Hewlett Packard
Enterprise

HPE OneView to HPI Mapping Developer's Guide

Developer's Guide

Published: January 2017
Edition: 1.0

Hewlett Packard
Enterprise

Contents

Hewlett Packard
Enterprise

List of figures

List of tables

Introduction

HPE OneView(OV) is powerful converged management that reduces infrastructure complexity with automation simplicity. Its software-defined approach can help your IT teams capture their best practices for "get it right" repeatability every time. HPE OneView supports lights-out automation and provides a simple, fast, and efficient path to Infrastructure-as-a-Service, allowing you to get to a hybrid cloud. Synergy Composer is the management module that resides within the HPE Synergy Frameand can be paired with other tools to simplify daily tasks, warn of potential issues, and assist with repairs. Synergy Composer provides a REST/JSON interface for managing the HPE OneView & Synergy Frame.

Hewlett Packard
Enterprise

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, offering access to various types of hardware.

The OpenHPI ov_rest plug-in enables HPI support for HPE OneView. The OpenHPI ov_rest plug-in supports Out-of-Band Management, which allows it to run on any server inside or outside the Synergy Ring. The HPI application may run one or more instances of the OpenHPI ov_rest plug-in in parallel with the plug-ins and communicates with the OneView using the REST/JSON interface. The plug-in discovers the HPE OneView hardware resources and then populates OpenHPI data structures. The OpenHPI ov_rest plug-in then retrieves the hardware events asynchronously and converts them into OpenHPI events.

In the OpenHPI source tree, this plug-in is called ov_rest and is referenced by the name libov_rest 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 HPE OneView.

Additional Resources

For more information about the Synergy Composer, including the HPE OneView User Guide, see the following website:
https://www.hpe.com/us/en/integrated-systems/synergy.html
https://www.hpe.com/us/en/integrated-systems/software.html
http://www.openhpi.org/Downloads
https://sourceforge.net/projects/openhpi/files/openhpi-stable/

Hewlett Packard
Enterprise

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.

HPE Encourages Your Comments

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

Obtaining the ov_rest Plug-in

The ov_rest plug-in is included in OpenHPI version 3.7.1 and later. The OpenHPI source can be downloaded from the OpenHPI website located at:
http://www.openhpi.org/Downloads

Hewlett Packard
Enterprise

Building the OpenHPI Source

The ov_rest plug-in is 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:

To disable the ov_rest plug-in build:
--disable-ov_rest

The ov_rest plug-in requires that the libcurl-devel, json-c-devel, and librabbitmq-devel packages are installed in order to build successfully. HPE 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 make the RPMs for later installation, verify that you have root privleges, and then enter the following command:
make rpm

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

Configuring the OneView

HPE OneView runs on Synergy composer. You must set up a user account in the Synergy composer for the HPE Synergy Frame Ring of enclosures you want to manage. To setup or change the login and/or password, refer to the HPE Synergy Composer Administrator User Guide. The user account for the plug-in on the Synergy Composer must have administrator-level or operator-level privileges.

OpenHPI ov_rest Plug-in Configuration File

The HPE OneView is the management interface for the Synergy composer that manages a ring of Synergy Frames. The HPE Synergy Frame Ring can have one or more Synergy composers: one active, one standby, and one or more composers in the backup. One floating IP address points to the active composer. When a failover happens and the standby composer becomes active, the floating IP address points to the current active composer. The failover is therefore transparent to the user.Therefore, for any given scenario only one IP address exists to communicate with Synergy composers. The OpenHPI ov_rest plug-in is configured in the openhpi.conf file located in the /etc/openhpi directory. You can configure one or more ov_rest plug-in instances along with other plug-ins in the openhpi.conf file.

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

Table 1: OpenHPI ov_rest Plug-in Configuration Details.

PARAMETER	**DESCRIPTION**
ENTITY_ROOT	Indicates the entity root of the entity path. The root path for the discovered resources is generated by adding the prefix entity_root to the location of the resource in the chassis.The active Synergy composer points to this and the ENTITY_ROOT prefix becomes the root of the entity path for all the other resources in the frame. For example, {RACK, 3} where the RACK corresponds to the synergy composer specified in the ACTIVE_OA IP address below.
OV_USER_NAME	Holds the OV user name. It is used for authenticating with OV.

OV_PASSWORD	Holds the OV password. It is used for authenticating with OV.
ACTIVE_OV	Holds the Active OV IP address.

HPE Synergy Frame Resources

The HPE Synergy Frame contains the following hardware resources:

- Synergy Composers—Two slots for each enclosure but only two Synergy composers filled up for entire ring
- Server Blades—Twelve half-height blades or six full-height blades or a combination of both
- Interconnects—Six single-wide interconnects or two double-wide interconnects or a combination of both
- Synergy Frame Link Modules (Link Modules)—Two redundant link modules
- Fans—Ten fans
- Power Subsystem—One power subsystem with six power supplies
- Storage or IO Blades—Double-wide, half-height blades

HPE OneView Supported Hardware Resources

- Synergy 480 Gen9
- Synergy 660 Gen9
- Virtual Connect SE 40Gb F8 Module for Synergy
- Synergy D3940 Storage Module

NOTE: The HPE ring of Synergy Frames is managed by one OneView instance. The following illustration shows ones such ring.

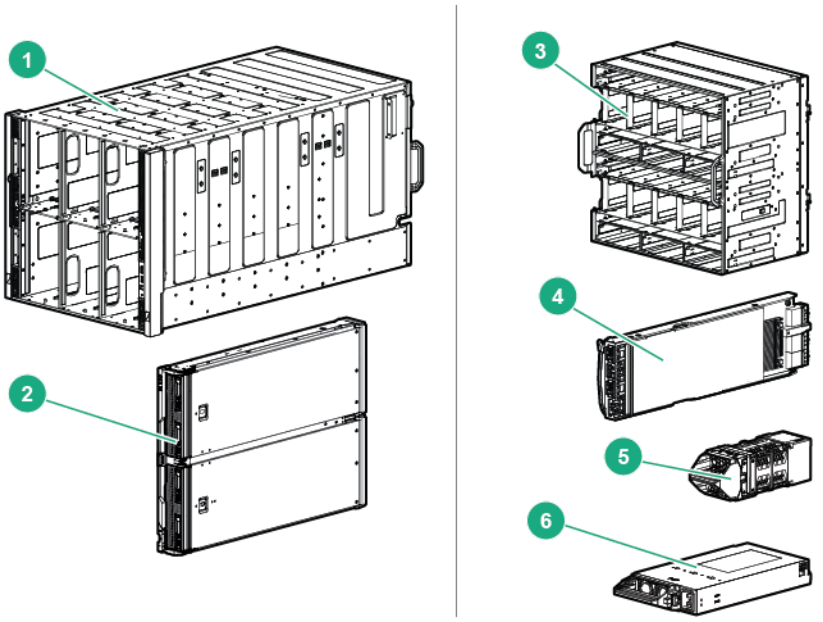

Figure 1: HPE Synergy Frame Hardware Resources

Item	Description
1	Synergy 12000 Frame
2	Server blade
3	Rear of Synergy 12000 Frame
4	Storage blade
5	Fan
6	Power supply

Resource Presence Table (RPT) Mapping

RPT mapping of the HPE Synergy Frame resources to the HPI resources is specified in Table 2.

Table 2: Resource Mapping.

HPE Synergy Frame System Resource	HPI Resource
SYNERGY COMPOSER	RACK
SYNERGY FRAME	SYSTEM_CHASSIS
SERVER BLADE	SYSTEM_BLADE
STORAGE BLADE	DISK_BLADE
VIRTUAL CONNECT OR INTERCONNECT	SWITCH_BLADE
FAN	
	FAN
POWER SUPPLY	POWER_SUPPLY

The HPE Synergy Framecontains the server blades, interconnects, OVs, fans, and power supplies. Therefore, the entity paths for the HPE Synergy Frame resources are as provided in Table3.

Table 3: Resource Entity Path

Area Type	Supported Field Types
SYNERGY COMPOSER	{RACK:0}{RACK, Composer Number}
SYNERGY FRAME	{RACK:0}{SYSTEM_CHASSIS, Enclosure Number}
SERVER BLADE	{RACK:0}{SYSTEM_CHASSIS, Enclosure Number} {SYSTEM_BLADE, Blade Slot Number}
STORAGE BLADE	{RACK:0}{SYSTEM_CHASSIS, Enclosure Number} {DISK_BLADE, Blade Slot Number}
INTERCONNECT	{RACK:0}{SYSTEM_CHASSIS, Enclosure Number} {SWITCH_BLADE, Interconnect Slot Number}
VIRTUAL CONNECT	{RACK:0}{SYSTEM_CHASSIS, Enclosure Number} {SWITCH_BLADE, Interconnect Slot Number}
FAN	{RACK:0}{SYSTEM_CHASSIS, Enclosure Number} {FAN, Fan Slot number }
POWER SUPPLY	{RACK:0}{SYSTEM_CHASSIS, Enclosure Number} {POWER_SUPPLY, Power Supply Slot Number}

The supported resource capabilities for HPE Synergy Frame resources are specified in Table 4.

Table 4: Resource Capability

HPE Synergy Frame System Resource	HPI Resource Capability
SYNERGY COMPOSER	CONTROL, INVENTORY_DATA, RDR, RESOURCE, SENSOR
SYNERGY FRAME	RESOURCE, RDR, INVENTORY_DATA, SENSOR, CONTROL
SERVER BLADE	RESOURCE, RDR, INVENTORY_DATA, SENSOR ,FRU, MANAGED_HOTSWAP, POWER, RESET, CONTROL
STORAGE BLADE	CONTROL, FRU, INVENTORY_DATA, MANAGED_HOTSWAP, RDR, RESET, RESOURCE, SENSOR
INTERCONNECT VIRTUAL CONNECT	RESOURCE, RDR, INVENTORY_DATA,, SENSOR ,FRU, MANAGED_HOTSWAP, POWER, RESET, CONTROL
FAN	RESOURCE, RDR, INVENTORY_DATA, SENSOR ,FRU
POWER SUPPLY	RESOURCE, RDR, INVENTORY_DATA, SENSOR ,FRU

RDR Mapping

There are a few general points that are applicable for all HPE Synergy Frame Resource Data Records (RDRs). These points are detailed in the following list:

- Power controls are supported only on Server Blades and Interconnects (Switches).
- Control mode support for Power Controls and UID Controls 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 for Power Controls and UID LED controls are not supported by the resource with control capability due to a hardware limitation in supporting the transitory states for power and UID LED.
- All sensors in the Synergy Frame are only of the data type SAHPI_SENSOR_READING_TYPE_FLOAT64.
- 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.

HPE Synergy FrameRDRs

Table 5 and Table 6 detail the HPE Synergy FrameRDRs.

Table 5: HPE Synergy Frame Control RDRs

Control Name	Control Number	Control Type	Control Output Type	Default Mode	Supported Values
UID LED STATE	OV_REST_UID_CNTRL	DIGITAL	LED	MANUAL	SAHPI_CTRL_STATE_OFF SAHPI_CTRL_STATE_ON

Table 6: HPE Synergy Frame Inventory RDRs

AREA TYPE	SUPPORTED FIELD TYPES
PRODUCT_INFO	PRODUCT_NAME, MANUFACTURER,

	CUSTOM (URI)
CHASSIS_INFO	PART_NUMBER, SERIAL_NUMBER

Synergy Composer RDRs

Table 7 and Table 8 detail the Synergy Composer RDRs.

Table 7: Synergy Composer Control RDRs

Control Name	Control Number	Control Type	Control Output Type	Default Mode	Supported Values
UID LED STATE	OV_REST_UID_CNTRL	DIGITAL	LED	MANUAL	SAHPI_CTRL_STATE_OFF SAHPI_CTRL_STATE_ON

Table 8: Synergy Composer Inventory RDRs

Area Type	Supported Field Types
PRODUCT_INFO	PRODUCT_NAME, MANUFACTURER, PRODUCT_VERSION
CHASSIS_INFO	SERIAL_NUMBER

Server Blade, Disk, and IO RDRs

Table 9, Table 10, and Table 11 detail the server blade, disk, and IO blade RDRs.

Table 9: Server Blade Control RDRs

Control Name	Control Number	Control Type	Control Output Type	Default Mode	Supported Values
POWER STATE	OV_REST_PWR_CNTRL	DIGITAL	POWER_STATE	MANUAL	SAHPI_CTRL_STATE_OFF SAHPI_CTRL_STATE_ON
UID LED STATE	OV_REST_UID_CNTRL	DIGITAL	LED	MANUAL	SAHPI_CTRL_STATE_OFF SAHPI_CTRL_STATE_ON

Table 10: Disk and IO Blade Control RDRs

Control Name	Control Number	Control Type	Control Output Type	Default Mode	Supported Values
POWER STATE	OV_REST_PWR_CNTRL	DIGITAL	POWER_STATE	MANUAL	SAHPI_CTRL_STATE_OFF SAHPI_CTRL_STATE_ON
UID LED STATE	OV_REST_UID_CNTRL	DIGITAL	LED	MANUAL	SAHPI_CTRL_STATE_OFF SAHPI_CTRL_STATE_ON

Table 11: Server Blade, Disk, and IO Blade Inventory RDRs

Area Type	Supported Field Types
PRODUCT_INFO	PRODUCT_NAME, MANUFACTURER,

	PRODUCT_VERSION, CUSTOM (URI)
BOARD_INFO	PART_NUMBER, SERIAL_NUMBER

Interconnect RDRs

Table 12 and Table 13 detail the interconnect RDRs

Table 12: Interconnect Control RDRs

Control Name	Control Number	Control Type	Control Output Type	Default Mode	Supported Values
POWER STATE	OV_REST_PWR_CNTRL	DIGITAL	POWER_STATE	MANUAL	SAHPI_CTRL_STATE_OFF SAHPI_CTRL_STATE_ON
UID LED STATE	OV_REST_UID_CNTRL	DIGITAL	LED	MANUAL	SAHPI_CTRL_STATE_OFF SAHPI_CTRL_STATE_ON

Table 13: Interconnect Inventory RDRs

Area Type	Supported Field Types
PRODUCT_INFO	PRODUCT_NAME, MANUFACTURER, CUSTOM (URI)
BOARD_INFO	PART_NUMBER, SERIAL_NUMBER

Table 14: Fan Inventory RDRs

Area Type	Supported Field Types
PRODUCT_INFO	PRODUCT_NAME
BOARD_INFO	PART_NUMBER, SERIAL_NUMBER

.

Table 15: Power Supply Inventory RDRs

Area Type	Supported Field Types
PRODUCT_INFO	PRODUCT_NAME
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 ov_rest plug-in. It discovers HPE Synergy Frames and its hardware resources and populates the RPT in the OpenHPI framework. The RPT table-related APIs are supported by the OpenHPI framework.

Controls

Table 16 provides a list of all control-related APIs and their functions.

Table 16: 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 17 provides a list of all inventory data repository-related APIs and their functions.

Table 17: Inventory Data 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 ov_rest plug-in. Therefore, all Watchdog timer-related APIs return the following message/code:

SA_ERR_HPI_UNSUPPORTED_API

Annunciators

The annunciator-related APIs are not supported in the ov_rest plug-in. Therefore, all annunciator-related APIs return the following message/code:

SA_ERR_HPI_UNSUPPORTED_API

Diagnostics Initiator Management Instrument (DIMI)

DIMI-related APIs are not supported in the ov_rest plug-in. Therefore, all DIMI-related APIs return the following message/code:

SA_ERR_HPI_UNSUPPORTED_API

Firmware Initiator Management Instrument (FUMI)

FUMI-related APIs are not supported in the ov_rest plug-in. Therefore, all FUMI-related API's return the following message/code:

SA_ERR_HPI_UNSUPPORTED_API

Hot Swap Operations

HPE Synergy Frame FRUs currently implement the HPI Unmanaged and Managed Hot Swap Models.

Unmanaged Hot Swap Model

The HPE Synergy Frame supports the HPI Unmanaged Hot Swap model for the Synergy Composer, drive enclosure, fan, and power supply FRUs Therefore, the Hot Swap APIs are not supported for these resources. These FRUs do generate Hot Swap Events.

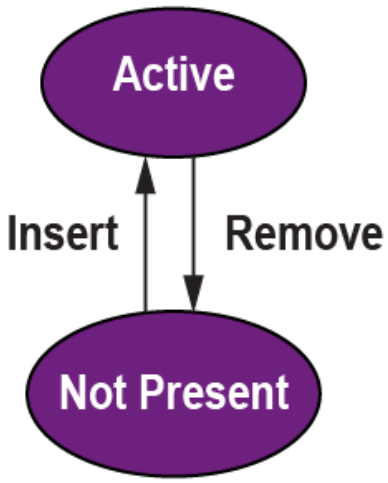

Figure 2: Unmanaged Hot Swap Model

Managed Hot Swap Model

The HPE Synergy System supports the HPI Five State Managed Hot Swap Model for server blade, interconnect and virtual connect FRUs. The ov_rest 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 State respectively.

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

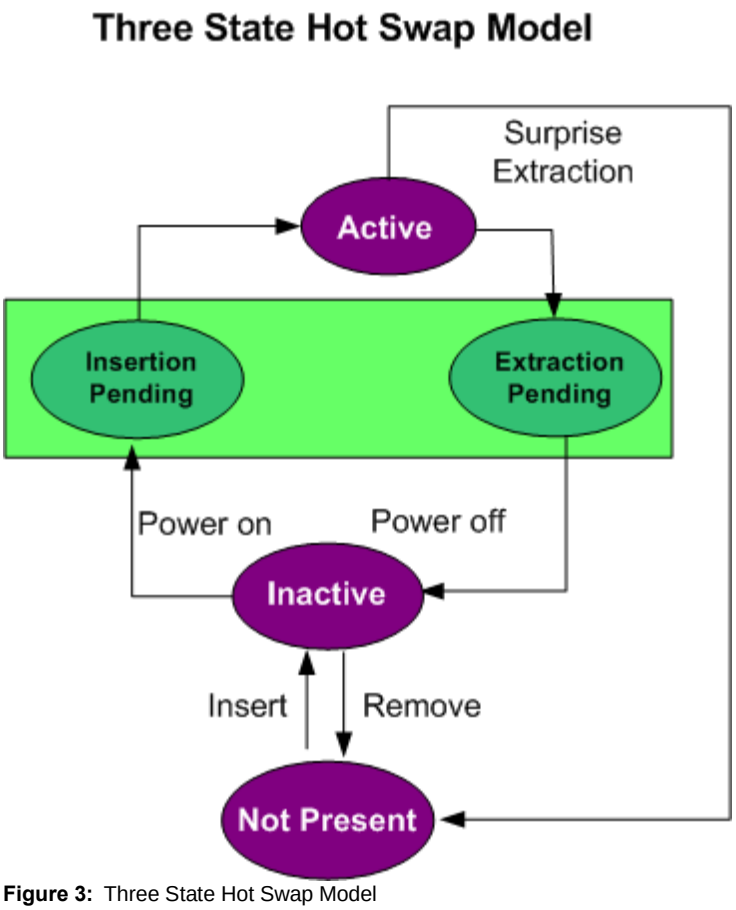

Figure 3: Three State Hot Swap Model

Behavior of HotSwap API

- 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 HPE Synergy Frame System 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 18 provides a list of resources and hot swap events triggered by particular actions.

Table 18: Hot Swap Events

Resource Name	Action	Hot Swap Events		Event Severity
		Previous State	Current State	
SERVER BLADE, INTERCONNECT BLADE (SWITCH)	Insertion	NOT_PRESENT	INSERTION_PENDING	INFORMATIONAL
	Power on after insertion	INSERTION_PENDING	ACTIVE	Resource Severity in RPT Entry
				INFORMATIONAL
	Extraction on Power On state	ACTIVE	NOT_PRESENT	
				INFORMATIONAL
	Extraction on Power On state	INACTIVE	NOT_PRESENT	
				INFORMATIONAL
	Power off	1st– ACTIVE	EXTRACTION_PENDING	INFORMATIONAL
		2nd– EXTRACTION_PENDING	INACTIVE	INFORMATIONAL
	Power on	1st– ACTIVE	INSERTION_PENDING	INFORMATIONAL
		2nd– EXTRACTION_PENDING	ACTIVE	
IO BLADE, OA, FAN, STORAGE BLADE, POWER SUPPLY	Insertion	NOT_PRESENT	ACTIVE	INFORMATIONAL
	Extraction	ACTIVE	NOT_PRESENT	Resource Severity in RPT Entry

Configuration

The saHpiParmControl() API is not supported in the ov_rest plug-in. Therefore, the saHpiParmControl() API returns the following messae/code:

SA_ERR_HPI_UNSUPPORTED_API

Load Management

Load management-related APIs are not supported in the ov_rest plug-in. Therefore, all load management-related APIs return the following message/code:

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 ov_rest plug-in retrieves the hardware events from Synergy Composer by using the State Change Message Buss (SCMB) mechanism. When the ov_rest plug-in starts, the plug-in makes a request to Synergy Composer for Locked and Active hardware events and immediately starts discovering the hardware resources and buffering events in to memory. When the ov_rest plug-in finishes discovering the hardware resources, the plug-in begins listening to SCMB bus and processes the events and alerts asynchronously. The ov_rest plug-in processes the newly retrieved events and converts some of them into HPI events, pushing them into the event processing queue of the OpenHPI framework.

Many of the Locked and Active Alerts/Events are handled as OEM events because they do not have supported sensor numbers at this time. In addition, the event description is too big to push into the event log since it supports only 255 charactors; therefore, only part of the description is pushed into the event log. The complete event is pushed into a file (/var/lib/openhpi/ov_rest/oem_events.log) along with a full decription and corrective action.

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 HPE Synergy Frame System does not allow alteration of the events log; subsequently, the Event Log Management APIs are not supported in the ov_rest plug-in. However, they are supported by the OpenHPI framework, and their operations are limited only to the Domain Event Log Level.

OpenHPI ov_rest Plug-in Limitations and Known Issues

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

- The OpenHPI ov_rest plug-in does not support setting the AutoInsertor AutoExtract timeouts.
- The OpenHPI ov_rest plug-in does not support FUMI, DIMI, and load management APIs.
- This OpenHPI ov_rest plug-in is not tested with scenarios such as Synergy Composer Switchover/Failover.

Appendix A

The RDR numbers used in the ov_rest plug-in for Sensors, Controls, and Inventory RDR types are listed in the SaHpiOvRest.h field that is provided in the standard include directory of OpenHPI. This file can be referenced for RDR numbers by the application that intends to use the ov_rest plug-in.
The SaHpiOvRest.h file contains the following lines:

```
#ifndef __SAHPIOVREST_H
#define __SAHPIOVREST_H
```

```
        /* UID control */
        #define OV_REST_UID_CNTRL                   (SaHpiCtrlNumT)   0x000
        /* Power control */
        #define OV_REST_PWR_CNTRL                    (SaHpiCtrlNumT)   0x001
/* Operational status sensor */
#define OV_REST_SEN_OPER_STATUS              (SaHpiSensorNumT) 0x000

#endif  //__SAHPIOVREST_H
```