OpenHPI for HP ProLiant Rack Mount Server
Developers Guide

Published: August 2012
Edition: 4.0

Contents

Tables

Introduction

The OpenHPI Developer's Guide for the HP ProLiant Rack Mount Server describes how to install and configure the iLO2 RIBCL plug-in and provides a product overview, and information on resource mappings and event processing.

Intended Audience

This document is intended for application developers, programmers, and database administrators who are responsible for developing, testing, administering and maintaining OpenHPI enablement on HP ProLiant Rack Mount servers using the iLO2 RIBCL plug-in.

Additional Resources

For more information about iLO, see the *HP Integrated Lights-Out 2 User Guide* located at:

http://www.hp.com/servers/lights-out

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.

Overview

OpenHPI provides an open source implementation of the Service Availability Forum (SAF) Hardware Platform Interface (HPI). HPI provides an interface for managing computer hardware like the HP ProLiant Rack Mount Server. The HP ProLiant Rack Mount Server requires an OpenHPI “plug-in” to support OpenHPI on its hardware. HP has developed the iLO2 RIBCL plug-in to allow access to the HP ProLiant Rack Mount Server. The iLO2 RIBCL plug-in works with iLO2, iLO3 and later versions.

The iLO2 RIBCL plug-in is an interface to HP ProLiant Rack Mount servers, which use Remote Insight Board Command Language (RIBCL) to communicate with the on-board Integrated Lights-Out 2 (iLO2) management processor.

NOTE: The iLO2 RIBCL plug-in is referenced as ilo2_ribcl in the OpenHPI tree, and by the name libilo2_ribcl in all OpenHPI configuration files.

For more information and documentation on the OpenHPI project, refer to the following web sites:

- http://www.openhpi.org
- http://sourceforge.net/projects/openhpi
- http://openhpi.sourceforge.net/manual/book1.html

Supported Systems

The iLO2 RIBCL plug-in is supported on the following HP ProLiant Rack Mount Servers:

- DL360 G5, G6, G7, G8
- DL365 G1, G5, G6
- DL380 G5, G6, G7, G8
- DL385 G2, G5, G6, G7

Unsupported Systems

The iLO2 RIBCL plug-in is not supported on the DL580 and DL585 HP ProLiant Rack Mount Servers.

OpenHPI Releases

The iLO2 RIBCL plug-in is included in the OpenHPI 2.12 release. For earlier OpenHPI releases, the iLO2 RIBCL plug-in patches can be downloaded.

Future enhancements and defect fixes for this plug-in are posted to the openhpi-devel mailing list located at:

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

You can monitor this list or perform a search for the string *ilo2_ribcl* to find relevant content in the mailing list archives.

System Requirements

The iLO2 RIBCL plug-in is installed and enabled automatically when you compile and install OpenHPI. Instructions are provided in the `openhpi/README` file. You may run into errors during the OpenHPI configuration and build process if your build system does not contain the necessary software packages.

The following section describes what you need to successfully build OpenHPI with the iLO2 RIBCL plug-in and contains information on installation and firmware requirements.

Installation Requirements

To successfully install the iLO2 RIBCL plug-in during the OpenHPI build process, you must have the following packages installed on your Management Server (host system):

- `openssl-devel` version 0.9.8a or later
- `libxml2-devel` version 2.6.23 or later

HP recommends that you use the latest version that is available for your distribution.

NOTE: You may disable the automatic build of the iLO2 RIBCL plug-in by passing the configure flag `--disable-ilo2_ribcl` during the configure process. The configuration file `./configure -disable-ilo2_ribcl` builds OpenHPI without the iLO2 RIBCL plug-in. For more information, see the `README` file that is packaged with the OpenHPI distribution.

Firmware Requirements

Verify that iLO2, iLO3 and iLO4 has the latest firmware on all target HP ProLiant Rack Mount server systems to be managed. The Lights-Out advanced pack licensing level is required to access advanced manageability features such as Power Management.

For detailed instructions on performing the firmware upgrade, see the respective *HP Integrated Lights-Out User Guide* located at:

http://www.hp.com/servers/lights-out

Configuring the iLO2 RIBCL Plug-In

The OpenHPI iLO2 RIBCL plug-in is configured in the `/etc/openhpi/openhpi.conf`. OpenHPI configuration file. Using your preferred text editor, edit the file and configure the iLO2 RIBCL plug-in instance.

This section contains information on the following topics:

- Setting the iLO Username and Password
- Modifying the iLO2 RIBCL Plug-In section in the OpenHPI Configuration File

Setting the iLO User Name and Password

You must set up a user account on the iLO management processor for each HP ProLiant Rack Mount server that you want to manage. The iLO is configured at the factory with a default username and password. The default username and password can be found on the iLO2 Network Settings tag attached to the server. HP recommends changing the default username and password after logging in to iLO for the first time.

The login and password in the `/etc/openhpi/openhpi.conf` configuration file must correspond to an iLO local user name and password. In addition, to support any HPI API that involves power control or reset, verify the iLO local user name and password has the *Virtual Power and Reset* privilege.

To setup or change the login and password, refer to the "iLO Setup" and "Configuring iLO2" chapters in the respective *HP Integrated Lights-Out User Guide* located at:

http://www.hp.com/servers/lights-out

Modifying the iLO2 RIBCL Plug-In Section in the OpenHPI Configuration File

The iLO2 RIBCL plug-in supports several interfaces for configuration. The following is an example configuration stanza for the iLO2 RIBCL plug-in:

```
handler libilo2_ribcl {
    entity_root = "{RACK_MOUNTED_SERVER,8}"
    ilo_ribcl_hostname = "x.x.x.x" # <iLO IP address>
    ilo_ribcl_portstr = "443" # iLO2 RIBCL SSL server
port number
    ilo_ribcl_username = "username" # <iLO username>
    ilo_ribcl_password = "password" # <iLO password>
}
```

You must update the entries in the `/etc/openhpi/openhpi.conf` configuration file for your particular configuration as follows:

- The value of `ilo_ribcl_hostname` must be set to the TCP/IP address of the iLO on the system you want to manage.
- The value of `ilo_ribcl_portstr` specifies the Web Server SSL Port used by the iLO2 RIBCL plug-in. The default configuration value is 443. Change this value only if you have changed the Web Server SSL Port configuration in iLO2.
- The value of `ilo_ribcl_username` and `ilo_ribcl_password` must correspond to an iLO local username and password.

HP ProLiant Server Resources

This section covers the following topics:

- HP ProLiant Server Model Differences
- RPT Mapping and HP ProLiant Server Entity Types
- System Chassis Resources
- Processor Resources
- Memory Resources
- Power Supply Resources
- Power Module (VRM) Resources
- Fan Resources

All removable resources use the simple hot swap model, and specify `SA_CAPABILITY_FRU` in the `ResourceCapabilities` of their Resource Presence Table entry. The System Chassis is the only resource that is not a field replaceable unit. For details on how hot swap works with HP ProLiant servers, see "Hot Swap Operations" (page 16).

The `ResourceSeverity` field in the Resource Presence Table entry for all ProLiant resources is set to `SAHPI_CRITICAL`.

RPT Mapping and HP ProLiant Server Entity Types

The following tables outline the RPT mappings, entity type definitions, and entity path examples supported by the iLO2 RIBCL plug-in.

Table 1 RPT Mappings

Resource	Entity Path	Capabilities
HP ProLiant Rack Mount Chassis	{entity_root} {RACK_MOUNTED_SERVER,#}	SA_CAPABILITY_CONTROL SA_CAPABILITY_INVENTORY_DATA SA_CAPABILITY_POWER SA_CAPABILITY_RDR SA_CAPABILITY_RESET SA_CAPABILITY_RESOURCE SA_CAPABILITY_SENSO
Processor	{RACK_MOUNTED_SERVER,#} {PROCESSOR,#}	SA_CAPABILITY_FRU SA_CAPABILITY_INVENTORY_DATA SA_CAPABILITY_RDR SA_CAPABILITY_RESOURCE

Table 1 RPT Mappings *(continued)*

Resource	Entity Path	Capabilities
Memory Device	{RACK_MOUNTED_SERVER,#} {MEMORY_DEVICE,#}	SA_CAPABILITY_FRU SA_CAPABILITY_INVENTORY_DATA SA_CAPABILITY_RDR SA_CAPABILITY_RESOURCE
Power Supply	{RACK_MOUNTED_SERVER,#} {POWER_SUPPLY,#}	SA_CAPABILITY_FRU SA_CAPABILITY_RESOURCE
Power Module	{RACK_MOUNTED_SERVER,#} {POWER_MODULE,#}	SA_CAPABILITY_FRU SA_CAPABILITY_RESOURCE

Table 2 HP ProLiant Server Entity Type Definitions

Entity Name	Entity Type
ProLiant Rack Mount Server Chassis	RACK_MOUNTED_SERVER
Processor	PROCESSOR
Memory Module	MEMORY_DEVICE
Power Supply Module	POWER_SUPPLY
Power Module	POWER_MODULE
Fan	COOLING_DEVICE

Table 3 Entity Path Examples

Resources	Example Entity Path
ProLiant Rack Mount Server Chassis	{RACK_MOUNTED_SERVER,9}
Processor(s)	{RACK_MOUNTED_SERVER,9}{PROCESSOR,1} {RACK_MOUNTED_SERVER,9}{PROCESSOR,2}
Memory Module	{RACK_MOUNTED_SERVER,9}{MEMORY_DEVICE,1} {RACK_MOUNTED_SERVER,9}{MEMORY_DEVICE,2}
Power Module	{RACK_MOUNTED_SERVER,9}{POWER_MODULE,1} {RACK_MOUNTED_SERVER,9}{POWER_MODULE,2}
Fan(s)	{RACK_MOUNTED_SERVER,9}{COOLING_DEVICE,1} {RACK_MOUNTED_SERVER,9}{COOLING_DEVICE,2})

System Chassis Resources

The entity location for the system chassis is specified in the `openhpi.conf` configuration file as the entity location for the `entity_root`.

The resource tag for the system chassis is constructed by combining the system model number with the system serial number and the entity location of the `entity_root` enclosed within parenthesis. For example, the following syntax specifies a resource tag for the DL360 G5 server with the number `5` as the entity location.

Example Syntax

```
ProLiant DL360 G5 SN:MXQ73703A6 (5)
```

Processor Resources

The entity location for a processor resource is the slot number for that processor given by the iLO2 RIBCL plug-in.

The resource tag for a processor resource is the label value returned from the RIBCL GET_HOST_DATA command. It is displayed as Proc *N*, where *N* is the processor slot number.

Memory Resources

The entity location for a memory DIMM resource is the slot number for that DIMM given by the iLO2 RIBCL plug-in.

The resource tag for a memory DIMM resource is the label value returned from the RIBCL GET_HOST_DATA command. It is displayed as DIMM *NX*, where *N* is the slot number and *X* is a capital letter as shown in the following example syntax.

Example Syntax

```
DIMM 2A
```

Future supported systems may have a different format for the DIMM resource tag.

Power Supply Resources

The entity location for the power supply resource is the bay number for that power supply given by the iLO2 RIBCL plug-in.

The resource tag for a power supply resource is the label value returned for that power supply from the RIBCL GET_EMBEDDED_HEALTH_DATA command. It is displayed as Power Supply *N*, where *N* is the power supply

Power Module (VRM) Resources

The entity location for a power module resource is the VRM slot number for that module given by the iLO2 RIBCL plug-in. The resource tag for a power module resource is the label value returned from the GET_EMBEDDED_HEALTH_DATA command. It is displayed as VRM *N*, where *N* is the slot number.

Fan Resources

The entity location for a fan resource is the fan index number given by the iLO2 RIBCL plug-in. Note, for the DL360 server and the DL365 server, the index number specifies a block of several fans.

The resource tag for a fan resource is constructed by combining the label and zone values for the fan that are returned by the GET_EMBEDDED_HEALTH_DATA command.

Example Syntax

For DL380 and DL385 Servers:

```
Fan 2 Location I/O Board
```

Example Syntax

For DL360 and DL365 Servers:

```
Fan Block 2 Location CPU
```

The DL360 and DL 365 Servers report information for block fans.

Resource RDR Mappings

All inventory data repositories, areas, and fields are read-only. The supported HPI inventory APIs include the following:

- saHpiIdrInfoGet()
- saHpiIdrAreaHeaderGet()
- saHpiIdrFieldGet()

Sensor readings are updated only during a discovery operation due to the overhead of the iLo2 communication latency.

Sensor readings are also updated with the periodic resource discovery performed by the openhpid daemon's `oh_discovery_thread_loop`, and occur approximately every three minutes. The sensor reading returned from `saHpiSensorReadingGet()` is the cached value obtained during the most recent discovery operation.

This section contains information on the following topics:

- HP ProLiant Chassis RDRs
- Processor RDRs
- Memory RDRs
- Power Supply RDRs
- Fan (Cooling Device) RDRs
- Voltage Regulator Module (VRM) RDRs
- Sample hpitop Output for Supported Servers

HP ProLiant Chassis RDRs

Chassis Controls

Table 4 identifies the chassis controls available and supported by the iLO2 RIBCL plug-in. Table 6 on page 15 provides a summary of the chassis controls that are available and supported by the iLO2 RIBCL plug-in.

Table 4 Supported Controls

Control	Description
Unit Identification Light (UID) Control	Unit Identification Light status can be queried using the `saHpiControlGet` API and the light can be turned on or off using the `saHpiControlSet` API. Valid values for the Unit Identification Light (UID) Control include: • .. On(1) • .. Off(0)
Power Saver Control	The iLO Power Regulator Feature on HP ProLiant Rack Mount servers allow various power modes for the systems to run. The current Power Regulator value can be queried on all supported HP ProLiant Rack Mount servers. Changing the Power Regulator value is fully supported on DL360 and DL380. DL365 does not support the Power Saver Set feature. DL385 supports only the HP Static Low Power Mode. The valid modes for Power Regulator Setting (Power Saver) Control include: • OS Control Mode or Disabled Mode for iLO(1) • HP Static Low Power Mode(2) • HP Dynamic Power Savings Mode(3) • HP Static High Performance Mode(4)
Auto Power Control	The *iLO Automatic Power On* and *Automatic Power On with Delay* features allow users to change the values to suit their needs. Valid values for the Automatic Power On include: • Enabled with a minimum delay(1) • Disabled(2) • Enabled with random delay up to 60 seconds(3) • Enabled with 15 seconds delay(15) • Enabled with 30 seconds delay(30)

Table 4 Supported Controls *(continued)*

Control	Description
	• Enabled with 45 seconds delay(45) • Enabled with 60 seconds delay(60) **NOTE:** The values presented are valid on iLO3 and iLO4 managed systems. However, iLO3 and iLO4 set these fields to a RANDOM value up to a maximum of 120 seconds. **NOTE:** iLO4 has an additional option called "Restore Last Power State"

Table 5 Chassis Control Summary

Control	Output Type	Type
Unit Identification Light (UID) Control	SAHPI_CTRL_LED	SAHPI_CTRL_TYPE_DIGITAL
Power Saver Control	SAHPI_CTRL_GENERIC	SAHPI_CTRL_TYPE_DISCRETE
Auto Power Control	SAHPI_CTRL_GENERIC	SAHPI_CTRL_TYPE_DISCRETE

Chassis Inventory

The system chassis IDR is read-only and includes an area called `SAHPI_IDR_AREATYPE_CHASSIS_INFO` that contains the following four read-only fields:

Table 6 Chassis Inventory Fields

Field ID	Field Type	Value
1	SAHPI_IDR_FIELDTYPE_PRODUCT_NAME	Model number
2	SAHPI_IDR_FIELDTYPE_SERIAL_NUMBER	Serial number
3	SAHPI_IDR_FIELDTYPE_MANUFACTURER	“Hewlett Packard”
4	SAHPI_IDR_FIELDTYPE_CUSTOM	iLo3 and later firmware revision

The iL0 firmware version returned by the custom Field ID 4 is expressed as:

`iLo_Firmware: <major rev>.<minor rev>`

Example Chassis IDR Output

The following is an example of the chassis IDR output that is returned from the `hpiinv` command.

```
Resource[0] Tag: ProLiant DL385 G2 SN:2UX72901KM ( 4) has inventory capability
{RACK_MOUNTED_SERVER:4} {ROOT:0}

RDR[30000]: Inventory, IdrId=0 ProLiant DL385 G2
SN:2UX72901KM ( 4) Inventory

AreaId[1] Chassis Area

  FieldId[1] Product Name : ProLiant DL385 G2

  FieldId[2] Serial Number : 2UX72901KM

  FieldId[3] Manufacturer : Hewlett Packard

  FieldId[4] Custom Field : iLo_Firmware: 1.30
```

Chassis Sensors

There are three severity type sensors located on the System Chassis resource. These sensors correspond to the system's general health, and display information given in the `HEALTH_AT_A_GLANCE` stanza returned by the `GET_EMBEDDED_HEALTH RIBCL` command.

All three sensors have the RDR's `EventCtrl` element set to `SAHPI_SEC_PER_EVENT`, which allows you to change the sensor enable, sensor event enable, and assert/deassert masks via the OpenHPI APIs. In addition, the System Chassis resource does not have the `SAHPI_CAPABILITY_DEASSERTS` capability set, so the assert and deassert masks can be asymmetrical.

Table 7 Chassis Sensors HPI Data

Sensor	Type	Category	Events
Fan Health	SAHPI_FAN	SAHPI_EC_SEVERITY	SAHPI_ES_OK SAHPI_ES_MAJOR_FROM_LESS SAHPI_ES_MAJOR_FROM_CRITICAL SAHPI_ES_CRITICAL
Temperature Health	SAHPI_TEMPERATURE,	SAHPI_EC_SEVERITY SAHPI_EC_THRESHOLD	SAHPI_ES_OK SAHPI_ES_CRITICAL
Power Supply Health	SAHPI_POWER_SUPPLY	SAHPI_EC_SEVERITY	SAHPI_ES_OK SAHPI_ES_MAJOR_FROM_LESS SAHPI_ES_MAJOR_FROM_CRITICAL SAHPI_ES_CRITICAL

Table 8 Chassis Sensor Values

Sensor	RIBCL Value	HP Generated Event	Sensor Reading
Fan Health	OK	SAHPI_ES_OK	0
	Degraded	SAHPI_ES_MAJOR_FROM_LESS SAHPI_ES_MAJOR_FROM_CRITICAL	1
	Failed	SAHPI_ES_CRITICAL	2
Temperature Health	OK	SAHPI_ES_OK	0
	Failed	SAHPI_ES_CRITICAL	2
Power Supply Health	OK	SAHPI_ES_OK	0
	Degraded	SAHPI_ES_MAJOR_FROM_LESS SAHPI_ES_MAJOR_FROM_CRITICAL	1
	Failed	SAHPI_ES_CRITICAL	2

Processor RDRs

The IDR for a processor resource is read-only and includes an area called `SAHPI_IDR_AREATYPE_BOARD_INFO` that contains the following single, read-only field:

Table 9 Processor Inventory Field

Field ID	Field Type	Value
1	SAHPI_IDR_FIELDTYPE_CUSTOM	Processor Speed

The processor speed returned by the custom Field ID 1 is the processor's rated speed. The processor speed returned by the custom Field ID 1 is expressed as:

`Speed: <speed> MHz`

Example Processor IDR Output

The following is an example of the processor IDR output that is returned by the `hpiinv` command.

```
Resource[26] Tag: Proc 1 has inventory capability
{PROCESSOR:1} {RACK_MOUNTED_SERVER:4} {ROOT:0}
RDR[30000]: Inventory, IdrId=0 Proc 1 Inventory
AreaId[1] Board Area
   FieldId[1] Custom Field : Speed: 2200 MHz
```

Memory RDRs

The IDR for a memory resource is read-only and includes an area called `SAHPI_IDR_AREATYPE_BOARD_INFO` that contains the following two, read-only fields:

Table 10 Memory Inventory Fields

Field ID	Field Type	Value
1	SAHPI_IDR_FIELDTYPE_CUSTOM	Memory Size
2	SAHPI_IDR_FIELDTYPE_CUSTOM	Memory Speed

The memory size is returned by the custom Field ID 1 and is expressed as:

`Size: <size> MB`

The memory speed is returned by the custom Field ID 2 and is expressed as:

`Size: <speed> MHz`

Example Memory IDR Output

The following is an example of the memory IDR output returned from the `hpiinv` command:

```
Resource[28] Tag: DIMM 1A has inventory capability
{MEMORY_DEVICE:1} {RACK_MOUNTED_SERVER:4} {ROOT:0}
RDR[30000]: Inventory, IdrId=0 DIMM 1A Inventory
AreaId[1] Board Area
  FieldId[1] Custom Field : Size: 1024 MB
  FieldId[2] Custom Field : Speed: 667 MHz
```

Power Supply RDRs

Currently, the power supply resource has no additional RDRs

Fan (Cooling Device)

Currently, the fan resource has no additional RDRs.

Voltage Regulator Module (VRM) RDRs

Currently, the power module (voltage regulator) resource has no additional RDRs.

Sample hpitop Output for Supported Servers

Table 11 The following is a sample hpitop output for all supported HP ProLiant Rack Mount servers.

Table 11 Sample hpitop Output

Supported Server	Output
DL380 G6	{RACK_MOUNTED_SERVER,1} | +--- {RACK_MOUNTED_SERVER,1} | |__ Control Num: 1, Type: DIGITAL, Output Type: LED, Tag: Unit Identification Light (UID) Values: On(1)/Off(0) | |__ Control Num: 2, Type: DISCRETE, Output Type: GENERIC, Tag: Power Regulator Control Power Modes: Disabled(1)/Low(2)/DynamicSavings(3)/High(4) | |__ Control Num: 3, Type: DISCRETE, Output Type: GENERIC, Tag: Auto Power Control Delay:Min.(1)/Disabled(2)/random (3)/15 sec (15)/30 sec (30)/45 sec(45)/60 sec(60) | |__ Sensor Num: 1, Type: FAN, Category: SEVERITY, Tag: System fans health indicator: Ok(0)/Degraded(1)/Failed(2) | |__ Sensor Num: 2, Type: TEMPERATURE, Category: SEVERITY, Tag: System temperature health indicator: Ok(0)/Failed(2) | |__ Sensor Num: 3, Type: POWER_SUPPLY, Category: SEVERITY, Tag: System power supply health indicator:

Table 11 Sample hpitop Output *(continued)*

Supported Server	Output

```
Ok(0)/Degraded(1)/Failed(2)
    |     |__ Sensor Num: 4, Type: TEMPERATURE, Category: THRESHOLD, Tag: Temp 1: Ambient
    |     |__ Sensor Num: 5, Type: TEMPERATURE, Category: THRESHOLD, Tag: Temp 2: CPU 1
    |     |__ Sensor Num: 6, Type: TEMPERATURE, Category: THRESHOLD, Tag: Temp 3: CPU 2
    |     |__ Sensor Num: 7, Type: TEMPERATURE, Category: THRESHOLD, Tag: Temp 4: Memory
    |     |__ Sensor Num: 8, Type: TEMPERATURE, Category: THRESHOLD, Tag: Temp 5: Memory
    |     |__ Sensor Num: 9, Type: TEMPERATURE, Category: THRESHOLD, Tag: Temp 6: Memory
    |     |__ Sensor Num: 10, Type: TEMPERATURE, Category: THRESHOLD, Tag: Temp 7: Memory
    |     |__ Sensor Num: 11, Type: TEMPERATURE, Category: THRESHOLD, Tag: Temp 8: Power Supply
    |     |__ Sensor Num: 12, Type: TEMPERATURE, Category: THRESHOLD, Tag: Temp 9: Power Supply
    |     |__ Sensor Num: 13, Type: TEMPERATURE, Category: THRESHOLD, Tag: Temp 10: I/O Board
    |     |__ Sensor Num: 14, Type: TEMPERATURE, Category: THRESHOLD, Tag: Temp 11: I/O Board
    |     |__ Sensor Num: 15, Type: TEMPERATURE, Category: THRESHOLD, Tag: Temp 12: I/O Board
    |     |__ Sensor Num: 16, Type: TEMPERATURE, Category: THRESHOLD, Tag: Temp 13: I/O Board
    |     |__ Sensor Num: 17, Type: TEMPERATURE, Category: THRESHOLD, Tag: Temp 14: I/O Board
    |     |__ Sensor Num: 18, Type: TEMPERATURE, Category: THRESHOLD, Tag: Temp 15: I/O Board
    |     |__ Sensor Num: 19, Type: TEMPERATURE, Category: THRESHOLD, Tag: Temp 16: I/O Board
    |     |__ Sensor Num: 20, Type: TEMPERATURE, Category: THRESHOLD, Tag: Temp 17: I/O Board
    |     |__ Sensor Num: 21, Type: TEMPERATURE, Category: THRESHOLD, Tag: Temp 18: I/O Board
    |     |__ Sensor Num: 22, Type: TEMPERATURE, Category: THRESHOLD, Tag: Temp 19: CPU
    |     |__ Sensor Num: 23, Type: TEMPERATURE, Category: THRESHOLD, Tag: Temp 20: CPU
    |     |__ Sensor Num: 24, Type: TEMPERATURE, Category: THRESHOLD, Tag: Temp 21: CPU
    |     |__ Sensor Num: 25, Type: TEMPERATURE, Category: THRESHOLD, Tag: Temp 22: CPU
    |     |__ Sensor Num: 26, Type: TEMPERATURE, Category: THRESHOLD, Tag: Temp 23: I/O Board
    |     |__ Sensor Num: 27, Type: TEMPERATURE, Category: THRESHOLD, Tag: Temp 24: Memory
    |     |__ Sensor Num: 28, Type: TEMPERATURE, Category: THRESHOLD, Tag: Temp 25: Memory
    |     |__ Sensor Num: 29, Type: TEMPERATURE, Category: THRESHOLD, Tag: Temp 26: Memory
    |     |__ Sensor Num: 30, Type: TEMPERATURE, Category: THRESHOLD, Tag: Temp 27: I/O Board
    |     |__ Sensor Num: 31, Type: TEMPERATURE, Category: THRESHOLD, Tag: Temp 28: I/O Board
    |     |__ Sensor Num: 32, Type: TEMPERATURE, Category: THRESHOLD, Tag: Temp 29: System
    |     |__ Sensor Num: 33, Type: TEMPERATURE, Category: THRESHOLD, Tag: Temp 30: I/O Board
    |     |__ Inventory Num: 0, Num Areas: 1, Tag: ProLiant DL380 G6 SN:USE019NB77       ( 1) Inventory
    |
    +--- {RACK_MOUNTED_SERVER,1}{PROCESSOR,1}
    |     |__ Inventory Num: 0, Num Areas: 1, Tag: Proc 1 Inventory
    |
    +--- {RACK_MOUNTED_SERVER,1}{PROCESSOR,2}
    |     |__ Inventory Num: 0, Num Areas: 1, Tag: Proc 2 Inventory
    |
    +--- {RACK_MOUNTED_SERVER,1}{MEMORY_DEVICE,3}
    |     |__ Inventory Num: 0, Num Areas: 1, Tag: PROC 1 DIMM 3A Inventory
    |
    +--- {RACK_MOUNTED_SERVER,1}{MEMORY_DEVICE,6}
    |     |__ Inventory Num: 0, Num Areas: 1, Tag: PROC 1 DIMM 6B Inventory
    |
    +--- {RACK_MOUNTED_SERVER,1}{MEMORY_DEVICE,9}
    |     |__ Inventory Num: 0, Num Areas: 1, Tag: PROC 1 DIMM 9C Inventory
    |
    +--- {RACK_MOUNTED_SERVER,1}{MEMORY_DEVICE,12}
    |     |__ Inventory Num: 0, Num Areas: 1, Tag: PROC 2 DIMM 3A Inventory
    |
    +--- {RACK_MOUNTED_SERVER,1}{MEMORY_DEVICE,15}
    |     |__ Inventory Num: 0, Num Areas: 1, Tag: PROC 2 DIMM 6B Inventory
    |
    +--- {RACK_MOUNTED_SERVER,1}{MEMORY_DEVICE,18}
    |     |__ Inventory Num: 0, Num Areas: 1, Tag: PROC 2 DIMM 9C Inventory
    |
    +--- {RACK_MOUNTED_SERVER,1}{POWER_SUPPLY,1}
    |
    +--- {RACK_MOUNTED_SERVER,1}{POWER_SUPPLY,2}
    |
    +--- {RACK_MOUNTED_SERVER,1}{COOLING_DEVICE,1}
    |
    +--- {RACK_MOUNTED_SERVER,1}{COOLING_DEVICE,2}
    |
    +--- {RACK_MOUNTED_SERVER,1}{COOLING_DEVICE,3}
    |
    +--- {RACK_MOUNTED_SERVER,1}{COOLING_DEVICE,4}
    |
    +--- {RACK_MOUNTED_SERVER,1}{COOLING_DEVICE,5}
    |
    +--- {RACK_MOUNTED_SERVER,1}{COOLING_DEVICE,6}
    |
End of {RACK_MOUNTED_SERVER,1}
```

Hot Swap Operations

This section covers information for Field Replaceable Units (FRU) Non-managed Hot Swap.

FRU Non-Managed Hot Swap

There are no managed hot swappable components in HP's ProLiant Rack Mount servers. As a result, the iLO2 RIBCL plug-in implements the simple hot swap model events, *Not Present* or *Active* as defined in the *SAF-HPI Specification for Field Replaceable Units (FRUs)*. The FRUs may include units such as CPUs, memory modules, fans, and power supplies. These FRUs can only be replaced when the system is powered down.

Resource Events and Hot Swap Events

The Rack Mounted Server itself (the top level resource) is not a FRU and does not send resource or hot swap events. It has the SAHPI_CAPABILITY_RESOURCE bit set in its ResourceCapabilities, but does not set the SAHPI_CAPABILITY_FRU flag.

All other resources within the Rack Mount Server are FRUs, but do not support managed hot swap. These resources have the SAHPI_CAPABILITY_FRU flag set in their ResourceCapabilities, but do not have the SAHPI_CAPABILITY_MANAGED_HOTSWAP flag set. These resources support a subset of the HPI hot swap states (Active or Not Present). Some of these resources can also send resource events.

All FRU resources within the server can send hot swap events with a HotSwapState of either SAHPI_HS_STATE_ACTIVE or SAHPI_HS_STATE_NOT_PRESENT. For instance, if while the openhpid is running, a managed Rack Mount Server was shut down and one of the memory DIMMS was removed, when power to the iLo is restored, the plug-in would issue a hot swap event with a HotSwapState of SAHPI_STATE_NOT_PRESENT for that memory DIMM resource. If a new memory DIMM was added, the plug-in would send a hot swap event for the new MEMORY_DEVICE resource with a HotSwapState of SAHPI_HS_STATE_ACTIVE.

Many of the FRU resources can also send a resource event as well, with a ResourceEventType of either SAHPI_RESE_RESOURCE_FAILURE or SAHPI_RESE_RESOURCE_RESTORED. The only resources that currently do not support these resource event types are the processor and the memory DIMMS, as the RIBCL interface currently does not pass a failure status for these two resource types.

Table 12 Resource Events and Hot Swap States

Entity Type	Resource Event Types	Hot Swap States
RACK_MOUNTED_SERVER	—	—
PROCESSOR	—	SAHPI_HS_STATE_ACTIVE
		SAHPI_HS_STATE_NOT_PRESENT
MEMORY_DEVICE	—	SAHPI_HS_STATE_ACTIVE
		SAHPI_HS_STATE_NOT_PRESENT
POWER_SUPPLY	SAHPI_RESE_RESOURCE_FAILURE	SAHPI_HS_STATE_ACTIVE
	SAHPI_RESE_RESOURCE_RESTORED	SAHPI_HS_STATE_NOT_PRESENT
POWER_MODULE	SAHPI_RESE_RESOURCE_FAILURE	SAHPI_HS_STATE_ACTIVE
	SAHPI_RESE_RESOURCE_RESTORED	SAHPI_HS_STATE_NOT_PRESENT
COOLING_DEVICE	SAHPI_RESE_RESOURCE_FAILURE,	SAHPI_HS_STATE_ACTIVE
	SAHPI_RESE_RESOURCE_RESTORED	SAHPI_HS_STATE_NOT_PRESENT

Resource Event Log

None of the resources in a ProLiant Rack Mount Server maintain resource event logs. Therefore, all resources will not have the SAHPI_CAPABILITY_EVENT_LOG flag set in their ResourceCapabilities within their RPT entry. All of the saHpiEventLog*() APIs will return SA_ERR_HPI_CAPABILITY when used with these resources.

OpenHPI Domain Event Log

The OpenHPI framework uses a Domain Event Log (DEL) to store events occurring in a domain, for all plug-ins and instances of plug-ins. The size of the DEL is configurable with a default maximum size of 10,000 events.

You can change the default maximum size using the OPENHPI_DEL_SIZE_LIMIT option in the openhpi.conf file before starting openhpid. Please note that this is a dynamically, configurable parameter. If you set OPENHPI_DEL_SIZE_LIMIT = 0, the DEL is unlimited in size.

In addition, options to save the DEL to disk using the (OPENHPI_DEL_SAVE) and set the minimum severity level of events that are saved in the DEL (OPENHPI_LOG_SEV) are provided in the openhpi.conf configuration file.

OpenHPI Domain Alarm Table

The Domain Alarm Table (DAT) stores events generated with severities greater than or equal to SAHPI_MINOR. The size of the DAT is unlimited by default, but may be restricted by using the OPENHPI_DAT_SIZE_LIMIT configuration option located in the openhpi.conf file.

OpenHPI iLO2 RIBCL Plug-In Supported APIs

Table 13 provides a list of HPI FunctionAPIs and their associated support status and supporting modules.

Table 13 Supported APIs for iLO2 RIBCL Plug-In

HPI FunctionAPI	Support Status	Supporting Module
General		
saHpiVersionGet	Yes	OpenHPI framework
Session Management		
saHpiSessionOpen	Yes	OpenHPI framework
saHpiSessionClose	Yes	OpenHPI framework
Domain Discovery		
saHpiDiscover	Yes	iLO2 RIBCL plug-in
saHpiDomainInfoGet	Yes	OpenHPI framework
saHpiDrtEntryGet	Yes	OpenHPI framework
saHpiDomainTagSet	Yes	OpenHPI framework
Resource Presence Table		
saHpiRptEntryGet	Yes	OpenHPI framework
saHpiRptEntryGetByResourceId	Yes	OpenHPI framework
saHpiResourceSeveritySet	Yes	iLO2 RIBCL plug-in
saHpiResourceTagSet	Yes	iLO2 RIBCL plug-in
saHpiResourceIdGet	Yes	OpenHPI framework
saHpiGetIdByEntityPath	Yes	OpenHPI framework
saHpiGetChildEntityPath	Yes	OpenHPI framework
Event Log Management		
saHpiEventLogInfoGet	Yes	OpenHPI framework
saHpiEventLogEntryGet	Yes	OpenHPI framework
saHpiEventLogEntryAdd	Yes	OpenHPI framework
saHpiEventLogClear	Yes	OpenHPI framework
saHpiEventLogTimeGet	Yes	OpenHPI framework

Table 13 Supported APIs for iLO2 RIBCL Plug-In *(continued)*

HPI FunctionAPI	Support Status	Supporting Module
saHpiEventLogTimeSet	Yes	OpenHPI framework
saHpiEventLogStateGet	Yes	OpenHPI framework
saHpiEventLogStateSet	Yes	OpenHPI framework
saHpiEventLogOverflowReset	Yes	OpenHPI framework
saHpiEventLogCapabilitiesGet	Yes	OpenHPI framework
Event		
saHpiSubscribe	Yes	OpenHPI framework
saHpiUnsubscribe	Yes	OpenHPI framework
saHpiEventGet	Yes	—
saHpiEventAdd	Yes	OpenHPI framework
Domain Alarm Table		
saHpiAlarmGetNext	Yes	OpenHPI framework
saHpiAlarmGet	Yes	OpenHPI framework
saHpiAlarmAcknowledge	Yes	OpenHPI framework
saHp iAlarmAdd	Yes	OpenHPI framework
saHp iAlarmDelete	Yes	OpenHPI framework
Resource Data Record Management		
saHpiRdrGet	Yes	OpenHPI framework
saHpiRdrGetByInstrumentId	Yes	OpenHPI framework
Sensor		
saHpiSensorReadingGet	Yes	iLO2 RIBCL plug-in
saHpiSensorThresholdsGet		—
saHpiSensorThresholdsSet		—
saHpiSensorTypeGetd	Yes	OpenHPI framework
saHpiSensorEnableGet	Yes	iLO2 RIBCL plug-in
saHpiSensorEnableSet	Yes	iLO2 RIBCL plug-in
saHpiSensorEventEnableGet	Yes	iLO2 RIBCL plug-in
saHpiSensorEventEnableSet	Yes	iLO2 RIBCL plug-in
saHpiSensorEventMasksGet	Yes	iLO2 RIBCL plug-in
saHpiSensorEventMasksSet	Yes	iLO2 RIBCL plug-in
Control		
saHpiControlTypeGet	Yes	OpenHPI framework
saHpiControlGet	Yes	iLO2 RIBCL plug-in
saHpiControlSet	Yes	—
Inventory Data Repository		
saHpiIdrInfoGet-	Yes	iLO2 RIBCL plug-in

Table 13 Supported APIs for iLO2 RIBCL Plug-In *(continued)*

HPI FunctionAPI	Support Status	Supporting Module
saHpiIdrAreaHeaderGet	Yes	iLO2 RIBCL plug-in
saHpiIdrAreaAdd	No. The plug-in supports only read-only IDRs.	—
saHpiIdrAreaDelete	No. The plug-in supports only read-only IDRs.	—
saHpiIdrAreaAddById	No. The plug-in supports only read-only IDRs.	—
saHpiIdrFieldGet	Yes	iLO2 RIBCL plug-in
saHpiIdrFieldAdd	No. The plug-in supports only read-only IDRs.	—
saHpiIdrFieldAddById	No. The plug-in supports only read-only IDRs.	—
saHpiIdrFieldSet	No. The plug-in supports only read-only IDRs.	—
saHpiIdrFieldDelete	No. The plug-in supports only read-only IDRs.	—
saHpiResourceLoadIdGet	No. The plug-in supports only read-only IDRs.	—
saHpiResourceLoadIdSet	No. The plug-in supports only read-only IDRs.	—
Watchdog Timer		
saHpiWatchdogTimerGet	No. RIBCL does not export Annunciator watchdog controls.	—
saHpiWatchdogTimerSet	No. RIBCL does not export Annunciator watchdog controls.	—
saHpiWatchdogTimerReset	No. RIBCL does not export Annunciator watchdog controls.	—
saHpiAnnunciatorGetNext	No. RIBCL does not export Annunciator controls.	—
saHpiAnnunciatorGet	No. RIBCL does not export Annunciator controls.	—
saHpiAnnunciatorAcknowledge	No. RIBCL does not export Annunciator controls.	—
saHpiAnnunciatorAdd	No. RIBCL does not export Annunciator controls.	—
saHpiAnnunciatorDelete	No. RIBCL does not export Annunciator controls.	—
saHpiAnnunciatorModeGet	No. RIBCL does not export Annunciator controls.	—
saHpiAnnunciatorModeSet	No. RIBCL does not export Annunciator controls.	—
Hotswap Management		
saHpiHotSwapPolicyCancel	No. ProLiant Rack Mounts do not have any hot swap components.	—
saHpiResourceActiveSet	No. ProLiant Rack Mounts do not have any hot swap components.	—
saHpiResourceInactiveSet	No. ProLiant Rack Mounts do not have any hot swap components.	—
saHpiResourceFailedRemove	No. ProLiant Rack Mounts do not have any hot swap components.	—
saHpiAutoInsertTimeoutGet	No. ProLiant Rack Mounts do not have any hot swap components.	—
saHpiAutoInsertTimeoutSet	No. ProLiant Rack Mounts do not have any hot swap components.	—
saHpiAutoExtractTimeoutGet	No. ProLiant Rack Mounts do not have any hot swap components.	—

Table 13 Supported APIs for iLO2 RIBCL Plug-In *(continued)*

HPI FunctionAPI	Support Status	Supporting Module
saHpiAutoExtractTimeoutSet	No. ProLiant Rack Mounts do not have any hot swap components.	—
saHpiHotSwapStateGet	No. ProLiant Rack Mounts do not have any hot swap components.	—
saHpiHotSwapActionRequest	No. ProLiant Rack Mounts do not have any hot swap components.	—
saHpiHotSwapIndicatorStateGet	No. ProLiant Rack Mounts do not have any hot swap components.	—
saHpiHotSwapIndicatorStateSet	No. ProLiant Rack Mounts do not have any hot swap components.	—
Configuration		
saHpiParmControl	No. ProLiant Rack Mounts do not have any configurable components.	—
Reset		
saHpiResourceResetStateGet	Yes. HP ProLiant Rack Mount Servers do not support pulsed reset and the only valid value this API returns is:`SAHPI_RESET_DEASERT`	iLO2 RIBCL plug-in
saHpiResourceResetStateSet	Yes. Supports cold and warm reset.	iLO2 RIBCL plug-in
Power		
saHpiResourcePowerStateGet	Yes	iLO2 RIBCL plug-in
saHpiResourcePowerStateSet	Yes	iLO2 RIBCL plug-in