B	Not supported	Not supported
Set Event Receiver	S/E	0x00	Supported	Supported
Get Event Receiver	S/E	0x01	Supported	Supported
Event Message	S/E	0x02	Supported	Supported
Get PEF Capabilities	S/E	0x10	Supported	Supported
Arm PEF Postpone Timer	S/E	0x11	Supported	Supported
Set PEF Configuration Parameters	S/E	0x12	Supported	Supported
Get PEF Configuration Parameters	S/E	0x13	Supported	Supported
Set Last Processed Event ID	S/E	0x14	Supported	Supported
Get Last Processed Event ID	S/E	0x15	Supported	Supported
Alert Immediate	S/E	0x16	Supported	Supported
PET Acknowledge	S/E	0x17	Supported	Supported
Get Device SDR Info	S/E	0x20	Supported	Supported
Get Device SDR	S/E	0x21	Supported	Supported

Reserve Device SDR Repository	S/E	0x22	Supported	Supported
Get Sensor Reading Factors	S/E	0x23	Supported	Supported
Set Sensor Hysteresis	S/E	0x24	Supported	Supported
Get Sensor Hysteresis	S/E	0x25	Supported	Supported
Set Sensor Threshold	S/E	0x26	Supported	Supported
Get Sensor Threshold	S/E	0x27	Supported	Supported
Set Sensor Event Enable	S/E	0x28	Supported	Supported
Get Sensor Event Enable	S/E	0x29	Supported	Supported
Re-arm Sensor Events	S/E	0x2A	Supported	Supported
Get Sensor Event Status	S/E	0x2B	Supported	Supported
Get Sensor Reading	S/E	0x2D	Supported	Supported
Set Sensor Type	S/E	0x2E	Supported	Supported
Get Sensor Type	S/E	0x2F	Supported	Supported
Get PICMG Properties	PICMG	0x00	Supported	Supported
Get Address Info	PICMG	0x01	Supported	Supported
Get Shelf Address Info	PICMG	0x02	Supported	Supported
Set Shelf Address Info	PICMG	0x03	Supported	Supported
FRU Control	PICMG	0x04	Supported	Supported
Get FRU LED Properties	PICMG	0x05	Supported	Supported
Get LED Color Capabilities	PICMG	0x06	Supported	Supported
Set FRU LED State	PICMG	0x07	Supported	Supported
Get FRU LED State	PICMG	0x08	Supported	Supported
Set IPMB State	PICMG	0x09	Supported	Supported
Set FRU Activation Policy	PICMG	0x0A	Supported	Supported
Get FRU Activation Policy	PICMG	0x0B	Supported	Supported
Set FRU Activation	PICMG	0x0C	Supported	Supported
Get Device Locator Record ID	PICMG	0x0D	Supported	Supported
Set Port State	PICMG	0x0E	Supported	Supported
Get Port State	PICMG	0x0F	Supported	Supported
Compute Power Properties	PICMG	0x10	Supported	Supported
Set Power Level	PICMG	0x11	Supported	Supported
Get Power Level	PICMG	0x12	Supported	Supported
Renegotiate Power	PICMG	0x13	Not supported	Supported
Get Fan Speed Properties	PICMG	0x14	Supported	Supported
Set Fan Level	PICMG	0x15	Supported	Supported
Get Fan Level	PICMG	0x16	Supported	Supported
Bused Resource	PICMG	0x17	Not supported	Supported
Get IPMB Link Info	PICMG	0x18	Supported	Supported
Get Shelf Power Allocation	PICMG	0x19	Supported	Supported
Get Shelf Manager IPMB Address	PICMG	0x1B	Supported	Supported
Set Fan Policy	PICMG	0x1C	Not supported	Not supported
Get Fan Policy	PICMG	0x1D	Not supported	Not supported

FRU Control Capabilities	PICMG	0x1E	Supported	Supported
FRU Inventory Device Lock Control	PICMG	0x1F	Supported	Supported
FRU Inventory Device Write	PICMG	0x20	Supported	Supported
Get Shelf Manager IP Addresses	PICMG	0x21	Supported	Supported

(*) This command is supported from the IPMB-0 side only if the configuration parameter ALLOW_ALL_COMMANDS_FROM_IPMB is set to TRUE.

The Shelf Manager defines several OEM-defined IPMI commands for the convenience of the System Manager. Currently defined commands are related to the following functional areas:

- TELCO Alarm control;
- Digital outputs control.

6.1. OEM Mechanisms for TELCO Alarm Control

These mechanisms provide an alternative for TELCO alarm control to the PEF-based mechanism. The mechanisms consist of one sensor and two OEM 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.

6.1.1. TELCO Alarm Sensor

This discrete sensor has sensor type DFh, event/reading type 6Fh (sensor-specific discrete) and is implemented on LUN 0 of the Shelf Manager IPM controller (20h). The following offsets are defined for the sensor:

0	Minor alarm active
1	Major alarm active
2	Critical alarm active
3	Alarm Cutoff active

6.1.2. OEM Command: Set/Clear TELCO Alarms

This OEM command is implemented by the Shelf Manager IPM controller (address 20h).

Network Function Code (NetFN): 3Eh

Command Code: D0h

	Byte	Data Field
Request Data	1	Bit mask of alarms to set: [7:3]: Reserved [2]: Set Critical Alarm [1]: Set Major Alarm [0]: Set Minor Alarm
	2	Bit mask of alarms to clear [7:3]: Reserved [2]: Clear Critical Alarm [1]: Clear Major Alarm [0]: Clear Minor Alarm
Response Data	1	Completion Code

6.1.3. OEM Command: Get TELCO Alarm Sensor Number

This OEM 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

	Byte	Data Field
Request Data	-	
Response Data	1	Completion Code
	2	The TELCO Alarm Sensor number on 20h

6.2. OEM Mechanisms for Controlling Digital Outputs

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 OEM commands are 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.

6.2.1. OEM Command: Query Digital Output Properties

This OEM command returns the number of available digital outputs.

Network Function Code (NetFN): 3Eh

Command Code: D4h

	Byte	Data Field
Request Data	1	FRU Device ID
Response Data	1	Completion Code
	2	The number of digital outputs, supported by this FRU

6.2.2. OEM Command: Get Digital Outputs

This OEM 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): 3Eh

Command Code: D3h

	Byte	Data Field
Request Data	1	FRU Device ID
	(2)	Starting group number; optional; defaults to 0
	(3)	Ending group number; optional; defaults to the last group
Response Data	1	Completion Code
	2	The state of digital outputs of the first requested group
	…	
	N	The state of digital outputs of the last requested group

6.2.3. OEM Command: Set/Clear Digital Outputs

This OEM 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): 3Eh

Command Code: D2h

	Byte	Data Field
Request Data	1	FRU Device ID
	2	The group number
	3	The bit mask of digital outputs to set within the group; bit 0 corresponds to the first digital output in the group; bit 7 corresponds to the last digital output in the group
	4	The bit mask of digital outputs to clear within the group; bit 0 corresponds to the first digital output in the group; bit 7 corresponds to the last digital output in the group
Response Data	1	Completion Code

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 the Shelf Manager functional operation is not affected and no switchover to the backup Shelf Manager is initiated.

On the other hand, if the physical IPM controller of the active Shelf Manager transitions to the state M1 as a result of opening the Hot Swap handle on the ShMC, 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.

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 reason for the implementation described above is that deactivation of the physical IPM controller of the active Shelf Manager can 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).

Chapter

A

A. Revision History

This section records the major revisions in this document, starting with release 2.1.0 of the Shelf Manager.

A.1 Release 2.1.0

- Section 2.2: The CLI commands 'gethysteresis', 'getipmbstate' are added to the table that summarizes the CLI commands.
- Section 3: The descriptions of the CLI commands 'gethysteresis', 'getipmbstate', 'sethysteresis', 'setipmbstate' are included.
- Section 4.2: The descriptions of Web interface for the CLI commands 'gethysteresis', 'sethysteresis' are included

A.2 Release 2.2.0

- Section 2.2: The table that summarizes the CLI commands identifies those that are available on the Backup Shelf Manager.
- Section 3: The descriptions of CLI commands 'getfruledstate', 'poll', 'setfruledstate', 'setpowerlevel' are included.
- Section 5.1.1: The access mode of the MIB variable 'board-basic-powered' is modified from 'read-only' to 'read-write'.

A.3 Release 2.3.0

- Overall: a change in Shelf Manager product name from 'IPM Sentry' to 'Pigeon Point' is implemented.
- Section 2.2: The table that summarizes the CLI commands shows additional commands available on the Backup Shelf Manager.

- Sections 3.2, 3.15: A new option ‘info’ is introduced for CLI commands ‘alarm’ and ‘frucontrol’.
- Section 3.25: 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 3.30: The CLI command ‘sendcmd’ is introduced.
- Section 3.32: The option ‘-t’ is introduced for 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 3.46.1: The option ‘-v’ (verbosity) is available for the CLI command ‘shelf power_management’.
- Section 4.2: The descriptions of Web interface for the CLI commands ‘alarm’, ‘getfruledstate’, ‘getipmbstate’, ‘getsensoreventenable’, ‘session’, ‘setfruledstate’, ‘setipmbstate’, ‘setsensoreventenable’ are included.
- Section 4.2.31: The Web interface for the CLI command ‘sel’ is updated.
- Section 5.1.4: The descriptions of the MIB variables ‘shelf-manager-active’ and ‘shelf-manager-reset’ are modified.
- Section 6: A table of IPMI commands implemented by the Shelf Manager is included. This table specifies whether a command is supported if it arrives from RMCP interface or from IPM controller
- Section 6.2.3: The command code for the OEM command Set/Clear Digital Outputs is corrected to 0xD2.
- New section 6.3: Deactivation scenarios for the active Shelf Manager are described.