



---

**System Management Interface based on  
BBS HPI-B 2. (Centellis 4440)**

User's Guide

P/N: 6806800P21G

August 2022

---



## Legal Disclaimer\*

SMART Embedded Computing, Inc. (SMART EC), dba Penguin Solutions™, assumes no responsibility for errors or omissions in these materials. **These materials are provided "AS IS" without warranty of any kind, either expressed or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.** SMART EC further does not warrant the accuracy or completeness of the information, text, graphics, links, or other items contained within these materials. SMART EC shall not be liable for any special, indirect, incidental, or consequential damages, including without limitation, lost revenues or lost profits, which may result from the use of these materials. SMART EC may make changes to these materials, or to the products described therein, at any time without notice. SMART EC makes no commitment to update the information contained within these materials.

Electronic versions of this material may be read online, downloaded for personal use, or referenced in another document as a URL to a SMART EC website. The text itself may not be published commercially in print or electronic form, edited, translated, or otherwise altered without the permission of SMART EC.

It is possible that this publication may contain reference to or information about SMART EC products, programming, or services that are not available in your country. Such references or information must not be construed to mean that SMART EC intends to announce such SMART EC products, programming, or services in your country.

## Limited and Restricted Rights Legend

If the documentation contained herein is supplied, directly or indirectly, to the U.S. Government, the following notice shall apply unless otherwise agreed to in writing by SMART EC.

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data clause at DFARS 252.227-7013 (Nov. 1995) and of the Rights in Noncommercial Computer Software and Documentation clause at DFARS 252.227-7014 (Jun. 1995).

### **SMART Embedded Computing, Inc., dba Penguin Solutions**

2900 S. Diablo Way, Suite 190

Tempe, Arizona 85282

USA

\*For full legal terms and conditions, visit <https://www.penguinsolutions.com/edge/legal/>

# Table of Contents

---

About this Manual .....	9
<b>1 System Management Interfaces Overview .....</b>	<b>13</b>
1.1 Introduction .....	13
1.2 Standard Compliances .....	13
1.3 Architecture .....	14
1.3.1 HPI Library .....	15
1.3.2 HPI Daemon .....	15
1.3.3 SNMP Agent .....	15
1.4 High Availability .....	17
<b>2 Software Installation and Configuration .....</b>	<b>19</b>
2.1 Overview .....	19
2.2 Installing and Configuring the HPI-B Daemon .....	20
2.2.1 Installing an HPI-B Daemon .....	20
2.2.2 Configuring an HPI Daemon .....	21
2.3 Setting Up HPI Clients .....	34
2.3.1 Installing HPI Clients .....	34
2.3.2 Configuring HPI Clients .....	35
2.4 Installing and Configuring an SNMP Agent .....	36
2.4.1 Installing an SNMP Agent .....	36
2.4.2 Configuring an SNMP Agent .....	37
<b>3 Developing Applications .....</b>	<b>39</b>
3.1 Overview .....	39
3.2 Building the Application .....	39
<b>4 Using HPI-B .....</b>	<b>41</b>
4.1 Overview .....	41
4.2 Limitations .....	41
4.2.1 Limitations with Respect to HPI-B Base Specification .....	41
4.2.2 Limitations with Respect to HPI-B AdvancedTCA Mapping Specification .....	42
4.3 Backward Compatibility .....	43
4.4 Working with the Multishelf Library .....	44

# Table of Contents

---

4.4.1	Overview	44
4.4.2	Accessing HPI Domains	45
4.4.3	How Domains and Shelves are Represented	45
4.4.3.1	Domain Management Resource	47
4.4.3.2	Shelf Management Resource	49
4.4.3.2.1	Domain Management Control	50
4.4.3.2.2	Connection State Sensor	50
4.4.3.3	HPI Domain Events	51
4.4.4	Discovering Shelves and Domains	51
4.4.5	Adding and Removing Shelves and Domains	52
4.4.5.1	Adding Shelves and Domains	52
4.4.5.2	Removing Shelves and Domains	53
4.5	Using the SNMP Agent	53
4.6	Penguin Edge Extensions	54
4.6.1	HPI Controls for Domain and Shelf Management	54
4.6.2	IPMI System Boot Options Support	54
4.6.3	POST Type Control	56
4.6.4	HPI Logging Support	57
4.6.5	Failed Slot Restore Control	59
4.6.6	HPI Restart Daemon Control	61
4.6.7	IPMI Command Control	62
4.6.8	Boot Bank Control	63
<b>5</b>	<b>Example Applications</b>	<b>65</b>
5.1	Overview	65
5.2	Example Application Source Files	65
5.3	List of Supported Example Applications	66
<b>6</b>	<b>Related Documentation</b>	<b>83</b>
6.1	Penguin Solutions Documentation	83
6.2	Related Specifications	83

# List of Tables

---

Table 1-1	HPI-B Standards Supported by Penguin Edge HPI-B Implementation	13
Table 2-1	RPM Files for HPI-B Clients and Daemons	19
Table 2-2	Overview of HPI-B Directories and Files on ATCA-F125/F140	20
Table 2-3	HPI Tags	22
Table 2-4	Plugin Specific Parameters	28
Table 2-5	Overview of HPI-B Directories and Files on Node Blades	34
Table 2-6	Multi-shelf Library Configuration File - HPI Daemon Entries	35
Table 2-7	IP Addresses/Ports of HPI Daemons in a Product Name Short Shelf	35
Table 2-8	SNMP Agent - Required Software Images	36
Table 2-9	Overview of SNMP Agent Related Directories and Files on ATCA-F125/F140	37
Table 3-1	Development RPMs - Directory Structure	39
Table 4-1	Backward Compatibility matrix	43
Table 4-2	Definition of Domain Management Resource	47
Table 4-3	Domain Management Control RDR	47
Table 4-4	Domain Management Control	48
Table 4-5	Domain Management Control State	48
Table 4-6	Shelf Management Resource	49
Table 4-7	Connection State RDR	50
Table 4-8	Domain Connection Sensor	50
Table 4-9	Domain Connection Sensor Reading	51
Table 4-10	Domain Connection Sensor Events	51
Table 4-11	Boot Option RDR	54
Table 4-12	Boot Option Control	55
Table 4-13	Boot Option State	55
Table 4-14	POST Type HPI Control RDR	56
Table 4-15	POST Type HPI Control State Values	57
Table 4-16	Log RDR	58
Table 4-17	Log Control	58
Table 4-18	Log Control State	58
Table 4-19	Failed Slot Restore Control RDR	60
Table 4-20	Failed Slot Restore Control States	60
Table 4-21	HPI Restart Daemon RDR	61
Table 4-22	HPI Restart Daemon State	61
Table 4-23	IPMI Command RDR	62
Table 4-24	IPMI Command Control	62
Table 4-25	IPMI Command State	63
Table 4-26	Boot Bank RDR	63

## List of Tables

---

Table 4-27	Boot Bank Control .....	63
Table 4-28	Boot Bank State .....	64
Table 6-1	Penguin Solutions Documentation .....	83
Table 6-2	Related Specifications .....	83

# List of Figures

---

Figure 1-1	Overview of HPI Usage in Systems . . . . .	14
Figure 1-2	SNMP Agent Highlevel Software Architecture . . . . .	16
Figure 4-1	Multishelf Library - Representation of Shelves as Domains . . . . .	44
Figure 4-2	HPI Multishelf Library - Overview of Related HPI Resources and Controls . . . . .	46

## List of Figures

---

# About this Manual

---

## Overview of Contents

This manual is divided into the following chapters and appendices.

- [Chapter 1, System Management Interfaces Overview on page 13](#) provides an overview on HPI-B in Centellis 4440 AdvancedTCA systems.
- [Chapter 2, Software Installation and Configuration on page 19](#) describes how to install and configure HPI-B clients and HPI-B daemons.
- [Chapter 3, Developing Applications on page 39](#) describes the necessary steps in order to build HPI-B client applications.
- [Chapter 4, Using HPI-B on page 41](#) describes in detail which HPI-B features are supported.
- [Appendix 5, Example Applications on page 65](#) briefly describes HPI-B example applications, which are delivered as part of the Centellis 4440 HPI-B distribution.
- [Appendix 6, Related Documentation on page 83](#) provides references to other, related documentation.

## Abbreviations

This document uses the following abbreviations:

Abbreviation	Description
AMC	Advanced Mezzanine Module
ATCA	Advanced Telecom Computing Architecture
BT	Block Transfer
CGE	Carrier Grade Edition
CPIO	Copy In/Out
CPU	Central Processing Unit
FRU	Field Replaceable Unit
HPI	Hardware Platform Interface
IA	Intel Architecture
ID	Identifier
IP	Internet Protocol
IPMI	Intelligent Platform Management Interface
LAN	Local Area Network

## About this Manual

---

Abbreviation	Description
MVL	Monta Vista Linux
OEM	Original Equipment Manufacturer
PICMG	PCI Industrial Computer Manufacturers Group
RMCP	Remote Management Control Protocol
RDR	Resource Data Record
RPM	RedHat Package Manager
RPT	Resource Presence Table
SAF	Service Availability Forum
SAI	Service Availability Interface
SAIM	Service Availability Interface Mapping
SAM	Shelf Management Alarm Module
SMI	Serial Management Interface
ShMC	Shelf Management Controller

## Conventions

The following table describes the conventions used throughout this manual..

Notation	Description
0x00000000	Typical notation for hexadecimal numbers (digits are 0 through F), for example used for addresses and offsets
0b0000	Same for binary numbers (digits are 0 and 1)
<b>bold</b>	Used to emphasize a word
Screen	Used for on-screen output and code related elements or commands. Sample of Programming used in a table (9pt)
<b>Courier + Bold</b>	Used to characterize user input and to separate it from system output
<i>Reference</i>	Used for references and for table and figure descriptions
File > Exit	Notation for selecting a submenu
<text>	Notation for variables and keys
[text]	Notation for software buttons to click on the screen and parameter description
...	Repeated item for example node 1, node 2, ..., node 12

Notation	Description
.	Omission of information from example/command that is not necessary at the time
..	Ranges, for example: 0..4 means one of the integers 0,1,2,3, and 4 (used in registers)
	Logical OR
	Indicates a hazardous situation which, if not avoided, could result in death or serious injury
	Indicates a hazardous situation which, if not avoided, may result in minor or moderate injury
	Indicates a property damage message
	Indicates a hot surface that could result in moderate or serious injury
	Indicates an electrical situation that could result in moderate injury or death
<p data-bbox="275 1234 382 1286"><b>Use ESD protection</b></p> 	Indicates that when working in an ESD environment care should be taken to use proper ESD practices
	No danger encountered, pay attention to important information

## About this Manual

---

# Summary of Changes

This manual has been revised and replaces all prior editions.

Part Number	Publication Date	Description
6806800P21F	August 2022	Rebrand to Penguin Solutions
6806800P21F	November 2019	Rebrand to SMART Embedded Computing. Removed references to Centellis 4620.
6806800P21E	December 2017	Updated copyrights section.
6806800P21D	June 2014	Rebranded to Artesyn.
6806800P21C	January 2014	Updated <a href="#">Table 2-3 on page 22</a> .
6806800P21B	June 2012	Added the following tables: <a href="#">HPI Tags on page 22</a> ; <a href="#">Plugin Specific Parameters on page 28</a> ; <a href="#">HPI Restart Daemon RDR on page 61</a> ; <a href="#">HPI Restart Daemon State on page 61</a> ; <a href="#">IPMI Command RDR on page 62</a> ; <a href="#">IPMI Command Control on page 62</a> ; <a href="#">IPMI Command State on page 63</a> ; <a href="#">Boot Bank RDR on page 63</a> ; <a href="#">Boot Bank Control on page 63</a> ; <a href="#">Boot Bank State on page 64</a> .  Updated <a href="#">Limitations with Respect to HPI-B AdvancedTCA Mapping Specification on page 42</a> .
6806800P21A	April 2012	Initial release

# System Management Interfaces Overview

## 1.1 Introduction

Penguin Solutions provides an SAF Hardware Platform Interface (HPI) as part of its AdvancedTCA platforms. HPI provides an industry standard interface to monitor and control highly available telecommunications system platforms. The ability to monitor and control these platforms is provided through a consistent and standard set of programmatic interfaces that are targeted for adoption by the telecom building block industry to significantly reduce product time-to-market and development costs while retaining or enhancing total system/network availability.

HPI provides the interface between the middleware software solution stack and the hardware solution stack, allowing portability of middleware software building blocks across many different hardware platforms and portability of hardware platforms across many different middleware software building blocks.

This guide describes the HPI-B implementation targeted at the Penguin Edge AdvancedTCA Product Name Short platform.

## 1.2 Standard Compliances

The Penguin Edge HPI-B implementation for the Product Name Short environment is compliant to the following standards.

*Table 1-1 HPI-B Standards Supported by Penguin Edge HPI-B Implementation*

Standard	Description
SAI-HPI-B.03.02	HPI-B base specification. It abstracts hardware platform characteristics into a data model consisting of entities and resources.
SAI-HPI-SNMP-B.01.01	HPI-B SNMP interface specification.
SAIM-HPI-B.03.02-xTCA	HPI-B-AdvancedTCA mapping specification. It provides a vendor independent hardware platform view of an AdvancedTCA system.

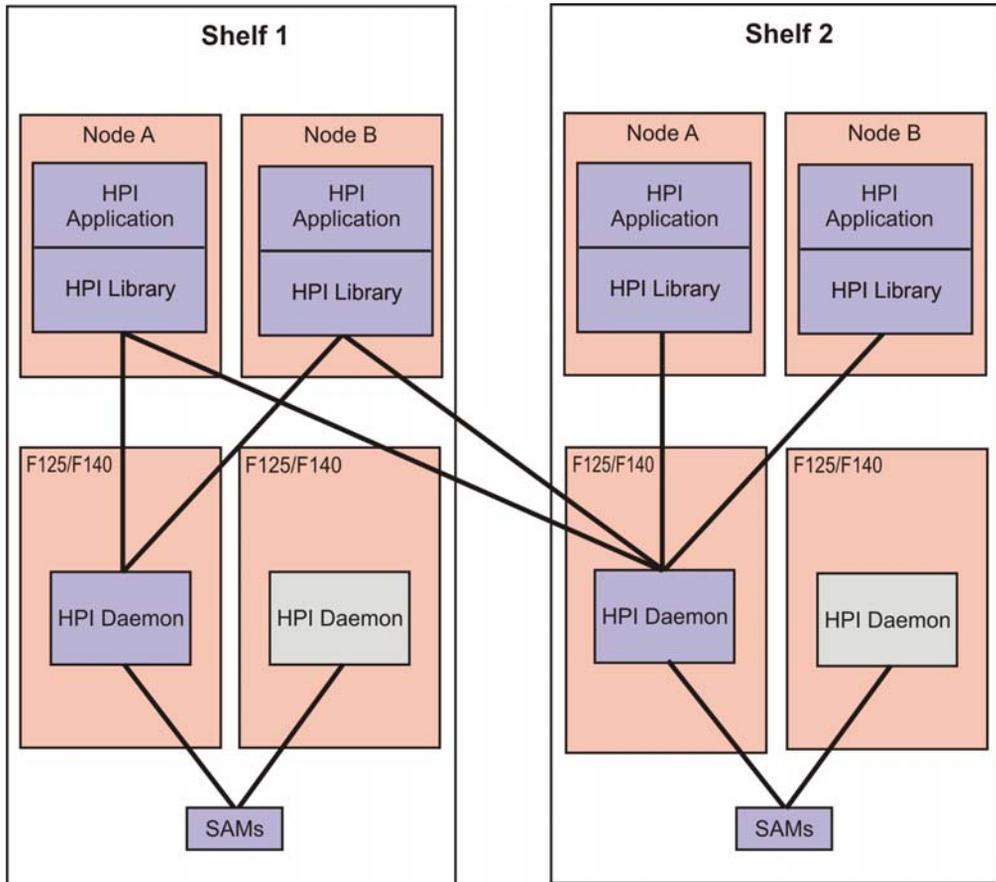
Although the standards listed above are fulfilled, some specific limitations apply. For details, refer to [Limitations on page 41](#).

### 1.3 Architecture

The Penguin Edge HPI implementation is provided in the form of a client-server architecture. In the Product Name Short system environment the server, called HPI daemon, runs on the ATCA-F125/F140 switch blade, and the client, which is constituted by an HPI library and an application which links that library, runs on any node within a shelf. Internally each HPI daemon is connected to a Shelf Management Alarm module (SAM) in the shelf. The communication between HPI daemon and SAM is realized via RMCP (Remote Management Control Protocol) which is an IP-encapsulation of IPMI commands. The communication between HPI daemon and HPI client is realized by an IP-based remote HPI communication protocol.

An HPI client may access one or more HPI daemons, and on the other hand, an HPI daemon may be accessed by one or more HPI clients. The following figure illustrates this.

Figure 1-1 Overview of HPI Usage in Systems



### 1.3.1 HPI Library

The HPI library is the primary user interface. It is intended to be used by applications that wish to control and monitor HPI managed components, such as ATCA shelves, blades and other FRUs. The HPI library is delivered as shared and as static library and has to be linked with an application.

There are two types of HPI libraries available:

- Single shelf library
- Multi shelf library

The single shelf library supports the communication with one HPI daemon only. It is only used internally, for example by the SNMP subagent and is not intended to be used by user applications. The multishelf library, on the other hand, supports the communication with one up to several HPI daemons. This becomes necessary if you wish to deploy redundancy in one HPI-B based shelf management system or if you wish to manage several shelves. The multishelf library is the library which you should build your applications on, it is the official interface to customer applications.

Details about supported combinations of CPU architecture/Linux distribution are given in [Chapter 2, \*Software Installation and Configuration\*, on page 19](#).

### 1.3.2 HPI Daemon

The HPI daemon within an Product Name Short system runs on the ATCA-F125/F140 blades and its main tasks are:

- Provide a single access point to control and monitor hardware components in a shelf
- Map information provided by the underlying Shelf Manager to HPI

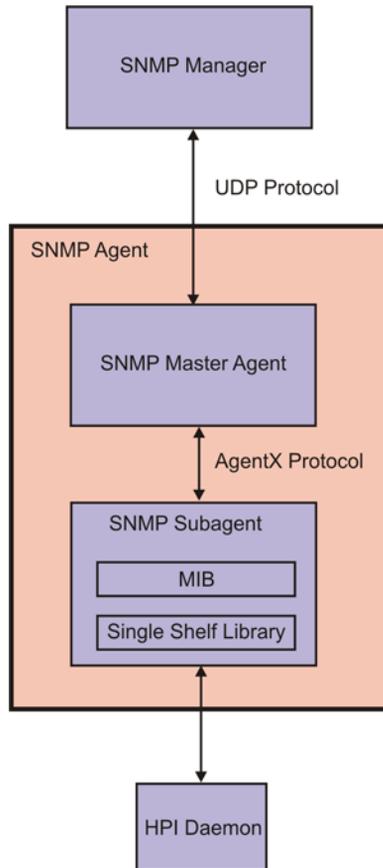
### 1.3.3 SNMP Agent

The Penguin Solutions HPI-B distribution includes an SNMP agent. It provides a management interface which allows you to remotely control and manage hardware components within the Product Name Short shelf. The supported Management Interface Base (MIB) is the standard SAI-HPI-SNMP-B.01.01.mib as defined by the Service Availability Forum.

## System Management Interfaces Overview

The SNMP agent consists of an SNMP master agent and an SNMP subagent. The SNMP master agent is based on Net-SNMP and handles the communication with an external SNMP manager. Internally, the SNMP master agent communicates with the SNMP subagent via the AgentX protocol. The SNMP subagent uses the HPI-B single shelf library and communicates with an HPI-B daemon. The following figure illustrates the software architecture of the SNMP agent.

Figure 1-2 SNMP Agent Highlevel Software Architecture



The SNMP agent software is intended to run on an ATCA-F125/F140 blade and should be preinstalled together with the remaining ATCA-F125/F140 software. If you need to reinstall or update the SNMP agent software, refer to [Installing an SNMP Agent on page 31](#). For details about how to configure the SNMP agent, refer to [Configuring an SNMP Agent on page 33](#).

### 1.4 High Availability

The Penguin Edge HPI-B implementation described in this manual supports the following two redundancy options:

- Active/active
- Cold-standby

The active/active option is the default configuration. Both HPI daemons in shelf are active and run simultaneously. Your application connects to one daemon and if the connection fails it connects to the second daemon. Note that the current HPI-B implementation does not replicate any data between the two daemons, this means data consistency is not guaranteed.

Alternatively you may choose to use the cold-standby redundancy option. In this configuration your application must make sure that only one HPI daemon is active at a time. If the daemon fails, your application starts up the second, previously inactive daemon and connects to it. During start-up the HPI daemon scans the current system environment. This way it is ensured that the daemon reflects the current system configuration.

## **System Management Interfaces Overview**

---

# Software Installation and Configuration

## 2.1 Overview

This section describes how to set-up HPI-B daemons and clients and how to install all files needed to run client applications. Generally, all files are delivered in the form of RPMs. The content of an RPM is reflected in its naming scheme.

The following table describes the used naming scheme of the client and daemon RPMs. The placeholder "architecture" stands for the supported CPU architecture of the respective blade where the client or daemon is to run and the following boards or architectures should be referred as example:

- x86  
Intel IA-32 bit blades, such as PrAMC-7210/7211
- x86\_64  
Intel IA-64-bit blades, such as the ATCA-7221
- ppc\_e500v2  
PowerPC based blades, such as the ATCA-F125/F140

*Table 2-1 RPM Files for HPI-B Clients and Daemons*

RPM File Name	Description
bbs-hpib-<version>- 1.<architecture>- <distribution>-<OS>.rpm	This RPM is the HPI-B base package. It contains shared libraries to be used by HPI-B clients and daemons, as well as compiled example applications and client configuration files. This package is required both by HPI-B daemons and clients.
bbs-hpib-daemon-<version>- 1.<architecture>- <distribution>-<os>.rpm.	This RPM contain all files which are related to the HPI-B daemon: executables, libraries and configuration file.

The files required for the HPI-B daemon come as part of the ATCA-F125/F140 software. Depending on the particular ATCA-F125/F140 release, the HPI-B daemon files are already preinstalled or not. Check the respective documentation of the ATCA-F125/F140 itself and of the system where the ATCA-F125/F140 is used.

If the HPI-B software is not preinstalled or if you want to upgrade the installed HPI-B software, then you can obtain the daemon files as an RPM file. The file can be obtained from Penguin Edge.

# 2.2 Installing and Configuring the HPI-B Daemon

This section describes how to install and configure an HPI-B daemon.

## 2.2.1 Installing an HPI-B Daemon

As previously mentioned, it depends on the ATCA-F125/F140 release if the HPI-B daemon files are preinstalled or not. If they are not preinstalled or if you want to upgrade existing HPI-B files, you need to obtain the desired RPM files and install them manually as follows.

### Installing the HPI-B Daemon Files

In order to install/upgrade the HPI-B daemon files on an ATCA-F125/F140 blade, proceed as follows.

1. Connect to the ATCA-F125/F140 blade where you wish to install the HPI-B daemon files.
2. Copy the RPM file to the ATCA-F125/F140.
3. If applicable, enter `rpm -e <Old HPI-B daemon package name>` to uninstall existing daemon files
4. If applicable, enter `rpm -e <Old HPI-B client base package name>` to uninstall an installed HPI-B client base package
5. Enter `rpm -i <New HPI-B client base package RPM>`  
This installs the new HPI-B client base package files.
6. Enter `rpm -i <New HPI-B daemon RPM>`  
This installs the new daemon files.

The following table lists all HPI-B related directories available on the ATCA-F125/F140 after the installation.

Table 2-2 Overview of HPI-B Directories and Files on ATCA-F125/F140

Directory	Description
<code>/opt/bladeservices/bin</code>	Contains HPI-B daemon binaries and compiled client example applications.
<code>/opt/bladeservices/etc/bbs-hpib</code>	Contains configuration files used to configure the HPI-B daemon and clients. See <a href="#">Configuring an HPI Daemon on page 21</a> for details on configuring the HPI daemon.
<code>/etc/init.d</code>	Daemon start/stop script
<code>/opt/bladeservices/lib</code>	Shared libraries

### 2.2.2 Configuring an HPI Daemon

At start-up, the HPI daemon reads the following configuration file:

```
/opt/bladeservices/etc/bbs-hpib/bbs-hpib.conf.
```

The only entry which you must configure manually once is the IP address and port used to access the SAMs in the Product Name Short. Both SAMs have a virtual IP address assigned to them which is to be used specifically for RMCP-based accesses. This is the IP address which you must specify in the configuration file.

The corresponding section in the configuration file is called `ipmidirect`. Within `ipmidirect`, you need to adapt the entries `addr` and `port`, where `addr` is the IP address and `port` is the port.

In a Product Name Short shelf the virtual IP address used to access the SAMs via RMCP is 192.168.24.11 and the port number is 623, by default. Thus, an entry in your configuration file should look as follows:

```
handler ipmidirect {
    entity_root = "{ADVANCEDTCA_CHASSIS,0}"
    name = "lan"          # RMCP
    addr = "192.168.24.11"
    port = "623"         # RMCP port

```

...

Both SAMs are operated in an active/stand-by mode. An internal redundancy mechanism ensures that the HPI-B daemon is always connected to the currently active SAM. Since a virtual IP address is used, this is transparent to the HPI-B daemons and HPI-B client applications.

## Software Installation and Configuration

Table 2-3 HPI Tags

Tag	Description	Allowed values	Default in Config. file	Built-in Default
OPENHPI_INTERACTIVE	interactive daemon with command line interface	"YES   NO"	[NO] - non-interactive YES - interactive	NO
OPENHPI_CONFIG_REPLICATION	replicate this configuration file to standby	"YES   NO"	[NO] - no replication YES - replicate	NO
OPENHPI_DEL_SIZE_LIMIT	sets the maximum size for the domain event log	<0...N>	[1000] - max size is 1000 0 - unlimited	10000
OPENHPI_ON_EP	Sets the entity path on which the application is running. This entity path will be returned when SaHpiResourceIdGet () is called	" {ADVANCEDTCA_CHASSIS, <shelf_id> }"	[ {ADVANCEDTCA_CHASSIS, 0} ]	{ADVANCEDTCA_CHASSIS, 0}
OPENHPI_EVT_QUEUE_LIMIT	sets the maximum number of events that are allowed in the session's event queue	<0...N>	[1000] - max size is 1000 0 - unlimited	10000
OPENHPI_DEL_SAVE	sets whether the domain event log will be persisted to disk or not. The event log is written to OPENHPI_VARPATH value	"YES   NO"	[NO] - not saved [YES] - saved to OPENHPI_VARPATH	NO
OPENHPI_DAT_SIZE_LIMIT	Sets the maximum size for the alarm table	<0...N>	[1000] - max size is 1000 0 - unlimited	1000

Table 2-3 HPI Tags (continued)

Tag	Description	Allowed values	Default in Config. file	Built-in Default
OPENHPI_DAT_USER_LIMIT	sets the maximum number of user type alarm entries allowed in the alarm table	<0...N>	[100] - max size is 100	100
OPENHPI_LOG_ON_SEV	sets the lowest severity level an event must meet to be logged in the domain event log	(Order: Highest to Lowest) "CRITICAL   MAJOR   MINOR   INFORMATIONAL   OK   DEBUG"	[MINOR] - all events with MINOR and above severity are logged  CRITICAL   MAJOR   INFORMATIONAL   OK   DEBUG - Events with given severity and above that are logged.	MINOR
OPENHPI_AUTO_INSERT_TIMEOUT	Sets the default auto-insert timeout. If the resource is not shelf manager auto activated, it starts an auto insert timer, otherwise the resource stays in insertion state until user explicitly issues saHpiResourceActiveSet API. (This information is taken from PICMG Fru Activation and Power Descriptor record)	"block   <0...N>"	[block] - auto insertion policy never executed until user issues saHpiResourceActiveSet ()  <0...N> Any positive value would give timeout in nanoseconds.	block

## Software Installation and Configuration

Table 2-3 HPI Tags (continued)

Tag	Description	Allowed values	Default in Config. file	Built-in Default
OPENHPI_AUTO_EXTRACT_TIMEOUT	Sets the default auto-extract timeout. If the resource is not shelf manager auto deactivated, it starts an auto extract timer, otherwise the resource stays in extraction state until user explicitly issues saHpiResourceInactiveSet API (This information is taken from PICMG Fru Activation and Power Descriptor record)	"block   <0...N>"	[60000000000] - 60000 seconds (in microseconds) block - auto extraction policy never executed until user issues saHpiResourceInactiveSet()	block
OPENHPI_PATH	Search path for plugins	Any valid search path	[/opt/bladeservices/lib/bbs-hpib]	/opt/bladeservices/lib/bbs-hpib
OPENHPI_VARPATH	path to store user events	Any valid path	[/var/lib/bbs-hpib] - If OPENHPI_DEL_SAVE is YES, then user events are stored in this path by default	/var/lib/bbs-hpib
OPENHPI_CONNECTION_PING_INTERVAL	Connection ping interval for the clients	<0...N>	[500] - 500 milliseconds is the default ping interval	1000 ms
OPENHPI_CONNECTION_PING_TIMEOUT	Connection timeout for clients	<0...N>	[5000] - 5000 milliseconds is the default timeout	5000

Table 2-3 HPI Tags (continued)

Tag	Description	Allowed values	Default in Config. file	Built-in Default
OPENHPID_MAX_NUMBER_OF_CONNECTIONS	maximum number of HPI clients	<0...N>	[ 30 ] - when the client connections reach 30, any further connections are rejected	30
OPENHPI_START_SNMP	Start SNMP after HPI-B daemon discovery	YES   NO	[ No ] - snmp is not started [ YES ] - snmp is started after HPI-B discovery	NO
OPENHPI_SNMP_SCRIPT	SNMP script to run if OPENHPI_START_SNMP is yes	<Path to snmp script>	[ /etc/init.d/hpibsnmp ] - Default snmp script to be executed	/etc/init.d/hpibsnmp
OPENHPI_DAEMON_PORT	Openhpi daemon port	port number	[ 4743 ] - default port for HPI-B daemon	4743
OPENHPI_LOGFILE	Path and Name of the log file	Any valid name	[ /var/log/bbs-hpib ] - log file name starts with bbs-hpib Ex: bbs-hpib00.log	/var/log/bbs-hpib
OPENHPI_LOGFILE_MAX	Sets the maximum number of log files to be created	<0...N>	[ 2 ] - By default, 3 log files are created	2
OPENHPI_LOGFILE_MAX_SIZE	Log file maximum size	<0...N>	[ 500000 ] - 500 KB is default maximum size	500000

## Software Installation and Configuration

Table 2-3 HPI Tags (continued)

Tag	Description	Allowed values	Default in Config. file	Built-in Default
OPENHPI_LOGFLAGS	Flags that control logging	"stdout   stderr   file   simplefile   syslog   prefix   threadid"	[file prefix] - log into a file with prefix(<time> [threadid] <log level> {facility}) in each line logged  Stdout - log to standard output  Stderr - standard error  Simplefile - only one file for logging and its name is OPENHPI_LOGFILE  Syslog - log into syslog  Threadid - logging of threadid is optional. With this option, threadid is also logged.	stderr
OPENHPI_LOG_ERROR_FACILITIES	Facilities that should be logged at error level	List of facilities mentioned in the config file	[all] - By default, all the facilities are logged at error level  To mention more than one facility, use    Ex: core daemon	all
OPENHPI_LOG_WARNING_FACILITIES	Facilities that should be logged at warning level	List of facilities mentioned in the config file	[all] - By default, all the facilities are logged at warning level	all
OPENHPI_LOG_INFO_FACILITIES	Facilities that should be logged at info level	List of facilities mentioned in the config file	[all] - By default, all the facilities are logged at info level	all

Table 2-3 HPI Tags (continued)

Tag	Description	Allowed values	Default in Config. file	Built-in Default
OPENHPI_LOG_DEBUG_FACILITIES	Facilities that should be logged at info level	List of facilities mentioned in the config file	[ ] - By default, no facilities are logged at debug level User has two options to change facilities at any log level - 1.Modifying config file - the changes are persistent and are reflected after restarting the hpib daemon 2.Using hpilog application - changes take effect immediately but are not persistent and removed once the daemon is restarted	" "
OPENHPI_LOGTHREAD	Separate thread for logging	"0   1"	[1] - start a separate thread for logging [0] - no separate thread If tag is not mentioned, no thread is started	0
OPENHPI_CRASHFILE	Crash handler log file	Any valid path	[/var/log/bbs-hpibdcrash.log] - default file into which stack trace of all the running threads is dumped when a crash occurs	/var/log/bbs-hpibdcrash.log
OPENHPI_HA_IP	IP address for redundant hpi-b daemon	HA IP	[ ] - empty	0.0.0.0
OPENHPI_HA_HWADDR_OFFSET	Hardware address of the first shelf manager	<65>	[65] - default value	8

## Software Installation and Configuration

Table 2-3 HPI Tags (continued)

Tag	Description	Allowed values	Default in Config. file	Built-in Default
OPENHPI_HA_NUM_PORTS	Number of ports to open per IP	<1...N>	[1] - only 1 port opened	5
OPENHPI_LOG_SEL_LOGS_IN_SYSLOG	Logging SEL logs into syslog in the same format as in Shelf Manager's SEL	"true   false"	[false] - not logged true - logged into syslog	false

Table 2-4 Plugin Specific Parameters

Tag	Description	Allowed values	Default in Config. file	Built-in Default
entity_root	Root entity value	{<chassis description> , <0...N>}	[ {ADVANCEDTCA_CHASSIS, 0} ] - Any entity's path would be starting with the above string	Can be evaluated in 2 ways 1. Uses shelf address as the integer 2. If step 1 fails, then uses 0
name	Interface name for ipmi connection	"smi   lan"	[lan] - connection over LAN smi- connection over SMI interface	"smi" - if HPI is running on shelf manager "lan" - if HPI is running on management controllers other than shelf manager

Table 2-4 Plugin Specific Parameters (continued)

Tag	Description	Allowed values	Default in Config. file	Built-in Default
addr	Shelf manager RMCP address	IP address	[192.168.24.11]	NA
port	RMCP port	Port number	[623]	NA
auth_type	RMCP authentication type	"none   md2   md5   straight"	[none]	none
auth_level	RMCP authentication level	"admin   operator"	[admin]	admin
username	RMCP username	Username	[ " " ]	" "
password	RMCP password	Password	[ " " ]	" "
DomainTag	Domain Tag	Domain tag	[ "ATCA Shelf" ]	NA
sel_filename	System event log file name	Any valid file name	[/var/lib/bbs-hpib/shelf_sel.db] Commented out by default	NA
max_number_of_sel_entries	Maximum number of SEL entries	<0...N>	[100] Commented out by default	NA
IpmiConnectionTimeout	IPMI Connection timeout	<0...N> msec	[1500] - 1500 milliseconds	NA
AtcaConnectionTimeout	IPMI driver/smi timeout	<0...N> msec	[1500] - 1500 milliseconds	1000
MaxOutstanding	Maximum number of outstanding IPMI commands	<0...N>	[10] - 10 commands by default. After this max is reached, any other request is awaited.	0

## Software Installation and Configuration

Table 2-4 Plugin Specific Parameters (continued)

Tag	Description	Allowed values	Default in Config. file	Built-in Default
ResourcePowerOffTimeout	A timeout after which the resource FRU Power State will be set to initial state. This timer is started soon after executing the Power Off/Power Cycle/hotswap operations on the resource.	<0...N>	[60000] - 60000 ms	60000
ResourcePowerOnTimeout	A timeout after which the resource FRU Power State will be set to initial state. This timer is started soon after executing the Power On/hotswap operations on the resource.	<0...N>	[20000] - 20000 ms	20000
unsolicited_events	Subscription for asynchronous event notification from the shelf manager	"yes   no"	[yes] - Subscribe for "AsyncEventNotification" from the Shelf Manager no - Do not subscribe for "AsyncEventNotification" from the Shelf Manager	no
clear_sel	Clear the SEL after reading events from it	"yes   no"	[no] - Do not clear the SEL yes - clear the SEL	no

Table 2-4 Plugin Specific Parameters (continued)

Tag	Description	Allowed values	Default in Config. file	Built-in Default
RmcpEnable	Enables/disables the RMCP server	"true   false"	[false] - disables the rmcp server true - enables the rmcp server	false
RmcpPort	Port on which RMCP service runs	Standard port	[623]	623
RmcpMaxConnection	Maximum number of connections active. After this count is reached, any new connections are rejected	<0...N>	[11] - 11 connections	10
RmcpConnectionTimeout	RMCP connection timeout	<0...N>	[60000] - 60000 msec	60000 msec
RmcpSessionTimeout	RMCP Session Timeout	<0...N>	[60000] - 60000 msec	60000 msec
RmcpPefAlerting	RMCP PEF alerting	"true   false"	[true] - supports PEF alerting false - does not support	false
RmcpPerMsgAuth	RMCP per msg authentication	"true   false"	[true] - authentication supported	false
RmcpPrivLimit	Privilege needed for using RMCP service	"callback   user   operator   admin"	[admin] - highest privilege	admin
RmcpAllowedAuthsCallback	Allowed authentication mechanisms for privilege level 'callback'	"none   md2   md5   straight"	"[none, md2, md5, straight]"	none

## Software Installation and Configuration

Table 2-4 Plugin Specific Parameters (continued)

Tag	Description	Allowed values	Default in Config. file	Built-in Default
RmcpAllowedAuthsUser	Allowed authentication mechanisms for privilege level 'user	"none   md2   md5   straight"	"[none, md2, md5, straight]"	none
RmcpAllowedAuthsOperator	Allowed authentication mechanisms for privilege level 'operator	"none   md2   md5   straight"	"[none, md2, md5, straight]"	none
RmcpAllowedAuthsAdmin	Allowed authentication mechanisms for privilege level 'operator	"none   md2   md5   straight"	"[none, md2, md5, straight]"	none
RmcpGuid	RMCP group user id	16 hex bytes	[01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16] - default guid	All zeros (16 bytes)
RmcpUser[X] X is an integer Ex: RmcpUser2, RmcpUser3, RmcpUser4 etc	RMCP users	<U><P><Pr><MS><A> U - username P - Password Pr - Privilege MS - Maximum number of sessions A - Authentication	[RmcpUser[2,3,4,5,6]] are available in config file Ex: RmcpUser6 = "Administrator Administrator admin 64 none md2 md5 straight"	NA
SelMaxEntries	Maximum number of entries in the SEL	<0..N>	[100]	1024

Table 2-4 Plugin Specific Parameters (continued)

Tag	Description	Allowed values	Default in Config. file	Built-in Default
<p>HPI_OEM_DISCRETE_SENSOR_DEF_SEVERITY</p>	<p>HPI-B Daemon will set the severity for the HPI events of non-threshold sensors based on the value configured for this tag.</p> <p>Note: If this tag is set to 3, then change the OPENHPI_LOG_ON_SEV tag to "INFORMATIONAL" instead of "MINOR" so that it logs events of "INFORMATIONAL" severity also into DEL.</p>	<p>&lt;0, 1, 2, 3&gt;</p>	<p>[ 3 ] - informational            0 - critical            1 - major            2 - minor</p> <p>Commented by default,            Please uncomment in the config file for the tag to take effect.</p>	<p>0 - critical            daemon will assign the default "critical" severity            for the events of non-threshold sensors(discrete/OEM)</p>

### 2.3 Setting Up HPI Clients

This section describes how to install/configure HPI clients on node blades.

#### 2.3.1 Installing HPI Clients

##### Procedure

In order to install/upgrade an HPI-B client package on a node blade, proceed as follows.

1. Connect to the node blade where you wish to install the HPI-B client package.
2. Copy the RPM file that you wish to install to the node blade. Refer to [Table 2-1 on page 19](#) for details on available RPM files for your particular node blade.
3. If applicable, enter `rpm -e <Old HPI-B client package name>` to uninstall an already installed client package.
4. Enter `rpm -i <New HPI-B client RPM file name>`  
This installs the HPI-B library package.

The following table lists the directories and their content available on the blade after installing the package.

*Table 2-5 Overview of HPI-B Directories and Files on Node Blades*

Directory	Description
<code>/opt/bladeservices/lib</code>	Contains example applications and shared libraries needed to run clients
<code>/opt/bladeservices/bin</code>	Contains precompiled example applications. They are controlled via the command line and can easily be identified through the prefix "hpi" in their names. Use the -h option to display supported command line parameters.
<code>/opt/bladeservices/etc/bbs-hpib</code>	Contains configuration files used to configure HPI client libraries. See for details.

## 2.3.2 Configuring HPI Clients

Before running your client, you need to configure the multishelf library on the node where the client is to run. The configuration has to be done in the following configuration file:

`/opt/bladeservices/etc/bbs-hpib/bbs-hpibmultishelf.conf`



**The multishelf library is intended to be used by your user applications. Some HPI-B example applications and also the SNMP subagent, however, use the single shelf library instead. The single shelf HPI-B library is configured via the following configuration file: `/opt/bladeservices/etc/bbs-hpib/bbs-bbs-hpibclient.conf`. The following descriptions apply to the single shelf library as well, except that only one IP address of an HPI daemon must be specified, as the single shelf library supports the communication with one HPI daemon only.**

Most of the entries should be left as they are. They have been set to values that are appropriate for most operations. The only settings that need to be adapted are those which are related to the HPI daemons that the multi-shelf library wishes to access. The following table shows the expected syntax of the related entries.

Table 2-6 Multi-shelf Library Configuration File - HPI Daemon Entries

Entry	Description
<code>[Shelf&lt;Domain Name&gt;]</code>	This indicates the start of the definition of an HPI daemon. The chosen domain name appears as name of the Shelf Management Resource and is used as Domain tag. See <a href="#">Shelf Management Resource on page 49</a> .
<code>Daemon=&lt;IP address of HPI daemon&gt;</code>	This is the IP address used to access an HPI daemon.
<code>Port=&lt;port number&gt;</code>	This is the port number. The HPI daemon uses 4743 as port.

Typically you will want to specify both HPI daemons in the shelf in the configuration file so that your application can establish a connection with the second HPI daemon in case the connection with the first HPI daemon fails.

The following table lists the IP addresses and ports of the HPI daemons in a Product Name Short shelf.

Table 2-7 IP Addresses/Ports of HPI Daemons in a Product Name Short Shelf

Location of HPI Daemon	IP Address	Port
Left ATCA-F125/F140 in a shelf	192.168.21.1	4743
Right ATCA-F125/F140 in a shelf	192.168.22.2	4743

# 2.4 Installing and Configuring an SNMP Agent

The SNMP agent is intended to run on the ATCA-F125/F140 and is bundled with the ATCA-F125/F140 Basic Blade Services (BBS) software. This means that under normal conditions there is no need to install the SNMP software manually. If for some reason you need to manually install an SNMP agent on an ATCA-F125/F140, follow the installation instructions given below.

You may want to change the default configuration of an SNMP agent. All necessary steps are described below as well.

For more information on using SNMP agent, refer to *HPI-B Subagent User Guide*.

## 2.4.1 Installing an SNMP Agent

### Procedure

The SNMP agent requires the following software images.

Table 2-8 SNMP Agent - Required Software Images

Software Image	Description
SNMP subagent package	This is an RPM with the following file name scheme: <code>bbs-hpib-snmp-&lt;version&gt;-1.&lt;architecture&gt;-&lt;distribution&gt;-&lt;OS&gt;.rpm</code> It can be obtained from Penguin Edge and contains SNMP agent binaries, config files as well as MIB files.
net-snmp 5.1.2 or higher	This is an open source suite which contains files needed by the SNMP master agent as well as useful SNMP command line tools. It is part of the ATCA-F125/F140 BBS software and therefore there should be no need to install it manually.
HPI-B client package	Since the SNMP subagent is an HPI-B client application, it relies on the HPI-B client package to be installed. For details about how to install and configure an HPI-B client package, refer to <a href="#">Setting Up HPI Clients on page 34</a> .

Make sure that the net-snmp 5.1.2 or higher package as well as the HPI-B client base package are installed on the ATCA-F125/F140 where you wish to install the SNMP agent. In order to install the SNMP subagent package, proceed as follows:

1. Connect to the ATCA-F125/F140 blade where you wish to install the HPI-B SNMP subagent package.
2. Copy the package file to the ATCA-F125/F140.

3. If applicable, enter `rpm -e <Old HPI-B SNMP subagent package name>` to uninstall an existing SNMP subagent package
4. Enter `rpm -i <New HPI-B SNMP subagent package name>` to install the new package.

After the installation of all required packages, the SNMP agent related files can be found in the following directories on the ATCA-F125/F140:

*Table 2-9 Overview of SNMP Agent Related Directories and Files on ATCA-F125/F140*

Directory	Description
<code>/opt/bladeservices/bin</code>	Contains HPI-B sub-agent daemon binaries
<code>/opt/bladeservices/share/snmp/mibs</code>	Contains MIB file SAI-HPI-SNMP-MIB-B.01.01.mib used by HPI-B SNMP subagent.
<code>/opt/bladeservices/etc/snmp</code>	Configuration file for Hpi-B subagent
<code>/usr/share/snmp/mibs</code>	Contains mib file SAF-TC-MIB.mib used by HPI-B SNMP subagent and Saf application
<code>/etc/init.d</code>	Contains HPI-B SNMP daemon start/stop scripts

### 2.4.2 Configuring an SNMP Agent

Once the SNMP agent files are installed and the single shelf library used by the SNMP subagent is configured appropriately as described in the previous sections, you need to configure the SNMP master agent as follows.

#### Configuring the SNMP Master Agent

The following configurations can be grouped into the following categories:

- General configuration
  - SNMP V2 and V3 related configuration
  - Trap message configuration
1. Open the SNMP master agent configuration file `/etc/snmp/snmpd.conf` in a text editor
  2. Make sure that the file contains the following entries:  

```
master agentx
AgentXTimeout 600
AgentXRetries 20
```

The values specified for `AgentXTimeout` and `AgentXRetries` should have at least the values 600 and 20.

## Software Installation and Configuration

---

- In order to configure the SNMP master agent for SNMP V2, add the following line to `snmpd.conf`:

```
com2sec notConfigure default rwcommunity
rwcommunity public
```

To test the SNMP V2 configuration, save the configuration file and perform a SNMP walk by entering the following command at the ATCA-F125/F140 command line:

```
# snmpwalk -v 2c -c public localhost -m
/opt/bladeservices/share/snmp/mibs/SAI-HPI-SNMP-MIB-B.01.01.mib
hpib0101
```
- In order to configure the SNMP master agent for SNMP V3, add the following line to `snmpd.conf`:

```
createUser LocalUser MD5 "LocalUserPassword" DES
"localUserPassword"
rwuser LocalUser
```

To test the SNMP V3 configuration, save the configuration file and perform a SNMP walk by entering the following command at the ATCA-F125/F140 command line:

```
snmpwalk -v3 -n "" -l AuthPriv -u "LocalUser" -a MD5-A
"LocalUserPassword" -x DES -X "LocalUserPassword" -m
/opt/bladeservices/share/snmp/mibs/SAI-HPI-SNMP-MIB-B.01.01.mib
localhost hpib0101
```
- If you are using **SNMP V2** and want to enable SNMP traps to be sent to a host, enter the following line to `snmpd.conf`

```
trap2sink <Host IP> public
```

<Host IP> is the IP address of the host where you want the traps to be sent to.
- If you are using **SNMP V3** and want to enable SNMP traps to be sent to a host, enter the following line to `snmpd.conf`

```
trapsess -e <Destination snmp v3 device engine-id> -v 3 -n "" -l
<Authoriation/Encryption detail> -u <User-name> -a MD5 -A
"<Authorization-Password>" -x DES -X "<Encryption Password>" <Dest-
Ip>:162
```

This creates a session with the host where the traps are to be sent to (as required by SNMP V3). An example entry might look as follows:

```
trapsess -e 0x800000000102030405 -v 3 -n "" -l AuthPriv -u
LocalUser -a MD5 -A "LocalUserPassword" -x DES -X
"LocalUserPassword" 192.168.21.1:162
```

# Developing Applications

## 3.1 Overview

This chapter describes how to develop applications that make use of the HPI-B library.

Depending on the CPU architecture of the target system where you want to run your HPI-B application and on the operating system, different RPM files are delivered which contain include files and static libraries needed for the application development. The naming scheme used for these files is as follows: `bbs-hpib-devel-<version>-1.<architecture>-<distribution>-<os>.rpm`

In order to run your HPI-B clients, you furthermore need to install the HPI-B client base package applicable to the blade where the client is running. See [Setting Up HPI Clients on page 34](#).

The HPI-B client base package contains compiled example applications which illustrate the use of HPI-B controls. For these example applications the source codes and an example make file are available as well. You may want to use the source code and the make file as a starting point for developing your own applications. For further details refer to [Example Applications on page 65](#).

## 3.2 Building the Application

If your development system is based on the same operating system/CPU architecture environment as the target system, then you can simply install the RPM files on the target system. If the development system is based on another operating system/CPU architecture environment and you consequently intend to do cross-compilation, then the RPM files should be converted to the `cpio` format and then extracted, using the standard Linux `rpm2cpio` tool.

In order to do this, you would for example enter the following at the command prompt:

1. `cd <working directory>`
2. `rpm2cpio <rpm file> | cpio -id`

After extracting the RPM or CPIO files, you obtain the following directories with the following contents.

Table 3-1 Development RPMs - Directory Structure

Directory	Content
<code>/opt/bladeservices/include/bbs-hpib</code>	Include files
<code>opt/bladeservices/lib</code> or <code>opt/bladeservices/lib64</code>	Static libraries



# Using HPI-B

---

## 4.1 Overview

This chapter provides information which is necessary when writing applications that are based on the Penguin Solutions HPI-B distribution. It lists limitations with respect to the HPI-B specification and describes extensions which were added by Penguin Edge.

## 4.2 Limitations

This section describes those HPI-B features which the Penguin Edge HPI-B implementation for the Centellis 4440 environment does not support.

### 4.2.1 Limitations with Respect to HPI-B Base Specification

The following limitations apply with respect to the compatibility with the SAI-HPI-B.01.02 specification. Note that these limitations apply to the current and also to all future Penguin Edge HPI-B releases for Centellis 4440 platforms. There are no plans to implement these features in the future.

- Firmware Update Management Instrument (FUMI) functionality not supported.
- Diagnostic Management Instrument (DIMI) functionality not supported.
- Limited `saHpiIdrAreaAdd()` call  
If the space is available, the function `saHpiIdrAreaAdd()` adds an OEM Inventory Area including two pre-defined fields as multi-record with a maximum size of 255 Byte. The first one is a read-only field containing the ManufacturerID (3 bytes). The second field (252 bytes) can be updated by the user. The Penguin Edge HPI-B implementation of `saHpiIdrAreaAdd()` does not support the creation of other types of Inventory Areas.
- Limited `saHpiIdrAreaDelete()` call  
Penguin Edge HPI-B does not allow deleting Inventory Area with `saHpiIdrAreaDelete()`, except OEM Multi Records not specified by the PICMG ATCA and AMC specifications.
- Unsupported `saHpiParamControl()` call  
Penguin Edge HPI-B does not support `saHpiParamControl()`.
- Unsupported resource event log  
Penguin Edge HPI-B does not support resource event logs.
- Unsupported Unicode character set  
Penguin Edge HPI-B does not support the Unicode character set.

## Using HPI-B

---

- Unsupported annunciator functionality  
Penguin Edge HPI-B does not support annunciator functionality. Our platforms do not have these features.
- Optional APIs `saHpiInitialize()` and `saHpiFinalize()` are not supported.

### 4.2.2 Limitations with Respect to HPI-B AdvancedTCA Mapping Specification

The following limitations apply with respect to the compatibility with the HPI-B-AdvancedTCA mapping specification SAIM-HPI-B.01.01-ATCA. Note that these limitations apply to the current and also to all future Penguin Edge HPI-B releases. There are no plans to implement these features in the future.

- Only physical slot numbers are supported  
Penguin Edge HPI-B only supports physical slot numbers in entity paths.
- Only "shall" and "should" requirements are supported  
Penguin Edge HPI-B only supports the "shall" and "should" requirements of the HPI-to-AdvancedTCA mapping specification SAIM-HPI-B.01.01-ATCA.
- Following Inventory records are not exposed to the HPI users through the HPI Inventory APIs:
  - Radial IPMB-0 Link Mapping Record (PICMG Record Id 15h), version 1
  - PICMG Form Factor Information Record (PICMG Record Id 2Eh)
  - Shelf Fan Geography Record (PICMG Record Id 1Bh)
  - Shelf Configuration Information Inventory
  - LED Description Record Fields (PICMG Record Id 2Fh) for AdvancedTCA FRUs.
- E-Keying sensors for the Synchronization Clock Bus and the Metallic Test Bus are not supported.
- According to mapping spec, Maximum power capability sensor is deprecated and provided only for backward compatibility.
- The Max value of Assigned Power Sensors for the Managed FRU slots shall not be set. HPI users can check the Max value for the Assigned Power Sensor of the Managing FRU slots to determine the total power capability of that slot along with all slots being managed under the same hardware address.

## 4.3 Backward Compatibility

This section describes the various aspects that HPI-B user should make a note while migrating from HPI-B 02.01 based applications to HPI-B 03.02, or in using the HPI-B 02.01 applications with the HPI-B 03.02 daemon. The following points could impact existing HPI-B client applications.



**It is recommended for the user to note the differences in the specification and then modify the client applications suitably.**

- Some management instruments and resources that were mandatory in SAIM-HPI-B.01.01-ATCA are either made optional or removed in the latest specification.
- Some of the RDR data has been modified to better describe the management instruments in this specification. This modification includes changing entity paths, sensor types, and control output types on various management instruments. HPI-B user applications compliant to HPI-B 02.01 specification will receive data as per the B03.02 xTCA mapping specification. These applications need to be modified to interpret the data correctly.
- The symbolic names of all defined values are changed. A user program compiled with previous header files should still be usable with HPI library implementations that are built using the new symbols defined in this specification. We have taken care of this part by back porting the Xtca prefix to Atca Prefix.

Following table describes the backward compatibility:

*Table 4-1 Backward Compatibility matrix*

HPI-B Daemon version	HPI-B Library version	Compatibility
B.03.02	B.02.01	Yes, functionality as B.03.02 daemon specs compliance. Refer to the backward compatibility notes in the Mapping specification.
B.02.01	B.03.02	Yes, Only B.02.01 functionality.

# 4.4 Working with the Multishelf Library

The multishelf HPI library allows your application to connect to and manage several shelves at the same time. For this purpose the multishelf library provides several HPI controls which allow the application to manage the connection to shelves and also to dynamically add and remove shelves to the HPI environment. This section provides all the information that you need to know in order to use these HPI controls and work with the HPI multishelf library.

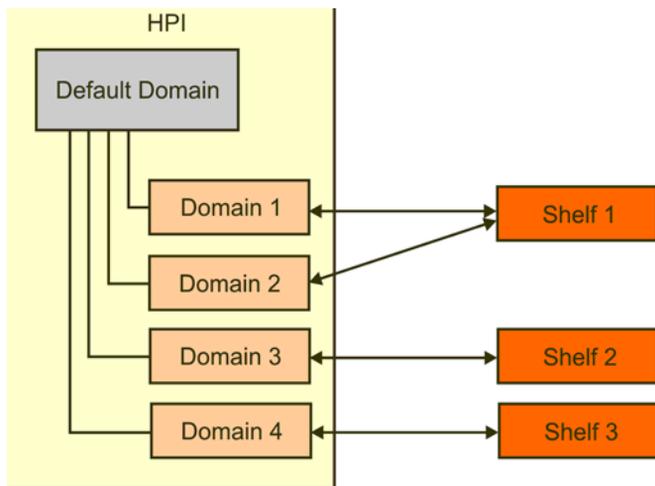
## 4.4.1 Overview

HPI uses the concept of domains. Generally, a domain represents one shelf. Furthermore there is a default domain. It acts as a container for all other domains and does itself not represent actual hardware.

It is possible for multiple domains to represent the same physical shelf. This is for example the case in typical AdvancedTCA systems which often provide two shelf managers with an HPI daemon running on each of them.

The following figure illustrates an example configuration with four domains and three shelves.

Figure 4-1 Multishelf Library - Representation of Shelves as Domains



Any FRUs available in a shelf are represented as HPI resources together with Resource Data Records (RDRs) corresponding to that FRU. Whenever a FRU is added to or removed from a shelf, the corresponding HPI resource/RDR is added/removed from the HPI domain.

## 4.4.2 Accessing HPI Domains

In order to access an HPI domain, you must open a session via the HPI call `saHpiSessionOpen()` and provide as first parameter the domain ID of the corresponding HPI domain. How to obtain the domain ID is described later within this section about the multishelf HPI library.

If you want to access several shelves, then you need to open several sessions simultaneously, one session for each HPI domain which represents a shelf. It is also possible to open several sessions for one HPI domain/shelf only as well.

When the connection to a shelf is lost, all running HPI calls which access the corresponding domain return immediately with the error code `SA_HPI_ERR_NO_RESPONSE`. All open sessions for the affected domain are automatically closed by the multishelf library. In the meantime, the library tries to regain access to the shelf. As soon as the connection is reestablished, the domain is recreated and the application can open another session and access the domain again. Whenever a domain is created or removed, an HPI event from the HPI Communication State sensor is generated in the default domain (see [Connection State Sensor on page 50](#) and [HPI Domain Events on page 51](#)).

HPI events are handled domain wide. This means that HPI events from a shelf or FRUs in that shelf are only visible and can only be received within the session that corresponds to that domain.

## 4.4.3 How Domains and Shelves are Represented

As previously mentioned, each HPI implementation has at least the default domain. It has the ID 0 assigned to it.

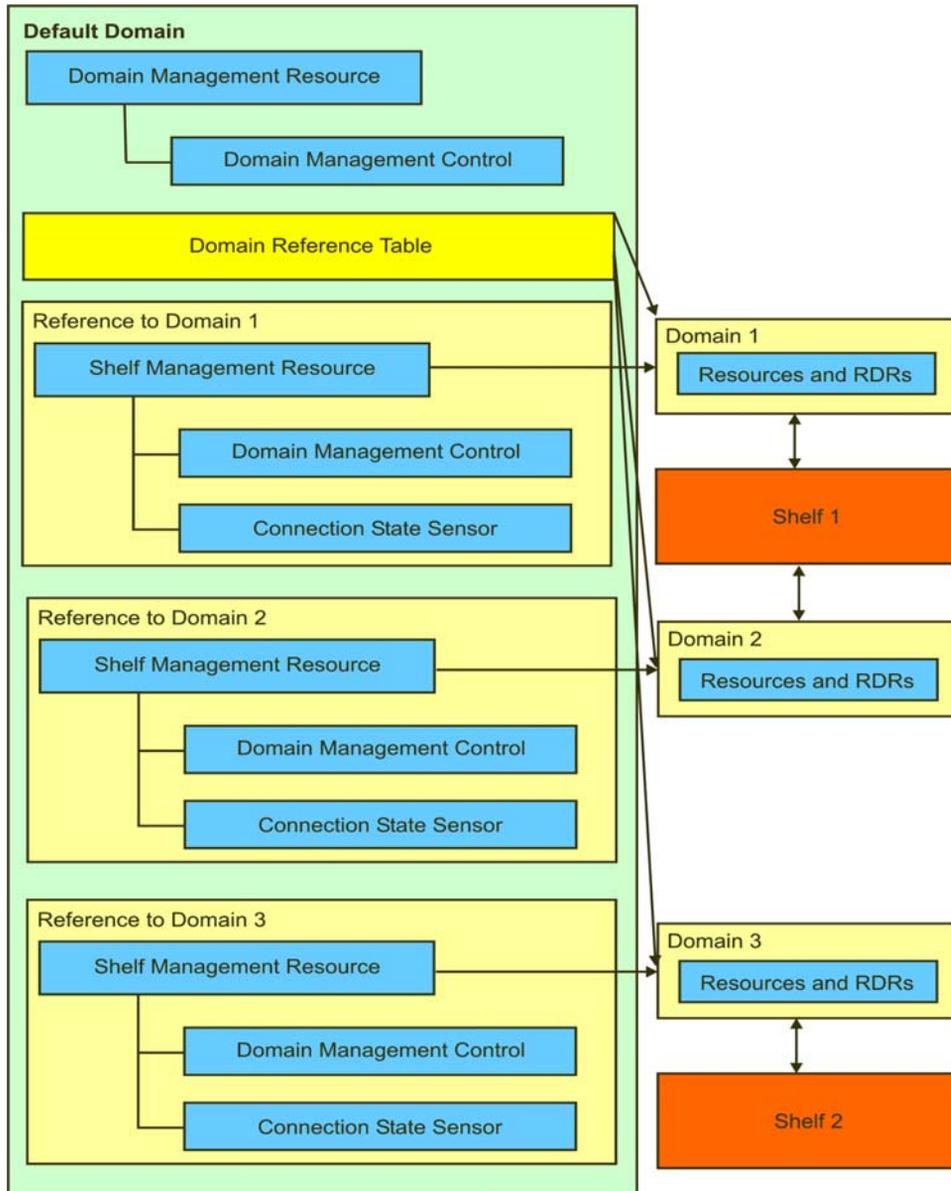
Starting with HPI-B, the default domain contains a Domain Reference Table, which contains references to all related domains and may be used by applications for discovery of available domains in the current configuration. For more information about the Domain Reference Table, refer to the HPI-B specification document of the SAI-HPI-B.02.01 standard.

In the Penguin Edge HPI-B implementation, the default domain furthermore contains multiple HPI resources which handle connected shelves and their corresponding HPI domains. These HPI resources were defined and added by Penguin Edge and are called Domain Management Resource and Shelf Management Resource.

The Domain Management Resource contains one HPI control and allows applications to add/remove HPI domains/shelves to the HPI environment. The Shelf Management Resource contains one HPI control and one HPI sensor and acts as reference to connected daemons. There is one Shelf Management Resource for each connected daemon. The following figure shows an example configuration with the HPI resources, controls and sensors which are related to the handling of multiple shelves/domains in it.

## Using HPI-B

Figure 4-2 HPI Multishelf Library - Overview of Related HPI Resources and Controls



In the following, the Domain Management Resource and the Shelf Management resource will be described in detail. A description of typical usage examples/scenarios will be given after that.

### 4.4.3.1 Domain Management Resource

The Domain Management Resource acts as container for the Domain Management Control and is defined as follows.

Table 4-2 Definition of Domain Management Resource

SaHpiRptEntryT	Value
EntryId	Assigned by HPI
ResourceId	Assigned by HPI
ResourceInfo	0 for all values
ResourceEntity	{RACK, 0} this can be changed with the multishelf library configuration file
ResourceCapabilities	SAHPI_CAPABILITY_RESOURCE   SAHPI_CAPABILITY_RDR   SAHPI_CAPABILITY_CONTROL
ResourceSeverity	SAHPI_MAJOR
DomainId	0
IdString	Domain management

This Domain Management control is only writable, not readable, and allows the application to dynamically add and remove domains.



**Adding/removing a domain using this HPI control has the same effect as adding/removing a shelf by adding/removing an entry in the multishelf library configuration file. Therefore, whenever you use this HPI control to add/remove a domain, the software automatically updates the configuration file as well.**

The RDR and the HPI control are defined as follows:

Table 4-3 Domain Management Control RDR

SaHpiRdrT	Value
RecordId	Assigned by HPI
RdrType	SAHPI_CTRL_RDR
Entity	The same entity like Domain Management Resource
RdrTypeUnion	Defined in <a href="#">Table 4-4</a> .

## Using HPI-B

Table 4-3 Domain Management Control RDR (continued)

SaHpiRdrT	Value
IdString	MOTHPI_CTRL_NAME_DOMAIN_MANAGEMENT

Table 4-4 Domain Management Control

SaHpiCtrlRecT	Value
Num	MOTHPI_CTRL_NUM_DOMAIN_MANAGEMENT
Ignore	SAHPI_FALSE
OutputType	SAHPI_CTRL_OEM
Type	SAHPI_CTRL_TYPE_OEM
TypeUnion - Oem -Mld	MOTHPI_MANUFACTURER_ID_MOTOROLA
TypeUnion - Oem -ConfigData	0
TypeUnion - Oem - Default - Mld	0
TypeUnion - Oem - Default - BodyLength	0
TypeUnion - Oem - Default - Body	0
Oem	0

Table 4-5 Domain Management Control State

SaHpiCtrlStateT	Value
Type	SAHPI_CTRL_TYPE_OEM
StateUnion - Oem - Mld	MOTHPI_MANUFACTURER_ID_MOTOROLA
StateUnion - Oem - BodyLength	Depends on the length of the resource name

Table 4-5 Domain Management Control State (continued)

SaHpiCtrlStateT	Value
StateUnion - Oem - Body	Sequence of n bytes, named [0] ... [n], with the following definitions: [0] - Command 0 - Get state 1 - Create domain (for set state) 2 - Remove domain (for set state) [1] - [4] - IP address (little endian order) [5][6] - port (little endian order) [7] ... [n] - Domain name as null-terminated string.

#### 4.4.3.2 Shelf Management Resource

The shelf management resource represents one daemon. For each configured daemon, the default domain creates one shelf management resource. The shelf management resource is defined as follows.

Table 4-6 Shelf Management Resource

SaHPIRptEntryT	Value
EntryId	Assigned by HPI
ResourceId	Assigned by HPI
resourceInfo	0 for all values
ResourceEntity	{SYS_MNGMT_SOFTWARE domain Id}
ResourceCapabilities	SAHPI_CAPABILITY_RESOURCE   SAHPI_CAPABILITY_RDR   SAHPI_CAPABILITY_CONTROL   SAHPI_CAPABILITY_SENSOR When the shelf is reachable: SAHPI_CAPABILITY_DOMAIN
ResourceSeverity	SAHPI_MAJOR
DomainId	Domain Id when the shelf is reachable
IdString	Domain name

The shelf management resource contains the following two RDRs:

## Using HPI-B

- Domain management control  
Read-only HPI control which allows the application to obtain the IP address, port, domain name and other connection parameters of a particular shelf
- Connection status sensor  
Contains information about the current status of the connection to a shelf

The definitions of both HPI controls are given in the following.

### 4.4.3.2.1 Domain Management Control

This HPI control is only readable. It has the same structure and definition as the HPI controls defined in the following tables: [Domain Management Control on page 48](#), [Domain Management Control State on page 48](#) and [Domain Management Control RDR on page 47](#).

### 4.4.3.2.2 Connection State Sensor

This HPI sensor represents the status of the connection to a shelf. When the connection is interrupted, an HPI event of type SAHPI\_ES\_OFF\_LINE is generated. When the connection is reestablished, an HPI event of type SAHPI\_ES\_ON\_LINE is generated. An application can access a shelf only when the state of this sensor is SAHPI\_ES\_ON\_LINE. The definition of this sensor is given in the following tables.

Table 4-7 Connection State RDR

SaHpiRdrT	Value
RecordId	Assigned by HPI
RdrType	SAHPI_SENSOR_RDR
Entity	The same entity as domain management resource
RdrTypeUnion	MOTHPI_SENSOR_NAME_DOMAIN_CONNECTION

Table 4-8 Domain Connection Sensor

SaHpiSensorRecT	Value
Num	MOTHPI_SENSOR_NUM_DOMAIN_CONNECTION
Type	SAHPI_CHASSIS
Category	SAHPI_EC_AVAILABILITY
EventCtrl	SAHPI_SEC_PER_EVENT
Events	SAHPI_ES_ON_LINE   SAHPI_ES_OFF_LINE

Table 4-8 Domain Connection Sensor (continued)

SaHpiSensorRecT	Value
Ignore	SAHPI_FALSE
DataFormat - ReadingFormats	SAHPI_SRF_EVENT_STATE
DataFormat - IsNumeric	SAHPI_FALSE
DataFormat - IsThreshold	SAHPI_FALSE
Oem	0

Table 4-9 Domain Connection Sensor Reading

SaHpiSensorReadingT	Value
ValuesPresent	SAHPI_SRF_EVENT_STATE
EventStatus - SensorStatus	SAHPI_SENSTAT_EVENTS_ENABLED   SAHPI_SENSTAT_SCAN_ENABLED
AssertEvents	SAHPI_ES_ON_LINE for a connection to a shelf SAHPI_ES_OFF_LINE when there is no connection to a shelf

### 4.4.3.3 HPI Domain Events

Depending on the connection status, the Domain Connection sensor can throw events. The following table provides details.

Table 4-10 Domain Connection Sensor Events

Event	Description
SAHPI_ES_OFF_LINE	No connection
SAHPI_ES_ON_LINE	Connection is established. Only now is the domain of the shelf accessible.

## 4.4.4 Discovering Shelves and Domains

As previously mentioned, one way to discover available domains is to use the Domain Reference Table. For further details, refer to the SAI-HPI-B.01.02 specification document.

## Using HPI-B

---

Alternatively, you can use the Shelf Management Resources defined in the default domain. Any shelves that were added to the HPI environment are represented by one Shelf Management Resource with the entry ResourceCapability set to SAHPI\_CAPABILITY\_DOMAIN. So in order to discover shelves/domains, you simply need to parse the default domain for HPI resources with the entry ResourceCapability set to SAHPI\_CAPABILITY\_DOMAIN. Then you can obtain the respective domain ID by reading the entry DomainId and open a session to the shelf using the obtained domain ID.

### 4.4.5 Adding and Removing Shelves and Domains

All shelves which are used by the HPI multishelf library are listed in the multishelf library configuration file. There are two ways of adding and removing shelves:

- Manually by editing the multishelf configuration file. Changes become effective after the next restart in this case.
- Dynamically by using the HPI domain management control

For a description of how to manually edit the multishelf library configuration file, refer to [Configuring HPI Clients on page 35](#).

A description of how to dynamically add and remove shelves, is given in the following two subsections.

#### 4.4.5.1 Adding Shelves and Domains

In order to dynamically add a shelf and create a domain, you must invoke the HPI function `saHpiControlStateSet()` and provide the following parameters.

Parameter	Value
SessionId	Handle to session context.
ResourceId	Resource ID of the addressed resource.
CtrlNum	Number of the control for which the state is being set.
CtrlState	Pointer to control state as described in table <a href="#">Domain Management Control State on page 48</a> . In this control state, fill the field <code>body</code> as follows: [0] : 1 (stands for "create domain") [1] - [4] : IP address of shelf you want to add (little endian order) [5] - [6] : port of shelf (little endian order) [7] - [n] : Domain name

After calling `saHpiControlStateSet()`, a new resource with the domain name you supplied as parameter is created. Additionally, a hot swap event of type ACTIVE for the created resource is sent. The newly created shelf is also added to the HPI multishelf configuration file as a new entry.



**All shelves must have different IP addresses.**

#### 4.4.5.2 Removing Shelves and Domains

In order to dynamically remove a shelf/domain, you must invoke the HPI function `saHpiControlStateSet()` and provide the following parameters.

Parameter	Value
SessionId	Handle to session context.
ResourceId	Resource ID of the addressed resource.
CtrlNum	Number of the control for which the state is being set.
CtrlState	<p>Pointer to control state as described in <a href="#">Table "Domain Management Control State" on page 45</a>.</p> <p>In this control state, fill the field <code>body</code> as follows:</p> <p>[0] : 2 (stands for "remove domain")</p> <p>[1] - [4] : IP address of shelf you want to add (little endian order)</p> <p>[5] - [6] : port of shelf (little endian order)</p> <p>[7] - [n] : Domain name</p>

If the domain/shelf specified in the call exists, the domain/shelf is removed, an HPI domain removed event is generated and any sessions that may be open to this domain are closed. Furthermore the domain shelf resource in the HPI domain is removed and a hot swap even of type "NOT\_PRESENT" is generated.

## 4.5 Using the SNMP Agent

Once the SNMP agent is installed and configured as described in [Installing and Configuring an SNMP Agent on page 36](#), you can start/stop the SNMP subagent and master agent as described below.

To start/stop the SNMP subagent, enter `# /etc/init.d/hpibsnmp < start > or <stop>` on the ATCA-F125/F140.

To start/stop the SNMP master agent, enter `# /etc/init.d/snmpd start|stop`

## Using HPI-B

---

To check if the SNMP mast is running, enter: `# ps -aux | grep snmp`

If it is running, the following should be displayed: `/usr/sbin/snmpd -Lsd -Lf /dev/null -p /var/run/snmpd.pid.`

For a detailed description of the SAI-HPI-B.01.01 MIB refer to the respective specifications which can be downloaded from the Service Availability Forum's website at <http://www.saforum.org>.

## 4.6 Penguin Edge Extensions

The following describes features which are not specified in the HPI-B specifications, but were added by Penguin Solutions.

### 4.6.1 HPI Controls for Domain and Shelf Management

This refers to the Domain Management Resource and Shelf Management Resource which were previously explained. Both are Penguin Edge-specific extensions. See [Working with the Multishelf Library on page 44](#).

### 4.6.2 IPMI System Boot Options Support

A FRU may have a payload which is capable of booting an operating system (OS). Usually, a boot firmware, such as BIOS or U-Boot, is started after the payload is powered up or reset. Via the System Boot Options Control you can set some options for the boot firmware. The boot firmware will read these settings from the IPMC.

In order to set or get the system boot options, you need to use the HPI Boot Option control.

This control maps the IPMI commands `Set System Boot Options` and `Get System Boot Options` to HPI.



**The Boot Option Control is only available for AdvancedTCA front blades and the shelf manager if the respective IPMC supports the `Set System Boot Option IPMI` command. Refer to the Intelligent Platform Management Interface Specification v2.0, section 28.12 `Set System Boot Options Command` and 28.13 `Get System Boot Options Command`, for further details. Furthermore refer to the respective IPMI Programmer's Reference manuals of the respective blades.**

Table 4-11 *Boot Option RDR*

SaHpiRdrT	Value
RecordId	Assigned by HPI
RdrType	SAHPI_CTRL_RDR

Table 4-11 Boot Option RDR (continued)

SaHpiRdrT	Value
Entity	The same entity as resource
RdrTypeUnion	Defined in next table
IdString	MOTHPI_CTRL_NAME_BOOT_OPTION

Table 4-12 Boot Option Control

SaHpiCtrlRecT	Value
Num	MOTHPI_CTRL_NUM_BOOT_OPTION
Ignore	SAHPI_FALSE
OutputType	SAHPI_CTRL_OEM
Type	SAHPI_CTRL_TYPE_OEM
TypeUnion.Oem.Mid	MOTHPI_MANUFACTURER_ID_MOTOROLA
Oem	0

Table 4-13 Boot Option State

SaHpiCtrlStateT	Value
Type	SAHPI_CTRL_TYPE_OEM
StateUnion.Oem.Mid	MOTHPI_MANUFACTURER_ID_MOTOROLA
StateUnion.Oem.BodyLength	<p><b>Get operations:</b> When input parameter: 3 When output parameter: Total length of the response data - 2</p> <p><b>Set operations:</b> Total length of the request data</p>

## Using HPI-B

Table 4-13 Boot Option State (continued)

SaHpiCtrlStateT	Value
StateUnion.Oem.Body	<b>Get operations:</b> When input parameter: Byte 0: Parameter selector Byte 1: Set selector Byte 2: Block selector When output parameter: Response data from the GetSystemBootOptions IPMI command without the first two bytes <b>Set operations:</b> Request data for the IPMI SetSystemBootOptions command

Example: On most Penguin Edge blades, the BIOS software is stored twice on a flash device, BIOS bank 1 and BIOS bank 2. The bank to boot from can be selected with the System Boot Options Control.

You have to select parameter 96 with the Boot Option Select Control and then set the BIOS 1 (0) or BIOS 2 (1) with the Boot Option Control.

The following example shows how to set BIOS 2 for a resource with the ID 120 using an example program which is delivered together with the HP-B development package:

```
hpibootoptions -r 120 96 1
```

The example program hpibootoptions, which is available in the base RPM package, shows how an option can be set or got.

### 4.6.3 POST Type Control

This HPI control allows you to set/get the Power-On Self Test (POST) type of the blade. Two POST types are configurable: long POST and short POST. Refer to the respective hardware user manual of the blade for details about both POST types. The definition of the respective HPI controls is given in the following tables.

Table 4-14 POST Type HPI Control RDR

SaHpiRdrT	Value	Notes
RecordId	Assigned by HPI implementation	Unique identifier for the RDR
RdrType	SAHPI_CTRL_RDR	

Table 4-14 POST Type HPI Control RDR (continued)

SaHpiRdrT	Value	Notes
RdrTypeUnion.CtrlRec.Num	MOTHPI_CTRL_NUM_POSTTYPE	
RdrTypeUnion.CtrlRec.OutputType	SAHPI_CTRL_OEM	Indicates that this control does not correlate to any of the given control output types, and describes a generic control output.
RdrTypeUnion.CtrlRec.Type	SAHPI_CTRL_TYPE_DISCRETE	
RdrTypeUnion.CtrlRec.TypeOfUnion.Discrete.Default	0x0	
RdrTypeUnion.CtrlRec.Oem	Assigned by HPI implementation	
IdString	MOTHPI_CTRL_NAME_POSTTYPE	

Table 4-15 POST Type HPI Control State Values

SaGPISateT	Value	Notes
Type	SAHPI_CTRL_TYPE_DISCRETE	
StateUnion.Discrete	Bits 7..0: 0: Short POST 1: Long POST Bits 15..8: CPU complex number Bits 31..16: Reserved	The POST type value is specified in the least significant byte.  The CPU number (or SET Selector byte) is normally zero. However, for blades that support more than one CPU complex, the processor complex is identified with this field.

## 4.6.4 HPI Logging Support

The actions taken by the HPI daemon are written to a log file. The path and name of the log file is defined in the configuration file `bbs-hpib.conf`.

To control what kind of information is written to the log file, the log control can be used.

## Using HPI-B

Refer to the header file `MotorolaHpi.h` for used defines.

Table 4-16 Log RDR

SaHpiRdrT	Value
RecordId	Assigned by HPI
RdrType	SAHPI_CTRL_RDR
Entity	The same entity as logical shelf resource
RdrTypeUnion	Defined in next table
IdString	MOTHPI_CTRL_NAME_LOG found in <code>MotorolaHpi.h</code>

Table 4-17 Log Control

SaHpiCtrlRecT	Value
Num	MOTHPI_CTRL_NUM_LOG
Ignore	SAHPI_FALSE
OutputType	SAHPI_CTRL_OEM
Type	SAHPI_CTRL_TYPE_OEM
TypeUnion.Oem.Mid	MOTHPI_MANUFACTURER_ID_MOTOROLA (2x)
TypeUnion.Oem.ConfigData	
TypeUnion.Oem.Default.Mid	MOTHPI_MANUFACTURER_ID_MOTOROLA (2x)
TypeUnion.Oem.Default.BodyLength	0
TypeUnion.Oem.Default.Body	0
Oem	0

Table 4-18 Log Control State

SaHpiCtrlStateT	Value
Type	SAHPI_CTRL_TYPE_OEM
StateUnion.Oem.Mid	MOTHPI_MANUFACTURER_ID_MOTOROLA
StateUnion.Oem.BodyLength	40 + length of log file name

Table 4-18 Log Control State (continued)

SaHpiCtrlStateT	Value
StateUnion.Oem.Body	<p>MOTHPI_LOG_ERROR_FACILITIES_OFFSET - facilities for that error logging is enabled</p> <p>MOTHPI_LOG_WARNING_FACILITIES_OFFSET - facilities for that warning logging is enabled</p> <p>MOTHPI_LOG_INFO_FACILITIES_OFFSET - facilities for that info logging is enabled</p> <p>MOTHPI_LOG_DEBUG_FACILITIES_OFFSET - facilities for that debug logging is enabled</p> <p>The logging facilities are defined in the file log_utils.h, enum oh_log_fac.</p> <p>MOTHPI_LOG_CURRENT_IDX_OFFSET - index of currently used log file</p> <p>[MOTHPI_LOG_OFFSET_PROPERTIES] - output locationdLogStdout - stdoutdLogStderr - stderrdLogFile - log file</p> <p>[MOTHPI_LOG_NUM_FILES_OFFSET] - number of logfile created</p> <p>[MOTHPI_LOG_MAX_FILE_SIZE_OFFSET] - maximum logfile size before creating a new one. This is a 32 bit field in MSB byte order. Use GetUInt32 in byte_utils.h to get host byte order.</p> <p>[MOTHPI_LOG_FILENAME_OFFSET] - log file name.</p>

## 4.6.5 Failed Slot Restore Control

"Failed Slot Restore Control" enables the user to initiate a re-discovery of the resource in the slot and sets the `ResourceFailed` flag in RPT to `false`. The `ResourceFailed` flag in the RPT Entry of slot resource is set to `true` if the discovery of the resource in the slot fails, or the resource in the slot is in failed state. The `ResourceFailed` flag of the slot resource can be set to `false` in any one of the following conditions:

1. Resource in failed state is removed using `saHpiResourceFailedRemove()`.
2. If any of the resource is inserted into the failed slot.

## Using HPI-B

An example application `hpslotrestore` can be used to set and get the new control "Failed Slot Restore" state.

Table 4-19 Failed Slot Restore Control RDR

SaHpiRdrT	Value	Notes
RecordId	Assigned by HPI implementation.	
RdrType	SAHPI_CTRL_RDR	
Entity	Entity path of the slot resource.	
IsFru	SAHPI_FALSE	
RdrTypeUnion.CtrlRec.Num	MOTHPI_CTRL_FAILED_SLOT_RESTORE_NUM	
RdrTypeUnion.CtrlRec.Type	SAHPI_CTRL_TYPE_DIGITAL	
RdrTypeUnion.CtrlRec.OutputType	SAHPI_CTRL_GENERIC	
RdrTypeUnion.CtrlRec.TypeUnion.Digital.Default	SAHPI_CTRL_STATE_OFF	
RdrTypeUnion.CtrlRec.DefaultMode.Mode	SAHPI_CTRL_MODE_MANUAL	
RdrTypeUnion.CtrlRec.DefaultMode.ReadOnly	SAHPI_TRUE	
RdrTypeUnion.CtrlRec.WriteOnly	SAHPI_FALSE	
RdrTypeUnion.CtrlRec.OEM	Assigned by HPI implementation.	
RdrTypeUnion.CtrlRec.IdString	MOTHPI_CTRL_FAILED_SLOT_RESTORE_NAME	

Table 4-20 Failed Slot Restore Control States

SaHpiCtrlStateT	Value	Notes
Type	SAHPI_CTRL_TYPE_DIGITAL	
StateUnion.Digital	SAHPI_CTRL_STATE_PULSE_ON   SAHPI_CTRL_STATE_OFF	

## 4.6.6 HPI Restart Daemon Control

HPI Restart Daemon control enables the user to restart the hpi-b daemon. hpirestartdaemon client application makes use of this control to restart the daemon.

Table 4-21 HPI Restart Daemon RDR

SaHpiRdrT	Value
RecordId	Assigned by HPI implementation
RdrType	SAHPI_CTRL_RDR
Entity	Entity path of 'Daemon Resource' Ex: {ADVANCEDTCA_CHASSIS, 6} {SYS_MGMNT_SOFTWARE, 0}
RdrTypeUnion.CtrlRec.Num	MOTHPI_CTRL_NUM_HPI_RESTART_DAEMON
RdrTypeUnion.CtrlRec.Type	SAHPI_CTRL_TYPE_DIGITAL
RdrTypeUnion.CtrlRec.OutputType	SAHPI_CTRL_GENERIC
RdrTypeUnion.CtrlRec.TypeUnion.Digital.Default	SAHPI_CTRL_STATE_OFF
RdrTypeUnion.CtrlRec.DefaultMode.Mode	SAHPI_CTRL_MODE_MANUAL
RdrTypeUnion.CtrlRec.DefaultMode.ReadOnly	SAHPI_TRUE
RdrTypeUnion.CtrlRec.WriteOnly	SAHPI_FALSE
RdrTypeUnion.CtrlRec.Oem	Assigned by HPI implementation.
RdrTypeUnion.CtrlRec.IdString	MOTHPI_CTRL_NAME_HPI_RESTART_DAEMON

Table 4-22 HPI Restart Daemon State

SaHpiCtrlStateT	Value
Type	SAHPI_CTRL_TYPE_DIGITAL
StateUnion.Digital	SAHPI_CTRL_STATE_PULSE_ON   SAHPI_CTRL_STATE_OFF

### 4.6.7 IPMI Command Control

IPMI command control is used to execute IPMI commands on a given resource id. This control is created for all the FRUs in the shelf. `hpiipmi` client application makes use of this control to get the IPMI commands executed.

Table 4-23 IPMI Command RDR

SaHpiRdrT	Value
RecordId	Assigned by HPI implementation
RdrType	SAHPI_CTRL_RDR
Entity	Entity path of the FRU Ex: {ADVANCEDTCA_CHASSIS,6}{FAN_TRAY_SLOT,1} {COOLING_UNIT,1}
RdrTypeUnion	Defined in next table
IdString	MOTHPI_CTRL_NAME_BOOT_OPTION

Table 4-24 IPMI Command Control

SaHpiCtrlRecT	Value
Num	MOTHPI_CTRL_NUM_IPMI_COMMAND
Type	SAHPI_CTRL_TYPE_OEM
OutputType	SAHPI_CTRL_OEM
TypeUnion.Oem.Mid	OHHPI_MANUFACTURER_ID_MOTOROLA
TypeUnion.Oem.ConfigData	
TypeUnion.Oem.Default.Mid	OHHPI_MANUFACTURER_ID_MOTOROLA
TypeUnion.Oem.Default.BodyLength	0
TypeUnion.Oem.Default.Body	0
DefaultMode.Mode	SAHPI_CTRL_MODE_MANUAL
DefaultMode.ReadOnly	SAHPI_TRUE
WriteOnly	SAHPI_FALSE
Oem	0

Table 4-25 IPMI Command State

SaHpiCtrlStateT	Value
Type	SAHPI_CTRL_TYPE_OEM
StateUnion.Oem.MId	OHHPI_MANUFACTURER_ID_MOTOROLA
StateUnion.Oem.BodyLength	IPMI command request/response length ( max - 255 ) Ex: When we issue a 'Get Device ID' command like, hpiipmi -d 1 -r 43 0 6 1 BodyLength would be 3 (0 6 1) For the response, the BodyLength would be 16 bytes
StateUnion.Oem.Body	IPMI command request/response bytes Request - all 0's Response - ipmi command response bytes

## 4.6.8 Boot Bank Control

Boot Bank control is used to switch the boot bank of Penguin Edge specific dual-flash-bank boards. This control is created for all front blades, AMCs, RTMs, and Shelf Managers. hpibootbanks client application makes use of this control to set/get boot bank parameters.

Table 4-26 Boot Bank RDR

SaHpiRdrT	Value
RecordId	Assigned by HPI implementation
RdrType	SAHPI_CTRL_RDR
Entity	Entity path of the managing FRU which owns the control Ex: {ADVANCEDTCA_CHASSIS, 6} {SHELF_MANAGER_SLOT, 1} {SHELF_MANAGER, 1}
RdrTypeUnion	Defined in next table
IdString	MOTHPI_CTRL_NAME_BOOT_BANK

Table 4-27 Boot Bank Control

SaHpiCtrlRecT	Value
Num	MOTHPI_CTRL_NUM_BOOT_BANK
Type	SAHPI_CTRL_TYPE_OEM

## Using HPI-B

Table 4-27 Boot Bank Control (continued)

SaHpiCtrlRecT	Value
OutputType	SAHPI_CTRL_OEM
TypeUnion.Oem.MId	OHHPI_MANUFACTURER_ID_MOTOROLA
TypeUnion.Oem.ConfigData	
TypeUnion.Oem.Default.MId	OHHPI_MANUFACTURER_ID_MOTOROLA
TypeUnion.Oem.Default.BodyLength	0
TypeUnion.Oem.Default.Body	0 (255 times)
DefaultMode.Mode	SAHPI_CTRL_MODE_MANUAL
DefaultMode.ReadOnly	SAHPI_FALSE
WriteOnly	SAHPI_FALSE
Oem	0

Table 4-28 Boot Bank State

SaHpiCtrlStateT	Value
Type	SAHPI_CTRL_TYPE_OEM
StateUnion.Oem.MId	OHHPI_MANUFACTURER_ID_MOTOROLA
StateUnion.Oem.BodyLength	2 ( max - 255 )
StateUnion.Oem.Body	For Set or Get, the body will be Offset 0 - processor id Offset 1 - bank id

# Example Applications

---

## 5.1 Overview

The HPI-B client base package contains precompiled example applications. They are invoked via the command line and can be configured via command line parameters. Each example application illustrates a certain feature of HPI-B and makes use of the respective HPI-B function calls.

After extracting the HPI-B client RPMs, the example applications can be found in the following directory: `/opt/bladeservices/bin`. You can obtain information about the command usage by invoking the application from the command line and providing `-h` as parameter.



**The HPI-B example programs are provided "as is" without any warranty of any kind, either express or implied. The entire risk as to the quality, operability and execution of the programs is with you. Should the programs prove to be faulty or incorrect, you assume the cost of all necessary servicing, repair or correction. In no event Penguin Solutions will be liable to you for any damages, any lost profits or other special, incidental or consequential damages arising out of the use or inability to use the programs.**

**Penguin Edge reserves the right to revise or remove the programs in subsequent releases without obligation of Penguin Edge to notify any person of such revision or changes.**

## 5.2 Example Application Source Files

In order to ease application development and help you to get familiar with the HPI-B API usage, Penguin Edge provides the source files of the example applications and an example make file. These files are available as different RPM files, depending on the operating system and CPU architecture. The naming scheme used for the RPMs is: `bbs-hpib-clientsrc-<version>-1.<CPU architecture>-<distribution>-<os>.rpm`

After installing the RPMs, the source files and the example make file are located in the following directory: `/opt/bladeservices/src/bbs-hpib/clients`.

### 5.3 List of Supported Example Applications

The following is an automatically generated output (based on the files contained in `/opt/bladeservices/bin` and the output obtained via the `-h` parameter) that describes all HPI example applications which were available when this manual was written and the usage of these applications. Note that in the meantime further example applications may have been added or the functionality of existing HPI example applications may have been changed slightly. The current usage and functionality can always be obtained by invoking the example application with the `-h` parameter.

```
#
# Lists the usage of all supported HPI-B example programs.
#
# Copyright (c) 2007 by Motorola GmbH
# Copyright (c) 2008, Emerson Network Power - Embedded Computing GmbH
#
#-----#
# hpiautotimer
#-----#
Usage: hpiautotimer [OPTION]...
HPI example application to manage the timeout values of the auto insert
timer and auto extract timer.
Options:
  -d DOMAIN_ID      use domain with id DOMAIN_ID
  -D                walk recursively through DRT
  -r RESOURCE_ID    use resource with id RESOURCE_ID
  -i INSERT_TIMEOUT set auto insert timeout value in msec to
INSERT_TIMEOUT
  -e EXTRACT_TIMEOUT set auto extract timeout value in msec to
EXTRACT_TIMEOUT
  -V                print version information and exit
  -h                display this help and exit
#-----#
# hpibootbanks
#-----#
Usage: hpibootbanks [OPTION]...
HPI example application to switch the boot bank of Motorola/Emerson
specific dual-flash-bank boards.
Options:
```

```
-d DOMAIN_ID    use domain with id DOMAIN_ID
-r RESOURCE_ID  use resource with id RESOURCE_ID
-b BANK_NUM     set boot bank number to BANK_NUM [0..<bankNumber>]
-p PROCESSOR_ID set boot bank of procesor with this ID
-V             print version information and exit
-h            display this help and exit

#-----#
# hpibootoptions
#-----#
Usage: hpibootoptions [OPTION]... [PARAM# PARAMS]
HPI example application to manage the system boot options defined in IPMI
v2.0 .
Options:
  -d DOMAIN_ID    use domain with id DOMAIN_ID
  -r RESOURCE_ID  use resource with id RESOURCE_ID
  -c              turn on console redirection
                  (Not applicable on some boards)
  -o              turn off console redirection
                  (Not applicable on some boards)
  -x              perform FRU cold-reset
  -V             print version information and exit
  -h            display this help and exit

Examples:
Set the system boot option 1 (service partition) to value 2 for resource
4 on domain 0:
  hpibootoptions -d 0 -r 4 1 2
Turn on console redirection and perform a cold-reset for resource 34 on
domain 0:
  hpibootoptions -d 0 -r 34 -c -x

#-----#
# hpibootparameter
#-----#
Usage: hpibootparameter [OPTION]... [PARAM# PARAMS]
HPI example application to manage IPMI boot parameter.
Options:
  -d DOMAIN_ID    use domain with id DOMAIN_ID
```

## Example Applications

---

```
-r RESOURCE_ID  use resource with id RESOURCE_ID
-g              get IPMI boot parameter USER area
-b              get IPMI boot parameter DEFAULT area
-s filename     set IPMI Boot parameter, read from file
-c              clear IPMI Boot parameter stored in USER area
-x              additionally output the IPMI boot parameter as hexdump
-V              print version information and exit
-h              display this help and exit

#-----#
# hpichassisstatus
#-----#
Usage: hpichassisstatus [OPTION]...
HPI example application to display the control state of the chassis status
control.
Options:
  -d DOMAIN_ID  use domain with id DOMAIN_ID
  -D            walk recursively through DRT
  -r RESOURCE_ID use resource with id RESOURCE_ID
  -V            print version information and exit
  -h            display this help and exit

#-----#
# hpicooling
#-----#
Usage: hpicooling [OPTION]...
HPI example application to control the cooling mode.
Note: Only applicable if the HPI daemon runs in Shelf Manager mode.
Options:
  -d DOMAIN_ID  use domain with id DOMAIN_ID
  -D            walk recursively through DRT
  -r RESOURCE_ID use resource with id RESOURCE_ID
  -m mode       set cooling mode [1=AUTO, 2=MANUAL]
  -V            print version information and exit
  -h            display this help and exit

#-----#
# hpidomain
#-----#
```

Usage: hpidomain [OPTION]...

HPI example application to display all domains found.

Options:

- V print version information and exit
- h display this help and exit

#-----#

# hpidomainel

#-----#

Usage: hpidomainel [OPTION]...

HPI example application to display the domain event log.

Options:

- d DOMAIN\_ID use domain with id DOMAIN\_ID
- c clear the event log
- A display everything
- t display RDR with the event log
- p display RPT with the event log
- x display debug messages
- V print version information and exit
- h display this help and exit

#-----#

# hpidomainself

#-----#

Usage: hpidomainself [OPTION]...

HPI example application to print the domain ID where this program is running on.

Note: Requires multishelf library.

Options:

- V print version information and exit
- h display this help and exit

#-----#

# hpifailedextract

#-----#

Usage: hpifailedextract [OPTION]...

HPI example to remove a failed resource using the failed extract control.

Options:

- d DOMAIN\_ID use domain with id DOMAIN\_ID

## Example Applications

---

```
-r RESOURCE_ID  use resource with id RESOURCE_ID to extract
-a              display all resources (default: list only failed resources)
-V             print version information and exit
-h             display this help and exit

#-----#
# hpifan
#-----#
Usage: hpifan [OPTION]...
HPI example application to control the cooling mode.
Note: Only applicable if the HPI daemon runs in Shelf Manager mode.
Options:
  -d DOMAIN_ID  use domain with id DOMAIN_ID
  -D            walk recursively through DRT
  -r RESOURCE_ID use resource with id RESOURCE_ID
  -s level      set cooling level (only in MANUAL mode)
  -V           print version information and exit
  -h           display this help and exit

#-----#
# hpifruactivation
#-----#
Usage: hpifruactivation [OPTION]...
HPI example application to manage the FRU activation mode.
Options:
  -d DOMAIN_ID  use domain with id DOMAIN_ID
  -D            walk recursively through DRT
  -r RESOURCE_ID use resource with id RESOURCE_ID
  -a {0|1}      disable/enable ShM activation
  -w DELAY      set delay before next power on to DELAY * 1/10 sec
  -V           print version information and exit
  -h           display this help and exit

#-----#
# hpifrudeactivation
#-----#
Usage: hpifrudeactivation [OPTION]...
HPI example application to manage the FRU deactivation mode.
Options:
```

```
-d DOMAIN_ID      use domain with id DOMAIN_ID
-D                walk recursively through DRT
-r RESOURCE_ID    use resource with id RESOURCE_ID
-a {0|1}          enable/disable ShM deactivation
-V                print version information and exit
-h                display this help and exit

#-----#
# hpifruipmreset
#-----#
Usage: hpifruipmreset [OPTION]...
HPI example application to set FRU IPMC Reset Control actions.
Options:
  -d DOMAIN_ID      use domain with id DOMAIN_ID
  -r RESOURCE_ID    use resource with id RESOURCE_ID
  -c                IPMC cold reset
  -w                IPMC warm reset
  -V                print version information and exit
  -h                display this help and exit

#-----#
# hpifruresetdiag
#-----#
Usage: hpifruresetdiag [OPTION]...
HPI example application to set FRU Reset and Diagnostic Control actions.
Options:
  -d DOMAIN_ID      use domain with id DOMAIN_ID
  -r RESOURCE_ID    use resource with id RESOURCE_ID
  -s VALUE           set control value VALUE
                   [1=Graceful Reboot, 2=Diagnostic Interrupt]
  -V                print version information and exit
  -h                display this help and exit

#-----#
# hpifumi
#-----#
Usage: hpifumi [OPTION]...
HPI example application to manage the firmware upgrade of FRUs using FUMI.
Options:
```

## Example Applications

---

```
-d DOMAIN_ID      use domain with id DOMAIN_ID
-r RESOURCE_ID   use resource with id RESOURCE_ID
-f FUMI_NUM      use fumi with number FUMI_NUM
-b BANK_NUM      use bank with number BANK_NUM
-n COMPONENT_ID use Component with number COMPONENT_ID
-Z Rollback_disable use value for rollback disable 0/1
-s URI           calls saHpiFumiSourceSet URI
-u              calls saHpiFumiInstallStart
-v              calls saHpiFumiSourceInfoValidateStart
-i              calls saHpiFumiSourceInfoGet
-t              calls saHpiFumiTargetInfoGet
-a              calls saHpiFumiActivate
-g              calls saHpiFumiUpgradeStatusGet
-y              calls saHpiFumiTargetVerifyStart
-c              calls saHpiFumiUpgradeCancel
-I              calls saHpiFumiSpecInfoGet
-D              calls saHpiFumiServiceImpactGet
-C              calls saHpiFumiSourceComponentInfoGet
-T              calls saHpiFumiTargetComponentInfoGet
-P              calls saHpiFumiLogicalTargetInfoGet
-L              calls saHpiFumiLogicalTargetComponentInfoGet
-M              calls saHpiFumiTargetVerifyMainStart
-G              calls saHpiFumiAutoRollbackDisableGet
-R              calls saHpiFumiAutoRollbackDisableSet
-A              calls saHpiFumiActivateStart
-U              calls saHpiFumiCleanUp
-V              print version information and exit
-h              display this help and exit

#-----#
# hpiha
#-----#
Usage: hpiha [OPTION]...
HPI example application to show the HA state and to initiate a switch-over.
Options:
  -d <domain ID>  use domain with ID <domain ID>
  -m              initiate a Shelf Manager switch-over
```

```
-s          initiate a HPI Daemon switch-over
-V         print version information and exit
-h         display this help and exit
#-----#
# hpihotswap
#-----#
Usage: hpihotswap [OPTION]...
HPI example application to invoke hotswap actions.
Note: If calling without options the application switches to interactive
mode.
Options:
  -d DOMAIN_ID    use domain with id DOMAIN_ID
  -s              show hotswap resources
  -a RESOURCE_ID  activate resource
  -t RESOURCE_ID  deactivate resource
  -i RESOURCE_ID  insert resource
  -e RESOURCE_ID  extract resource
  -p RESOURCE_ID  get powerstate of resource
  -u RESOURCE_ID  power up resource
  -o RESOURCE_ID  power down resource
  -y RESOURCE_ID  power cycle resource
  -c RESOURCE_ID  cold-reset resource
  -V             print version information and exit
  -h            display this help and exit
#-----#
# hpiidh
#-----#
Usage: hpiidh [OPTION]...
HPI example application to manage the Inventory Data
Options:
  -d DOMAIN_ID    use domain with id DOMAIN_ID
  -r RESOURCE_ID  use resource with id RESOURCE_ID
  -t TARGET        repository target: [[idr][:area][:field]]
  -a              add new custom Area
  -u              add new FORCE User Info Area
  -s STRING        set a Field specified by -t with STRING
```

## Example Applications

---

```
-f FILE          set a Field specified by -t from FILE
-w FILE          write Field specified by -t to FILE
-e              erase Area specified by -t
-x              display binary fields in HEX format
-V              print version information and exit
-h              display this help and exit

#-----#
# hpiipmb0
#-----#
Usage: hpiipmb0 [OPTION]...
HPI example application to set the IPMB-A or IPMB-B state control.
Options:
  -d DOMAIN_ID  use domain with id DOMAIN_ID
  -D            walk recursively through the DRT
  -r RESOURCE_ID use resource with id RESOURCE_ID
  -a {0|1}      isolate/join IPMB-A
  -b {0|1}      isolate/join IPMB-B
  -l LINK_NUM   use link number LINK_NUM when isolating the IPMB
  -V            print version information and exit
  -h            print this help and exit

#-----#
# hpiipmi
#-----#
Usage: hpiipmi [OPTION]... LUN NETFN CMD [DATA]
HPI example application to send native IPMI commands using the
Motorola/Emerson specific IPMI control.
LUN, NETFN, CMD and DATA will be interpreted as hexadecimal values
Options:
  -d DOMAIN_ID  use domain with id DOMAIN_ID
  -r RESOURCE_ID use resource with id RESOURCE_ID
  -V            print version information and exit
  -h            display this help and exit

Example:
Send GetDeviceId command (Lun=0, Netfn=6, Cmd=1) to resource 43 on domain
1:
    hpiipmi -d 1 -r 43 0 6 1
```

```
#-----#
# hpled
#-----#
Usage: hpled [OPTION]...
HPI example application to control the LEDs.
Options:
  -d DOMAIN_ID    use domain with id DOMAIN_ID
  -r RESOURCE_ID  use resource with id RESOURCE_ID
  -n CONTROL_ID   use control with id CONTROL_ID for set operations
  -a              set control mode to auto
  -l RATE         set led on duration rate in 1/100sec [0-255]
  -o RATE         set led off duration rate in 1/100sec [0-255]
  -t RATE         led test
  -V             print version information and exit
  -h             display this help and exit
#-----#
# hpilink
#-----#
Usage: hpilink [OPTION]...
HPI example application to display E-Keying link states and optionally
listen for link state events.
Options:
  -d DOMAIN_ID    use domain with id DOMAIN_ID
  -r RESOURCE_ID  use resource with id RESOURCE_ID
  -e             listen for E-Keying link state events
  -V             print version information and exit
  -h             display this help and exit
#-----#
# hpilist
#-----#
Usage: hpilist [OPTION]...
HPI example application to list all RPT/RDR entries and to get all events.
Options:
  -d DOMAIN_ID    use domain with id DOMAIN_ID
  -D             walk recursively through DRT
  -r RESOURCE_ID  use resource with id RESOURCE_ID
```

## Example Applications

---

```
-s                output short information (don't display RDR data)
-y SENSOR_NUM    return value of sensor number SENSOR_NUM of resource
with id RESOURCE_ID
-z [1/0]         enable/disable events from sensor number SENSOR_NUM of
resource with id RESOURCE_ID
-t              don't display Timestamps
-p CSIWDF        display only Control, Sensor, Watchdog, Inventory, Dimi, Fumi
                (ignored if no valid parameter is given)
-e              listen for events
-n              no discover (useful for just getting events)
-i              interactive mode (for development use only)
-V              print version information and exit
-h              display this help and exit
#-----#
# hpilog
#-----#
Usage: hpilog [OPTION]...
HPI example application to control the logger.
Options:
  -d <domain ID>  use domain with ID <domain ID>
  -c <ctrl num>   use control with number <cntrl num>
  -p <properties> set log properties to <properties>
  -q <properties> reset log properties <properties>
  -r <severities> <facilities>
                  reset log <severities> for <facilities>
  -s <severities> <facilities>
                  set log <severities> for <facilities>
  -V              print version information and exit
  -h              display this help and exit

Supported properties:
stdout|stderr|file|simplefile|syslog|prefix|threadid|
Supported severities: debug|info|warning|error|all
```

```
Supported facilities:
other|connection|transport|session|plugin|remote|daemon|client|core|ha|h
picall|resource|sensor|control|inventory|watchdog|dimi|fumi|hotswap|sel
ipmi|ipmicon|ipmidump|ipmimcthread|ipmidiscover|ipmimc|ipmisdrr|ipmiservi
ce|ipmicooling|ipmipower|ipmiptpekeying|ipmisub|softwareupgrade|redundan
cy|script|shfruvalidation|sdrrepository|rmcp|deassert|solmgmt|marshal|al
l
#-----#
# hpi-posttype
#-----#
Usage: hpi-posttype [OPTION]...
HPI example application to control POST type.
Options:
  -d DOMAIN_ID    use domain with id DOMAIN_ID
  -D              walk recursively through DRT
  -r RESOURCE_ID  use resource with id RESOURCE_ID
  -t POSTTYPE     set POST Type value
  -c CPU_NUM      set CPU number [default=0]
  -V              print version information and exit
  -h              display this help and exit
#-----#
# hpi-poweronsequence
#-----#
Usage: hpi-poweronsequence [OPTION]...
HPI example application to manage the power on sequence of FRUs during
initial startup.
Options:
  -d DOMAIN_ID    use domain with id DOMAIN_ID
  -D              walk recursively through DRT
  -r RESOURCE_ID  use resource with id RESOURCE_ID
  -p POSITION       use POSITION as power on position
  -s SLOT_RES_ID  set SLOT_RES_ID for specific position
                  (Requires '-r' and '-p' option)
  -c              commit power on sequence to FRU Info
  -V              print version information and exit
  -h              display this help and exit
#-----#
```

## Example Applications

---

```
# hpireset
#-----#
Usage: hpireset [OPTION]...
HPI example application to reset a FRU resource.
Options:
  -d DOMAIN_ID    use domain with id DOMAIN_ID
  -r RESOURCE_ID  use resource with id RESOURCE_ID
  -c              cold-reset resource (Requires '-r' option)
  -w              warm-reset resource (Requires '-r' option)
  -V              print version information and exit
  -h              display this help and exit
#-----#

# hpiresourceself
#-----#
Usage: hpiresourceself [OPTION]...
HPI example application to print the resource ID and entity path where
this program is running on.
Options:
  -d DOMAIN_ID    use domain with id DOMAIN_ID
  -V              print version information and exit
  -h              display this help and exit
#-----#

# hpirestartdaemon
#-----#
Usage: hpirestartdaemon [OPTION]...
HPI example application to restart the HPI daemon.
Options:
  -d DOMAIN_ID    use domain with id DOMAIN_ID
  -V              print version information and exit
  -h              display this help and exit
#-----#

# hpishaddr
#-----#
Usage: hpishaddr [OPTION]...
HPI example application to display and set the shelf address.
Options:
```

```
-d DOMAIN_ID    use domain with id DOMAIN_ID
-b HEX_STRING   set shelf address using binary hex string HEX_STRING
-V             print version information and exit
-h             display this help and exit
#-----#
# hpishelf
#-----#
Usage: hpishelf [OPTION]...
HPI example application to manage connections to domains.
Note: Requires multishelf library.
Options:
  -c DOMAIN_NAME  create domain with name DOMAIN_NAME
  -l DOMAIN_NAME  delete domain with name DOMAIN_NAME
  -i IP_ADDR      use IP address IP_ADDR to connect to domain
  -p PORT         use port PORT to connect to domain
  -V             print version information and exit
  -h             display this help and exit

Examples:
Add domain "Gandalf" with IP address 192.168.111.86:
  hpishelf -c Gandalf -i 192.168.111.86
Delete domain "Gandalf":
  hpishelf -l Gandalf
#-----#
# hpiship
#-----#
Usage: hpiship [OPTION]...
HPI example application to display and set the Shelf Manager IP address.
Options:
  -d DOMAIN_ID  use domain with id DOMAIN_ID
  -n CTRL_NUM   use control with number CTRL_NUM
  -i IP_ADDR    set IP address IP_ADDR
  -m NETMASK    set netmask NETMASK
  -g GW_ADDR    set default gateway address GW_ADDR
  -V           print version information and exit
  -h           display this help and exit
#-----#
```

## Example Applications

---

```
# hpislotrestore
#-----#
Usage: hpislotrestore [OPTION]...
HPI example application to initiate slot re-discovery
Options:
  -d DOMAIN_ID    use domain with id DOMAIN_ID
  -D              walk recursively through DRT
  -r RESOURCE_ID  use resource with id RESOURCE_ID
  -R              restore failed slot
  -V              print version information and exit
  -h              display this help and exit
#-----#
# hpisol
#-----#
Usage: hpisol [OPTION]... [PARAM# PARAMS]
HPI example application to manage IPMI v2.0 SOL (Serial over LAN)
settings.
Options:
  -d DOMAIN_ID    use domain with id DOMAIN_ID
  -r RESOURCE_ID  use resource with id RESOURCE_ID
  -V              print version information and exit
  -h              display this help and exit
Examples:
Show SOL configuration for domain 1:
  hpisol -d 1
Set SOL param 1 (SOL Enable) to value 1 for resource 4 on domain 1:
  hpisol -d 1 -r 4 1 1
#-----#
# hpitelcoalarm
#-----#
Usage: hpitelcoalarm [OPTION]...
HPI example application to control telco alarms.
Options:
  -d DOMAIN_ID    use domain with id DOMAIN_ID
  -r RESOURCE_ID  use resource with id RESOURCE_ID
  -i VALUE        set minor alarm value
```

```
-a VALUE      set major alarm value
-c VALUE      set critical alarm value
-V           print version information and exit
-h           display this help and exit
#-----#
# hptop
#-----#
Usage: hptop [OPTION]...
HPI example application to display system topology.
Options:
  -d DOMAIN_ID  use domain with id DOMAIN_ID
  -r RESOURCE_ID use resource with id RESOURCE_ID
  -A            display everything
  -p            display RPTs
  -s            display sensors
  -c            display controls
  -w            display watchdogs
  -i            display inventories
  -a            display annunciators
  -x            display debug messages
  -V           print version information and exit
  -h           display this help and exit
#-----#
# hpiversion
#-----#
Usage: hpiversion [OPTION]...
HPI example application to display the version of the different HPI
components.
Options:
  -d DOMAIN_ID  use domain with id DOMAIN_ID
  -i ITEM       print one of the following version item:
  -V           print version information and exit
  -h           display this help and exit
              HPI
              HPI-ATCA-MAPPING
              CLIENT
```

## Example Applications

---

CLIENT\_PROTOCOL  
MULTISHELF  
MULTISHELF-PROTOCOL  
DAEMON  
DAEMON-PROTOCOL  
DAEMON-HA-PROTOCOL

# Related Documentation

## 6.1 Penguin Solutions Documentation

Technical documentation can be found by using the Documentation Search at <https://www.penguinsolutions.com/edge/support/> or you can obtain electronic copies of documentation by contacting your local sales representative.

*Table 6-1 Penguin Solutions Documentation*

Document Title	Document Number
Centellis 4440 Release Document Collection	
HPI-B Subagent User Guide	6806800P17

## 6.2 Related Specifications

For additional information, refer to the following table for related specifications. As an additional help, a source for the listed document is provided. Please note that, while these sources have been verified, the information is subject to change without notice.

*Table 6-2 Related Specifications*

Organization	Document Title
Intel <a href="http://www.developer.intel.com/design/servers/ipm">http://www.developer.intel.com/design/servers/ipm</a>	Platform Management FRU Information Storage Definition v1.0 IPMI Specification v2.0
PICMG <a href="http://www.picmg.org/v2internal/specifications.htm">http://www.picmg.org/v2internal/specifications.htm</a>	PICMG 3.0 Revision 2.0 Advanced TCA Base Specification
Service Availability Forum <a href="http://www.saforum.org">http://www.saforum.org</a>	SAI-HPI-B03.02 Service Availability Forum Hardware Platform Interface specification SAIM-HPI-B.03.02-xTCA Service Availability Forum HPI-to-xTCA Mapping specification





# PENGUIN<sup>™</sup>

SOLUTIONS 

Penguin Solutions is a trade name used by SMART Embedded Computing, Inc., a wholly owned subsidiary of SMART Global Holdings, Inc. Penguin Edge is a trademark owned by Penguin Computing, Inc., a wholly owned subsidiary of SMART Global Holdings, Inc. All other logos, trade names, and trademarks are the property of their respective owners. ©2022 SMART Embedded Computing, Inc.