HP BladeSystem c-Class to HPI Mapping Developers Guide

Published: July 2008
Edition: 2.0

Table of Contents

List of Figures

List of Tables

Introduction

HP BladeSystem c-Class has brought energy efficient design to a whole new level in server infrastructure. Onboard Administrator (OA) is the management module that resides within the HP BladeSystem c-Class enclosure and can be paired with other tools to simplify daily tasks, warn of potential issues, and assist with repairs. OA provides a SOAP/XML interface for managing the HP BladeSystem c-Class.

OpenHPI provides an open source implementation of Hardware Platform Interface (HPI) defined by Service Availability Forum (SAF). OpenHPI's architecture contains a modular mechanism intended to make adding new hardware support easier. Several plug-ins exist in the OpenHPI source tree, giving access to various types of hardware.

The OpenHPI OA SOAP plug-in enables HPI support for HP BladeSystem c-Class enclosures. The OpenHPI OA SOAP plug-in supports Out-of-Band Management that allows it to run on any blade inside or outside the Blade System. The HPI application may run one or more instances of the OpenHPI OA SOAP plug-in in parallel with other plug-ins and communicates with the OA of HP BladeSystem c-Class enclosure using the SOAP/XML interface. The plug-in discovers the HP BladeSystem c-Class hardware resources and then populates OpenHPI data structures. The OpenHPI OA SOAP plug-in then retrieves the hardware events asynchronously and converts them into OpenHPI events.

In the OpenHPI source tree, this plug-in is called `oa_soap` and is referenced by the name `liboa_soap` in the OpenHPI configuration file.

Intended Audience

This document is intended for application developers, programmers, and database administrators who are responsible for developing, testing, administering, and maintaining HP BladeSystem c-Class enclosures.

Additional Resources

For more information about the Onboard Administrator, including the *HP BladeSystem Onboard Administrator User Guide*, see the following website:

http://www.hp.com/servers/blades

Typographic Conventions

This document uses the following typographic conventions.

`Command`

A command name or qualified command phrase.

`ComputerOut`

Text displayed by the computer.

Ctrl-x

A key sequence. A sequence such as **Ctrl-x** indicates that you must hold down the key labeled **Ctrl** while you press another key or button.

`ENVIRONVAR`

The name of an environment variable, for example, `PATH`.

[ERRORNAME]

The name of an error, usually returned in the `errno` variable.

Key

The name of a keyboard key. **Return** and **Enter** both refer to the same key.

Term

The defined use of an important word or phrase.

`UserInput`

Commands and other text that you type.

`VARIABLE`

The name of a placeholder in a command, function, or other syntax display that you replace with an actual value.

`\` *`(continuation character)`*

A backslash (\) at the end of a line of code (such as a command) indicates that the following line of code is contiguous, and you must not insert a line break. This convention facilitates the typesetting of long lines of code examples on a printed page. If you cut and paste sample code from this publication, ensure that you remove backslash characters at line endings.

...

The preceding element can be repeated an arbitrary number of times.

`|`

Separates items in a list of choices.

HP Encourages Your Comments

HP encourages your comments concerning this document. We are committed to providing documentation that meets your needs. Send any errors found, suggestions for improvement, or compliments to:

`docsfeedback@hp.com`

Include the document title, and any comment, error found, or suggestion for improvement you have concerning this document.

Obtaining the OA SOAP Plug-in

The OA SOAP plug-in is included in OpenHPI version 2.11.1 and later. If you are using version 2.11.1 or later, you can skip the "Installing the OA SOAP Plug-in for OpenHPI Version 2.10.2" section and proceed to "Building the OpenHPI Source" (page 9).

A back-ported patch is also available for OpenHPI version 2.10.2. You must perform additional steps to install the patch before you can build the OpenHPI source. The following section provides these steps. The OpenHPI source can be downloaded from the OpenHPI website located at:

http://www.openhpi.org/Downloads

Detailed instructions for installing and building OpenHPI version 2.10.2 are available in Chapter 2 of the *OpenHPI Manual* located at:

http://openhpi.sourceforge.net/manual/c83.html

Installing the OA SOAP Plug-in for OpenHPI Version 2.10.2

This section provides information for installing the OA SOAP plug-in as a patch on OpenHPI version 2.10.2.

For OpenHPI version 2.10.2, the OA SOAP plug-in is dependent on changes made in the ilo2_ribcl plug-in patch for HP ProLiant Rackmount servers. The oa_soap plug-in patch must be installed on top of OpenHPI version 2.10.2 and the ilo2_ribcl_plug-in. Because the patches for the oa_soap and ilo2_ribcl plug-ins modify several common files, you must follow the ordering of the dependencies when performing the installation. The ilo2_ribcl plug-in patch must always be installed before the oa_soap plug-in patch. To obtain a complete source tree for building OpenHPI, use the following procedure.

1. Download the OpenHPI version 2.10.2 source and unpack the tar file by entering the following command:

   `tar -zxvf openhpi-2.10.2.tar.gz`

2. To apply the ilo2_ribcl plug-in patch, perform the following steps:
   a. Go to the `openhpi-2.10.2` directory.
   b. Enter the following command:

      `patch -p1 -i <ilo2_ribcl_patch_filename>`

      where

      <ilo2_ribcl_patch_filename> is the name of the ilor2_ribcl_patch file obtained from the mailing list.

3. To apply the oa_soap plug-in patch, perform the following steps:
   a. Go to the `openhpi-2.10.2` directory.
   b. Enter the following command:

      `patch -p1 -i <oa_soap_patch_filename>`

      where

      <oa_soap_patch_filename> is the name of the oa_soap_patch file obtained from the mailing list.

4. To configure and build the plug-ins, enter the following commands:

   ```
   libtoolize --copy --force --automake
   aclocal
   autoheader
   automake --add-missing --copy --foreign
   autoconf
   ```

5. Build and install the OpenHPI binaries and libraries, as described in"Building the OpenHPI Source" (page 9).

The latest versions of the oa_soap and ilo2_ribcl plug-in patches for OpenHPI version 2.10.2 are published to the OpenHPI-devel mailing list. This list can be found on the OpenHPI project page located at:

http://sourceforge.net/mailarchive/forum.php?forum_name=openhpi-devel

Future enhancements and defect fixes for the plug-ins are published on this website. You can also monitor the Openhpi-devel mailing list or search the mailing list archives for these patch releases. Search for the strings `oa_soap` and `ilo2_ribcl`.

Building the OpenHPI Source

The oa_soap plug-in and the ilo2_ribcl plug-in are built by default during the OpenHPI build process. To disable the build for these plug-ins, add the appropriate configure flag during the configuration process:

Disable the oa_soap plug-in build:

```
--disable-oa_soap
```

Disable the ilo2_ribcl plug-in build:

```
--disable-ilo2_ribcl
```

The ilo2_ribcl plug-in requires that the `openssl-devel` and `libxml2-devel` packages are installed in order to build successfully. Most testing has been performed with openssl-devel version 0.9.8a and libxml2-devel version 2.6.23. HP recommends that you obtain the latest version that is available for your distribution.

The `README` file in the OpenHPI source directory provides more details on building.

To begin the build process, enter the following commands:

```
./configure
```

```
make
```

To install the updated OpenHPI daemon and libraries, verify you have root privileges and enter the following command:

```
make install
```

Configuring the Onboard Administrator

You must set up a user account in the Onboard Administrator (OA) for each HP BladeSystem c-Class enclosure that you want to manage. The OA is configured at the factory with a default user name and password, which can be found on the tag attached to the hardware. To setup or change the login and/or password, refer to the *HP BladeSystem Onboard Administrator User Guide*. The user account for the plug-in on the OA must have adminstrator-level privileges. You must also use OA firmware version 2.02 or later.

OpenHPI OA SOAP Plug-in Configuration File

The Onboard Administrator (OA) is the management module for the entire HP BladeSystem c-Class system. The HP BladeSystem c-Class system can have one or two (in redundant mode) OAs. If the Blade System is equipped with a single OA, then it is an Active OA. If the Blade System is equipped with two OAs, then one of them is Active and the other is StandBy. You should use the Active OA to manage the Blade System. The Active OA can be switched over to become the Standby OA using the web interface or by manually pulling the Active OA. Whenever an OA switchover occurs, the StandBy OA automatically becomes the Active OA.

The OpenHPI OA SOAP plug-in detects the Active and StandBy OAs, based on plug-in configuration details and starts interacting with the Active OA. Whenever an OA switchover occurs, the plug-in detects the switchover and begins interacting with the new Active OA.

The OpenHPI OA SOAP plug-in is configured in the `openhpi.conf` file located in the `/etc/openhpi/` directory. You can configure one or more OA SOAP plug-in instances along with other plug-ins in the `openhpi.conf` file.

The OpenHPI OA SOAP plug-in instance configuration parameters are listed in Table 1.

Table 1 OpenHPI OA SOAP Plug-in Configuration Details

Parameter	Description
entity_root	Indicates the entity root of the entity path. The entity path for the discovered resources are generated by adding the prefix *entity_root* to the location of the resource in the chassis.
OA_User_Name	Holds the OA user name. It is used for authenticating with OA.
OA_Password	Holds the OA password. It is used for authenticating with OA.
ACTIVE_OA	Holds the Active OA IP address.
STANDBY_OA	Holds the StandBy OA IP address. This parameter is optional.

NOTE: If the system only has one OA, then the *ACTIVE_OA* parameter should be specified and the *STANDBY_OA* parameter line should be commented out.

HP BladeSystem c-Class Resources

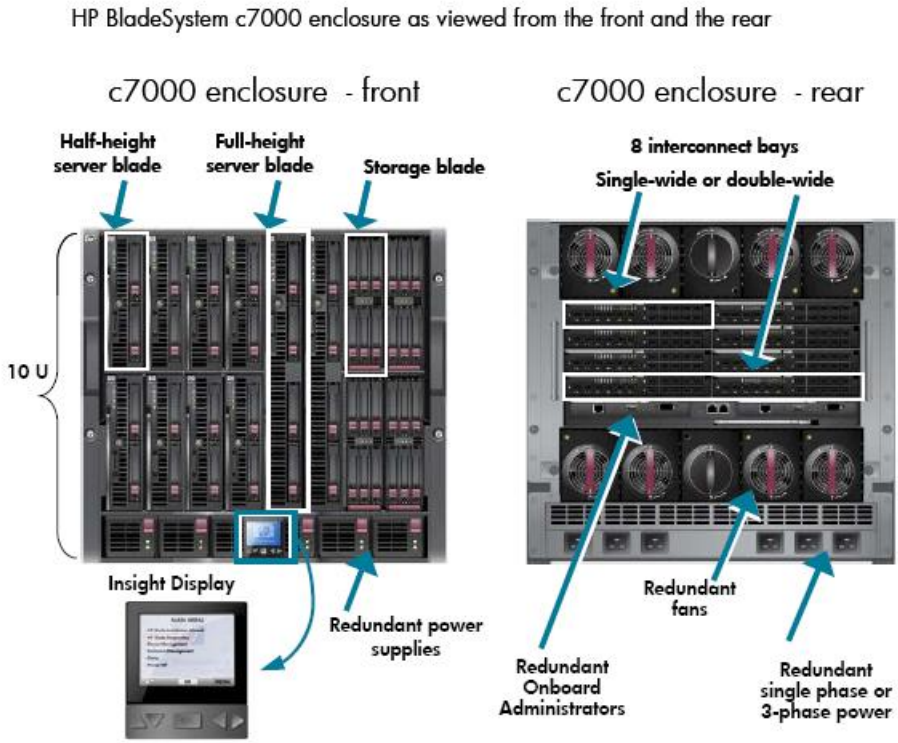
The HP BladeSystem c-Class enclosure c7000 contains the following hardware resources:

- Server Blades– 16 Half Blades or 8 Full Blades or a combination of both
- Interconnects– 8 Single-Wide Interconnects or 4 Double-Wide Interconnects or a combination of both.
- Onboard Administrators (Management Modules)– 2 Redundant OAs
- Fans– 10 Fans
- Power Subsystem– One Power Subsystem with 6 Power Supplies

NOTE: The HP BladeSystem c-Class Enclosure is considered one hardware resource.

As an example, shows the details of the HP BladeSystem c7000 enclosure.

Figure 1 HP BladeSystem c7000 Enclosure Hardware Resources



RPT Mapping

Resource Presence Table (RPT) mapping of the HP BladeSystem c-Class enclosure resources to the HPI resources is specified in Table 2.

Table 2 Resource Mapping

HP c-Class BladeSystem Resource	HPI Resource
c-Class Enclosure	*SYSTEM_CHASSIS*
Onboard Administrator (OA)	*SYS_MGMNT_MODULE*
Server Blade	*SYSTEM_BLADE*
Interconnect	*SWITCH_BLADE*
Fan	*COOLING_DEVICE*
Power Subsystem	*POWER_ MGMNT*
Power Supply	*POWER_SUPPLY*

The HP c-Class enclosure contains the Server Blades, Interconnects, OAs, Fans, and Power Supplies. Therefore, the entity paths for the HP BladeSystem c-Class resources are as provided in Table 3.

Table 3 Resource Entity Path

HP c-Class BladeSystem Resource	Entity Path
c-Class Enclosure	{SYSTEM_CHASSIS, Enclosure Number}
Onboard Administrator (OA)	{SYSTEM_CHASSIS, Enclosure Number} {SYS_MGMNT_MODULE, OA Slot Number}
Server Blade	{SYSTEM_CHASSIS, Enclosure Number} {SYSTEM_BLADE, Blade Slot Number}
Interconnect	{SYSTEM_CHASSIS, Enclosure Number} {SWITCH_BLADE, Interconnect Slot Number}
Fan	{SYSTEM_CHASSIS, Enclosure Number} {COOLING_DEVICE , Fan Slot Number}
Power Subsystem	{SYSTEM_CHASSIS, Enclosure Number} {POWER_MGMNT, Power Subsystem number =1}
Power Supply	{SYSTEM_CHASSIS, Enclosure Number} {POWER_MGMNT, Power Subsystem number =1} {POWER_SUPPLY, Power Supply Slot Number}

The supported resource capabilities for HP BladeSystem c-Class resources are specified in Table 4.

Table 4 Resource Capability

HP c-Class BladeSystem Resource	HPI Resource Capability
c-Class Enclosure	RESOURCE, RDR, INVENTORY_DATA, SENSOR
Onboard Administrator (OA)	RESOURCE, RDR, INVENTORY_DATA, SENSOR ,FRU
Server Blade	RESOURCE, RDR, INVENTORY_DATA, SENSOR ,FRU, MANAGED_HOTSWAP, POWER, RESET, CONTROL
Interconnect	RESOURCE, RDR, INVENTORY_DATA, SENSOR ,FRU, MANAGED_HOTSWAP, POWER, RESET, CONTROL
Fan	RESOURCE, RDR, INVENTORY_DATA, SENSOR ,FRU

Table 4 Resource Capability *(continued)*

HP c-Class BladeSystem Resource	HPI Resource Capability
Power Subsystem	RESOURCE, RDR, SENSOR
Power Supply	RESOURCE, RDR, INVENTORY_DATA, SENSOR ,FRU

RDR Mapping

There are a few general points that are applicable for all HP BladeSystem c-Class Resource Data Records (RDRs). These points are detailed in the following list:

- Controls are supported only on System Blades and Interconnects (Switches).
- Control mode support is manual and read only. For example, `CtrlRec.DefaultMode.Mode = SAHPI_CTRL_MODE_MANUAL.`
- The digital Control states `SAHPI_CTRL_STATE_PULSE_ON` and `SAHPI_CTRL_STATE_PULSE_OFF` are not supported by the System blades and Interconnects (Switches) due to a limitation in the hardware to support the transitory power states.
- All sensors in the Blade System are only of the data type `SAHPI_SENSOR_READING_TYPE_FLOAT64.`
- Thermal sensor events are supported only on System Blades and Interconnects (Switches).
- HPI applications can disable individual sensors. An example for all sensors is `SensorRec.EnableCtrl = SAHPI_TRUE.`
- HPI applications cannot set thresholds. An example for all threshold sensors is `SensorRec.ThresholdDefn.WriteThold = 0.`
- Only one IDR is supported for each resource.

HP Bladesystem c-Class Enclosure RDRs

Table 5 and Table 6 detail the HP BladeSystem c-Class Enclosure RDRs.

Table 5 HP BladeSystem c-Class Enclosure Sensor RDRs

Sensor Name	Sensor Number	Sensor Type	Event Category	Events	Sensor Data Units
Enclosure Temperature	OA_SOAP_ RES_SEN_ TEMP_NUM	SAHPI_ TEMPERATURE	SAHPI_EC_ THRESHOLD	SAHPI_ES_ UNSPECIFIED	SAHPI_SU_ DEGREES_C

Table 6 HP BladeSystem c-Class Enclosure Inventory RDRs

Area Type	Supported Field Types
PRODUCT_INFO	PRODUCT_NAME, MANUFACTURER
CHASSIS_INFO	PART_NUMBER, SERIAL_NUMBER
INTERNAL_USE	MANUFACTURER, PRODUCT_NAME, PART_NUMBER, SERIAL_NUMBER

OA RDRs

Table 7 and Table 8 detail the OA RDRs.

Table 7 OA Sensor RDRs

Sensor Name	Sensor Number	Sensor Type	Event Category	Events	Sensor Data Units
OA Temperature	OA_SOAP_ RES_SEN_ TEMP_NUM	SAHPI_ TEMPERATURE	SAHPI_EC_ THRESHOLD	SAHPI_ES_ UNSPECIFIED	SAHPI_SU_ DEGREES_C

Table 8 OA Inventory RDRs

Area Type	Supported Field Types
PRODUCT_INFO	PRODUCT_NAME, MANUFACTURER
BOARD_INFO	PART_NUMBER, SERIAL_NUMBER

Server Blade RDRs

Table 9, Table 10, and Table 11 detail the Server Blade RDRs.

Table 9 Server Blade Control RDRs

Control Name	Control Type	Control Output Type	Default Mode	Supported Values
Server Power Control	DIGITAL	POWER_STATE	MANUAL	SAHPI_CTRL_ STATE_OFF SAHPI_CTRL_ STATE_ON

Table 10 Server Blade Sensor RDRs

Sensor Name	Sensor Number	Sensor Type	Event Category	Events	Sensor Data Units
Server Board	OA_SOAP_ RES_SEN_ TEMP_NUM	SAHPI_ TEMPERATURE	SAHPI_EC_ THRESHOLD	SAHPI_ES_ UPPER_CRIT SAHPI_ES_ UPPER_MAJOR	SAHPI_SU_ DEGREES_C
Server Board Power Consumed	OA_SOAP_ RES_SEN_ POWER_ NUM	SAHPI_ POWER_UNIT	SAHPI_EC_ THRESHOLD	SAHPI_ES_ UNSPECIFIED	SAHPI_SU_ WATTS

Table 11 Server Blade Inventory RDRs

Area Type	Supported Field Types
PRODUCT_INFO	PRODUCT_NAME, MANUFACTURER
BOARD_INFO	PART_NUMBER, SERIAL_NUMBER

Interconnect RDRs

Table 12 , Table 13, andTable 14 detail the Interconnect RDRs.

Table 12 Interconnect Control RDRs

Control Name	Control Type	Control Output Type	Default Mode	Supported Values
Interconnect Power Control	DIGITAL	POWER_STATE	MANUAL	SAHPI_CTRL_ STATE_OFF SAHPI_CTRL_ STATE_ON

Table 13 Interconnect Sensor RDRs

Sensor Name	Sensor Number	Sensor Type	Event Category	Events	Sensor Data Units
Interconnect Temperature	OA_SOAP_ RES_SEN_ TEMP_NUM	SAHPI_ TEMPERATURE	SAHPI_EC_ THRESHOLD	SAHPI_ES_ UPPER_CRIT SAHPI_ES_ UPPER_MAJOR	SAHPI_SU_ DEGREES_C

Table 14 Interconnect Inventory RDRs

Area Type	Supported Field Types
PRODUCT_INFO	PRODUCT_NAME, MANUFACTURER
BOARD_INFO	PART_NUMBER, SERIAL_NUMBER

Fan RDRs

Table 15 and Table 16 detail the Fan RDRs.

Table 15 Fan Sensor RDRs

Sensor Name	Sensor Number	Sensor Type	Event Category	Events	Sensor Data Units
Fan Speed	OA_SOAP_ RES_SEN_ FAN_NUM	SAHPI_ COOLING_ DEVICE	SAHPI_EC_ THRESHOLD	SAHPI_ES_ UNSPECIFIED	SAHPI_SU_ RPM
Fan Power Consumed	OA_SOAP_ RES_SEN_ POWER_ NUM	SAHPI_ POWER_ UNIT	SAHPI_EC_ UNSPECIFIED	SAHPI_ES_ UNSPECIFIED	SAHPI_SU_ WATTS

Table 16 Fan Inventory RDRs

Area Type	Supported Field Types
PRODUCT_INFO	PRODUCT_NAME
BOARD_INFO	PART_NUMBER

Power Subsystem RDRs

Table 17 details the Power Subsystem RDRs.

Table 17 Power Subsystem RDRs

Sensor Name	Sensor Number	Sensor Type	Event Category	Events	Sensor Data Units
Power Subsystem Input Power	OA_SOAP_ RES_SEN_ IN_ POWER_ NUM	SAHPI_ POWER_SUPPLY	SAHPI_EC_ UNSPECIFIED	SAHPI_ES_ UNSPECIFIED	SAHPI_SU_ WATTS
Power Subsystem Output Power	OA_SOAP_ RES_SEN_ OUT_ POWER_ NUM	SAHPI_ POWER_SUPPLY	SAHPI_EC_ UNSPECIFIED	SAHPI_ES_ UNSPECIFIED	SAHPI_SU_ WATTS
Power Subsystem Power Consumed	OA_SOAP_ RES_SEN_ POWER_ NUM	SAHPI_ POWER_SUPPLY	SAHPI_EC_ UNSPECIFIED	SAHPI_EC_ UNSPECIFIED	SAHPI_SU_ WATTS
Power Subsystem Power Capacity	OA_SOAP_ RES_SEN_ POWER_ CAPACITY_ NUM	SAHPI_ POWER_SUPPLY	SAHPI_EC_ UNSPECIFIED	SAHPI_EC_ UNSPECIFIED	SAHPI_SU_ WATTS

Power Supply RDRs

Table 18 and Table 19 detail the Power Subsystem RDRs.

Table 18 Power Supply Sensor RDRs

Sensor Name	Sensor Number	Sensor Type	Event Category	Events	Sensor Data Units
Power Output	OA_SOAP_ RES_SEN_ POWER_ NUM	SAHPI_ POWER_ SUPPLY	SAHPI_EC_ UNSPECIFIED	SAHPI_ES_ UNSPECIFIED	SAHPI_SU_ WATTS

Table 19 Power Supply Inventory RDRs

Area Type	Supported Field Types
BOARD_INFO	PART_NUMBER, SERIAL_NUMBER

HPI APIs Support

By default, the OpenHPI framework supports Session Related APIs and Domain Related APIs. This section provides information for the APIs that are supported by HPI.

Resource Discovery

The *saHpiDiscover()* API is implemented in the OA SOAP plug-in. It discovers HP Bladesystem c-Class Enclosure hardware resources and populates the RPT in the OpenHPI framework. The RPT table-related APIs are supported by the OpenHPI framework.

Sensors

Table 20 provides a list of all sensor-related APIs and their functions.

Table 20 Sensor-Related APIs

Sensor API	Description
saHpiSensorReadingGet()	Returns the current reading for the given sensor of the specified resource.
saHpiSensorThresholdsGet()	Returns the current threshold reading for the given sensor of the specified resource.
saHpiSensorThresholdsSet()	Is not supported in OA SOAP plug-in. It always returns `SA_ERR_HPI_UNSUPPORTED_API`.
saHpiSensorTypeGet()	Is supported by OpenHPI framework.
saHpiSensorEnableGet()	Returns the current sensor enable status for the given sensor of the specified resource.
saHpiSensorEnableSet()	Sets the sensor enable status for the given sensor of the specified resource.
saHpiSensorEventEnableGet()	Returns the current sensor event enable status for the given sensor of the specified resource.
saHpiSensorEventEnableSet()	Sets the sensor event enable status for the given sensor of the specified resource.
saHpiSensorEventMasksGet()	Returns the assert and de-assert bit-mask values for the given sensor of the specified resource.
saHpiSensorEventMasksSet()	Sets the assert and de-assert bit-mask values for the given sensor of the specified resource.

Controls

Table 21 provides a list of all control-related APIs and their functions.

Table 21 Control-Related APIs

Control API	Description
saHpiControlTypeGet()	Is supported by OpenHPI framework.
saHpiControlGet()	Returns the current control state and mode for the given control object.
saHpiControlSet()	Sets the control state for the given control object.

Inventory Data Repositories

Table 22 provides a list of all inventory data repository related APIs and their functions.

Table 22 Inventory Date Repository APIs

Inventory Data API	Description
saHpiIdrInfoGet()	Returns the IDR details associated with the given resource.
saHpiIdrAreaHeaderGet()	Returns the IDR Area Header details for a specific area associated with a particular IDR.
saHpiIdrAreaAdd()	Adds an area to the specified IDR.
saHpiIdrAreaAddById()	Adds an area with a specified area id to the specified IDR.
saHpiIdrAreaDelete()	Deletes the specified area from the specified IDR.
saHpiIdrFieldGet()	Returns the Inventory Data Field information from a particular IDA and IDR.
saHpiIdrFieldAdd()	Adds a field to the specified IDA with a specified IDR.
saHpiIdrFieldAddById()	Adds a field with a specified field id to the specified IDA with a specified IDR.
saHpiIdrFieldSet()	Updates the Inventory Data Field for a particular IDA and IDR.
saHpiIdrFieldDelete()	Deletes the specified Inventory Data Field from a particular IDA and IDR.

Watchdog Timers

The Watchdog timer related APIs are not supported in the OA SOAP plug-in. Therefore, all Watchdog timer related APIs return the following:

`SA_ERR_HPI_UNSUPPORTED_API`

Annunciators

The Annunciator-related APIs are not supported in the OA SOAP plug-in. Therefore, all Annunciator-related APIs return the following:

`SA_ERR_HPI_UNSUPPORTED_API`

Diagnostics Initiator Management Instrument (DIMI)

DIMI related APIs are not supported in the OA SOAP plug-in. So, all DIMI related APIs always return `SA_ERR_HPI_UNSUPPORTED_API.`

Firmware Initiator Management Instrument (FUMI)

FUMI related APIs are not supported in the OA SOAP plug-in. So, all FUMI related API's always return `SA_ERR_HPI_UNSUPPORTED_API.`

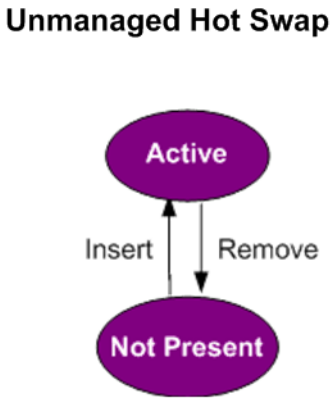
Hot Swap Operations

HP BladeSystem c-Class FRUs currently implement the HPI Unmanaged and Managed Hot Swap Models.

Unmanaged Hot Swap Model

The HP BladeSystem c-Class supports the HPI Unmanaged Hot Swap model for the OA, fan, and power supply FRUs. Therefore, the Hot Swap APIs are not supported for these resources. These FRUs do generate Hot Swap Events.

Figure 2 Unmanaged Hot Swap Model



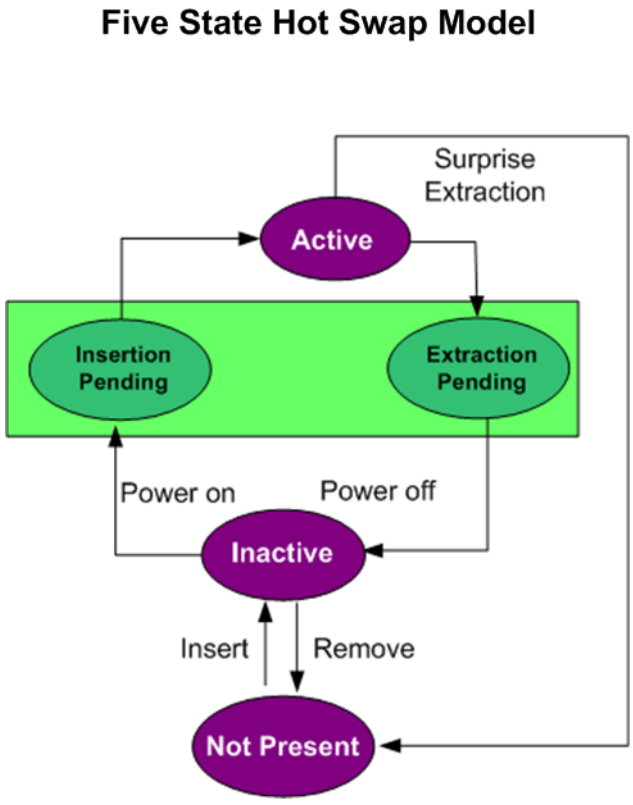
Managed Hot Swap Model

The HP BladeSystem c-Class supports the HPI Five State Managed Hot Swap Model for blade and interconnect FRUs.

The OA plug-in does not currently support the setting of an `AutoInsert` or `AutoExtract` timeout. Instead, these setting are fixed (read-only) and set to `SAHPI_TIMEOUT_IMMEDIATE`. This means that the managed FRUs do not stay in either the *Insertion Pending* or *Extraction Pending* states, but pass immediately into the Active or Inactive states respectively.

Figure 3 displays a simplified view of the hot swap states and transitions that are involved in the Five State Hot Swap Model.

Figure 3 Five State Hot Swap Model



The behavior of the Hot Swap APIs is as follows:

- *saHpiHotSwapStateGet()* API returns the current state of the FRU
- *saHpiHotSwapActionRequest()* API
  - Power on the FRU if it is in the *Inactive* state and the requested action is `SAHPI_HS_ACTION_INSERTION`
  - Power off the FRU if it is in the *Active* state and the requested action is `SAHPI_HS_ACTION_EXTRACTION`

The HP BladeSystem FRUs do not have a specific Hot Swap LED indicator. Therefore, the *saHpiHotSwapIndicatorStateGet()* and *saHpiHotSwapIndicatorStateSet()* APIs always return `SA_ERR_HPI_UNSUPPORTED_API`.

Because the *AutoInsert* and *AutoExtract* timeouts are READ_ONLY, the status of any remaining Hot Swap APIs are as follows:

- *saHpiHotSwapPolicyCancel()* API always returns `SA_ERR_HPI_INVALID_REQUEST`
- *saHpiResourceActiveSet()* API always returns `SA_ERR_HPI_INVALID_REQUEST`
- *saHpiResourceInactiveSet()* API always returns `SA_ERR_HPI_INVALID_REQUEST`
- *saHpiAutoInsertTimeoutGet()* API always returns `SAHPI_TIMEOUT_IMMEDIATE`
- *saHpiAutoInsertTimeoutSet()* API always returns `SA_ERR_HPI_READ_ONLY`
- *saHpiAutoExtractTimeoutGet()* API always returns `SAHPI_TIMEOUT_IMMEDIATE`
- *saHpiAutoExtractTimeoutSet()* API always returns `SA_ERR_HPI_READ_ONLY`

Table 23 provides a list of resources and hot swap events triggered by particular actions.

Table 23 Hot Swap Events

Resource Name	Action	Hot Swap Events	
		Previous State	**Current State**
Server Blade	Insertion	NOT_PRESENT	INSERTION_PENDING
Interconnect Blade (Switch)	Power on after insertion	INSERTION_PENDING	ACTIVE
	Extraction on power on	ACTIVE	NOT_PRESENT
	Extraction on power off	INACTIVE	NOT_PRESENT
	Power off	1st– ACTIVE	EXTRACTION_PENDING
		2nd– EXTRACTION_PENDING	INACTIVE
	Power on	1st– INACTIVE	INSERTION_PENDING
		2nd– INSERTION_PENDING	ACTIVE
OA	Insertion	NOT_PRESENT	ACTIVE
Fan	Extraction	ACTIVE	NOT_PRESENT
Power Supply			

Configuration

The *saHpiParmControl()* API is not supported in the OA SOAP plug-in. Therefore, the *saHpiParmControl()* API returns the following:

`SA_ERR_HPI_UNSUPPORTED_API`

Load Management

Load Management related APIs are not supported in the OA SOAP plug-in. Therefore, all Load Management related APIs always return `SA_ERR_HPI_UNSUPPORTED_API`.

Reset Management

The following list provides the status of all power management related APIs.

- *saHpiResourceResetStateGet()* API returns the current reset state of the given resource.
- *saHpiResourceResetStateSet ()* API functions return the following:
  - `SAHPI_RESET_ASSERT` request on the given resource will Power-Off the resource
  - `SAHPI_RESET_DEASSERT` request on the given resource will Power-On the resource.
  - Reset Management returns `INVALID_REQUEST` if the cold/warm reset is requested on a resource that is powered off.

Power Management

The status of all power management related APIs is as follows:

- *saHpiResourcePowerStateGet()* API returns the current power state of the given resource.
- *saHpiResourcePowerStateSet()* API functions are as follows:
  - `SAHPI_POWER_ON` request on the given resource will Power-On the resource if it is in Power-Off state.
  - `SAHPI_POWER_OFF` request on the given resource will Power-Off the resource if it is in Power-On state.
  - `SAHPI_POWER_CYCLE` request on the given resource will Power-Off and power-on the resource if it is in Power-On state.
  - `SAHPI_POWER_CYCLE` request on the given resource will Power-On the resource if it is in Power-Off state.

Alarms, Events, and Event Log Management

The OpenHPI OA SOAP plug-in retrieves the hardware events from OA by using a continuous polling mechanism. When the OA SOAP plug-in starts, the plug-in makes a request to OA for hardware events and OA immediately starts discovering the hardware resources and buffering events in to memory. When the OA SOAP plug-in completes the hardware resources discover, the plug-in begins continuous polling of hardware events. OA SOAP plug-in processes the newly retrieved events and converts some of them into HPI events and pushes them into the event-processing queue of the OpenHPI framework.

Event-related APIs and Alarm-related APIs work on the Domain Alarm Table and the Domain Event Log. Both of these are supported by the OpenHPI framework.

The HP BladeSystem c-Class does not allow alteration of the events log, subsequently, the Event Log Management APIs are not supported in the OA SOAP plug-in. However, they are supported by the OpenHPI framework and their operations are limited only to the Domain Event Log level.

OpenHPI OA SOAP Plug-in Limitations and Known Issues

The following is a list of limitations and known issues associated with the OpenHPI OA SOAP plug-in:

- The OpenHPI OA SOAP plug-in does not support cascaded HP BladeSystem c-Class enclosures.
- The OpenHPI OA SOAP plug-in does not support setting the `AutoInsert` or `AutoExtract` timeouts.
- The OpenHPI OA SOAP plug-in does not support FUMI, DIMI and Load Management APIs.

- The OpenHPI OA SOAP plug-in has only been tested with HP BladeSystem c-Class enclosures equipped with HP Proliant Blades.
- The OA Switchover requires a maximum of 180 seconds. During this time, the OpenHPI OA SOAP plug-in is not able to communicate with Active OA. During this OA Switchover, any HPI API call which requires communication with OA will fail.
- When the OA is inserted into the slot, it usually takes 40 seconds to stabilize. If Active OA is removed prior to stabilization of the newly inserted Standby OA, the OA SOAP plug-in will hang.
- If the OA SOAP plug-in is started with only one OA or if the Standby OA is removed before the OA SOAP plug-in is initialized, the OA SOAP plug-in does not recognize the Standby OA. Thereafter, if Active OA is removed and Standby OA is inserted, the OA SOAP plug-in does not recognize the Standby OA. The OA SOAP plug-in hangs until the previous Active OA is re-inserted and becomes Active again.
- The OA handles power management for all resources in the HP BladeSystem c-Class enclosure.

  If the OA detects a hardware or an overheat failure of any server blade, it powers off that blade, and then the real power-on command on the blade does not execute. The `saHpiResourcePowerStateSet` and `saHpiResourceResetStateSet` APIs on that blade will return an erroneous `SUCCESS` value.