hp WBEM solutions

hp technical data sheet

CPU Instance Provider

Provider Overview

The CPU Instance Provider is a Web Based Enterprise Management (WBEM) Instance Provider providing information about processor inventory on supported HP 9000 systems running HPUX.

Description

The CPU Instance Provider is a Web Based Enterprise Management (WBEM) Instance Provider. It provides information about processor inventory on supported HP 9000 systems running HPUX. This provider is compliant with the Common Information Model (CIM) 2.7 Schema, proposed by the Distributed Management Task Force (DMTF).

The Common Information Model (CIM) is an extensible, object-oriented data model that contains information about different parts of an enterprise. The CPU Instance Provider allows clients to query for information about the processor subsystem on the managed server using a management application that is compliant with the CIM 2.7 schema.

The CPU Instance Provider implements the processor-related CIM classes, proposed in the DMTF CIM 2.7 revision. In addition to the properties that belong to the standard CIM classes, the CPU Instance Provider serves information that is specific to HP servers, by implementing HP-specific CIM classes, derived from the standard DMTF classes.

The following Management Object Format (MOF) classes are handled by the CPU Instance Provider:

- HP_Processor and HPUX_Processor

  HP_Processor (subclass of CIM_Processor) and HPUX_Processor (subclass of HP_Processor) represent “logical” information about the processors, including status, the family of the processors, clock-speeds, etc.

- HP_ProcessorChip

  HP_ProcessorChip (subclass of CIM_Chip) represents “physical” information about the processor, such as the processor firmware revision, architecture revision, etc.

- HP_ProcessorLocation and HPUX_ProcessorLocation

  HP_ProcessorLocation (subclass of CIM_Location) and HPUX_ProcessorLocation (subclass of HP_ProcessorLocation) retrieves information about the physical location of the processor module and the core of the processor, as seen from the physical perspective. This includes identification of which cell the processor chip resides in, the slot numbers, cabinet numbers, etc.

- HP_ProcessorCollection

  HP_ProcessorCollection (subclass of HP_GroupSystemSpecificCollection) represents the processor-subsystem on the computer-system. The group-operational-status property of this class represents the health of the current system’s processor-subsystem.

In addition, the CPU Instance Provider also implements association classes to associate the instances of the different CIM classes mentioned above. These include:

- HP_RealizesProcessor (subclass of CIM_Realizes): This class identifies which logical Processor (HP_Processor) instance is associated to which physical processor (HP_ProcessorChip) instance.

- HP_ProcessorChipInLocation (subclass of CIM_PhysicalElementLocation): This class indicates the physical location (HP_ProcessorLocation) corresponding to a specific processor chip (HP_ProcessorChip).

- HP_ProcessorGroupHostedCollection (subclass of HP_GroupHostedCollection): This class represents the association between the processor subsystem (HP_ProcessorCollection) and the computer-system to which the subsystem belongs.

- HP_MemberOfProcessorCollection (subclass of CIM_MemberOfCollection): This class retrieves the “member-of” relationship between the processor-class-instances and the processor-collection (HP_ProcessorCollection). Instances of this class associate all CIM-

instances that contribute to group-operational-status of the HP_ProcessorCollection, with the instance of the HP_ProcessorCollection itself.

The MOF classes mentioned above (i.e. all MOF classes prefixed with "HP_") are HP-specific extensions to the CIM Schema, and are registered in the "root/cimv2" namespace.

The following example illustrates the relationship between the MOF classes mentioned above. On an PA-RISC-based server containing a single cell, with 4 processor-slots and one dual-core PA-RISC 8800 processor, the CIM instances returned by the CPU Instance Provider are as follows:

- 2 instances of HPUX_Processor (one for each of the processors visible to the running HP-UX kernel)
- 1 instance of HP_ProcessorChip (representing the single processor chip Field Replaceable Unit(FRU), i.e. the PA-RISC processor chip)
- 4 instances of HPUX_ProcessorLocation (one representing the slot occupied by the PA-RISC processor chip above, 3 empty slots)
- 2 instances of HPUX_RealizesProcessor (each one associating one of the 2 HPUX_Processor instances with the single HP_ProcessorChip instance)
- 1 instance of HPUX_ProcessorChipInLocation (associating the single HPUX_ProcessorChip instance with the HPUX_ProcessorLocation instance (location/slot in which it rests)).
- 1 instance of HP_ProcessorCollection (representing the single instance of the processor-subsystem).
- 2 instances of HP_MemberOfProcessorCollection (one for each instance of the CIM-class that contributes to the group-operational-status of the processor-subsystem).
- 1 instance of HP_ProcessorGroupHostedCollection (associating the single processor-subsystem to the computer-system-instance).

If the server also had a deconfigured PA-RISC 8800 processor, the situation would be very similar, with the following changes:

- There would be 2 instances of HP_ProcessorChip instead of 1 (to include the single deconfigured processor-module.)
- There would be 2 instances of HPUX_ProcessorChipInLocation (including the association between the deconfigured processor-module and the slot it occupies.)

All the rest remains as before. For instance, since the logical processors on the deconfigured processor-modules will not be visible to the OS kernel. Hence, the number of HPUX_Processor instances remains unchanged.

The situation with HP Integrity servers is analogous to the above.

For all the MOF classes mentioned above, the CPU Instance Provider supports the following standard CIM Operations:

- enumerateInstanceNames()
- enumerateInstances()
- getInstance()

The following CIM operations are not supported by the CPU Instance Provider:

- createInstance()
- deleteInstance()
- modifyInstance()

The invocation of any of these methods will result in a CIM_ERR_NOT_SUPPORTED exception.

Requirements

WBEMServices A.02.07 WBEM Services CORE Product

OnlineDiag B.11.31.03.YY

Supported managed resources

This provider provides logical information about system CPU's, "physical" attributes of the processor chip, and details of the physical-location of the processor.

The CPU Instance Provider provides only the information about the above resources. It does not provide any management, diagnostic or configuration capabilities for the above resources.

For a list of supported platforms, see the SFM Release Notes at, http://docs.hp.com/en/diag.

Setting up this Provider

Installing this Provider

The installation of the bundle SysFaultMgmt will set up this Provider.

Ensure that the appropriate version of HPWBEM services and OnlineDiag are installed as mentioned in the requirements section.

Use swinstall to install the product: “swinstall –s Fully_Qualified_Depot_Name SysFaultMgmt”

(Detailed instructions for installation of the System Fault Management product are available in the System Fault Management Administrator's Guide, available at http://docs.hp.com/en/diag.)

On installation, the shared-library files, executable binaries, configuration files and MOF definition and registration files will be available in the /opt/sfm/ directory, as follows:

- The provider library is libsfmproviders.1. This is available in /opt/sfm/lib/, along with all the other libraries it uses to implement the CPU Instance provider. A symbolic link is made in /opt/wbem/providers/lib/libsfmprovider.sl to link to the libsfmprovider.1 library in /opt/sfm/lib/.

The CIM MOF file containing the definitions of the HP-specific MOF classes, (namely HP_Processor.mof) will be available in /opt/sfm/schemas/mof. This directory will also include the provider registration file, namely SFMProvidersHPOnlyIaR.mof , SFMProvidersHPOnlyR.mof and SFMProvidersCommonR.mof. Note: All the HP-specific MOF classes will be registered under the “root/cimv2” namespace.

- The /opt/sfm/bin/ directory will contain the binary executable files that are used by the CPU Instance Provider. This includes the “sfmconfig” utility, that is used for sending notifications to the CPU Instance Provider (e.g. on updation to the configuration file).
- The /var/opt/sfm/conf/ directory will contain the (XML) configuration files of the SFM product.
- The /opt/sfm/msgcat/C directory will contain the catalog files for all the supported locales. (This is used for the localization of the message strings in CPU Instance Provider).
- The /var/opt/sfm/log/ directory will contain log files generated during the execution of the CPU Instance Provider.

All systems supported by SFM are supported by the CPU Instance Provider.

Configuring this Provider

CPU Provider uses a common configuration file along with Memory Instance Provider and EMSWrapper Indication Provider. So editing the configuration file will affect the other two providers as well. The configuration file can be found in – /var/opt/sfm/conf/FMLoggerConfig.xml

The file specifies the logging threshold severity, and the location of the log-file. The contents of the file are as follows:

```
<SFMConfig>
   <LoggerConfig>
                        <Severity> WARNING </Severity>
                        <Target> /var/opt/sfm/log/sfm.log </Target>
                          <FileSize> 20480 </FileSize>
                          <NBackupFiles> 3 </NBackupFiles>
  </LoggerConfig>
</SFMConfig>
```

In order to change the logging configuration, the following steps are to be followed:

1. Edit the configuration file /var/opt/sfm/conf/FMLoggerConfig.xml to change the threshold logging level and/or target.

a) Threshold: Possible values are (in increasing severity)

INFORMATIONAL

WARNING

ERROR

CRITICAL

STOPLOGGING

MILESTONE

NOTE: The INFORMATIONAL logging severity will generate a lot of log messages. It is strongly advised not to use this severity level for a long time, for the generated log-files may use a lot of disk space. The default (and recommended) threshold in the runtime environment is WARNING.

b) Target: Possible values include:

(i) STDOUT: All log messages are delivered to console.

(ii) The complete path to the file where the log messages are to be written

NOTE: The current implementation of the logging mechanism does not create the directories in the path of the log file. It assumes that the path to the log file (target specified in the configuration file) already exists. i.e., if the target is specified as "/abc/def/ghi.log", the directory "/abc/def/" should already exist. The super-user (root) must have write-access to this directory.

2. Run /opt/sfm/bin/sfmconfig command to specify the changed configuration file. i.e.

$ /opt/sfm/bin/sfmconfig –c /var/opt/sfm/conf/FMLoggerConfig.xml

Note that the complete path of the configuration file must be provided to the sfmconfig program.

Using this Provider

Schema supported by this Provider

The "Description" section explains in brief the different MOF classes supported by the CPU Instance Provider. The following tables list all the supported properties corresponding to these MOF classes, along with the properties inherited from the standard CIM MOF classes, as per CIM 2.7 schema specifications.

Note: All key properties corresponding to the CIM classes are supported by the CPU Instance Provider. The few non-key properties not supported (currently) by the CPU Instance Provider are not listed below.

Note:

1. All key properties, corresponding to the CIM classes, are supported by the CPU Instance Provider.
2. The non-key properties that are not supported by the CPU Instance Provider are not listed below.

Table 1: HP_Processor and HPUX_Processor Properties (Logical Processor Information):
Table 1 describes the properties of the HP_Processor and HPUX_Processor CIM classes. It has three columns. The first is the property name (including type and units), the second is the property inheritance (indicating which class or superclass defines the property), and the third is the property's value and data source. Each row describes a property.

Property name	Property inheritance	Property value (and data source)
string Caption	Inherited from CIM_ManagedElement	This value is always returned as "Processor (SPU)".
string Description	Inherited from CIM_ManagedElement	For PA-RISC-based servers, this string is set to "This is a PA RISC Processor, with the following details: ", followed by location-details for the processor. Location details include (where available): 1. Cabinet Number 2. Card Cage Number 3. Backplane Number 4. Cell Slot number 5. Slot number 6. SPU Number (as seen by the OS instance) 7. Hard Physical Address
string ElementName	Inherited from CIM_ManagedElement	This string is set to "PA RISC Processor".

String Name	Inherited from CIM_ManagedSystemElement	This string is set to “PA RISC Processor”.
uint16 OperationalStatus []	Inherited from CIM_ManagedSystemElement	The Value-Map associated with this property (as per the CIM 2.7 schema specification) is as follows: ValueMap {"0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14", "15", "16", "17"}, Values {"Unknown", "Other", "OK", "Degraded", "Stressed", "Predictive Failure", "Error", "Non-Recoverable Error", "Starting", "Stopping", "Stopped", "In Service", "No Contact", "Lost Communication", "Aborted", "Dormant", "Supporting Entity in Error", "Completed"} The OperationalStatus array contains multiple values, to indicate the different aspects of the CPU’s status. 1. OpStatus[0] indicates “summary health-status” of the processor. a. If there’s nothing wrong with the processor, this is set to 2 (OK). b. In case of errors (see #4 below), it reflects the value of OpStatus[3]. c. In case there is no error, but either the Activity or DeconfigurationStatus of the processor could not be determined, this value is set to 0 (Unknown). d. If the processor was deactivated because of errors, the value is set to 6. e. If, instead, the processor is found to be deactivated, the value is set to 15 (Dormant). 2. OpStatus[1] indicates the “activity-status” of the processor. A processor is “active” if processes may be scheduled on the processor. Otherwise, the processor is “inactive”. a. If the processor is active, the value is set to OK. b. If the processor was deactivated because of errors, the value is set to 6 (Error). c. If the processor is deactivated or idle, the value is set to 15 (Dormant). d. If the activity-status of the processor can’t be determined, the value is set to 0 (Unknown). 3. OpStatus[2] indicates the “deconfiguration-status” of the processor chip with which the processor is associated. a. If the processor chip is configured, the value is 2(OK). b. If the processor chip is marked for deconfiguration, the value is 9 (Stopping). c. If the processor chip is deconfigured, the value is 10 (Stopped). d. If the processor chip is marked for reconfiguration, the value is 8 (Starting). e. If the deconfiguration-status could not be determined, the value is set to 0 (Unknown). 4. OpStatus[3] indicates the “error-status” of the processor. a. In case there are no errors associated with the processor, the value is set to 2 (OK). b. If there are serious errors on the processor, the value is set to 6 (Error). c. If the processor might fail in the future, the value is set to 5 (Predictive Failure).

string StatusDescriptions[]	Inherited from CIM_ManagedSystemElement	This contains string descriptions for the status values returned in the Operational Status array described above. Each value in the StatusDescriptions array corresponds to the (localized) verbose status description for the value at the same index in the OperationalStatus array. The Strings describing the different values in the OperationalStatus Array are as follows: 1. StatDesc[0]: This is the summary description of the status of the processor. a. If there's nothing wrong with the processor, the value will be "Processor is OK". b. If the processor has errors, the value will reflect StatDesc[3], explained below. c. If the processor has no errors, but the activity/deconfiguration status of the processor could not be determined, the value will be set to "Processor is in an UKNOWN state." d. If the processor is deactivated because of errors, the value will be set to "The processor has been deactivated because of errors." e. If the Activity status of the processor is found to be idle/deactivated, the value will reflect StatDesc[1], described below. 2. StatDesc[1] indicates the "activity-status" of the Processor. a. If the processor is Active, the value is "Processor is currently Active." b. If the processor was deactivated because of errors, the value is set to "Processor has been deactivated because of errors." c. If the processor is idle/dormant, the value is "Processor is Idle. There are no processes scheduled on this processor. d. If the "activity-status" can't be determined, the value is "Activity-status of the processor could not be determined." 3. StatDesc[2] indicates the "deconfiguration-status" of the processor. a. If the processor is configured, the value is "Processor is currently Configured." b. If the processor is marked for deconfiguration, the value is "The processor-module is currently configured, and will be deconfigured in the next reboot." c. If the processor is deconfigured, the value will be " The processor-module is currently deconfigured." d. If the processor is marked for configuration, the value will be "The processor-module is currently deconfigured, and will be configured in the next reboot." e. If the Deconfiguration status of the processor can't be determined, the value

		will be “DeconfigurationStatus of the processor could not be determined.” 4. StatDesc[3] indicates the “error-status” of the processor. a. If the processor has no errors, the value will be “Processor-Module has no errors.” b. If the processor has errors, the value will be “Processor-Module has errors.” c. If the processor will fail in the future, the value will be “Processor is currently ok, but it might fail in the future.”
string SystemCreationClassName [Key]	Inherited from CIM_LogicalDevice	Fixed string “CIM_ComputerSystem”
string SystemName [Key]	Inherited from CIM_LogicalDevice	The host name of the server.
string CreationClassName [Key]	Inherited from CIM_LogicalDevice	This is set to the name of the instantiated sub-class, i.e. “HPUX_Processor”.
string DeviceID [Key]	Inherited from CIM_LogicalDevice	For HP 9000 systems, this is set to the Hard Physical Address of the processor.
string LID	Property of HP_Processor	The Hardware Physical Address (HPA) (PA platform) of the processor,or the contents of the processor's Local ID (LID) register (Itanium(R)-based platforms)
Uint16 Family	Inherited from CIM_Processor	The Value-Map for this property looks as follows: ValueMap {"1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14", "15", "16", "17", "18", "19", "24", "25", "26", "27", "28", "29", "30", "31", "32", "33", "34", "35", "36", "37", "38", "39", "48", "49", "50", "51", "52", "53", "54", "55", "64", "65", "66", "67", "68", "69", "80", "81", "82", "83", "84", "85", "86", "87", "88", "96", "97", "98", "99", "100", "101", "112", "120", "121", "128", "130", "144", "145", "146", "147", "148", "149", "150", "160", "176", "177", "178", "179", "180", "181", "182", "183", "184", "190", "200", "201", "202", "250", "251", "260", "261", "280", "281", "300", "301", "302", "320", "350", "500"} Values {"Other", "Unknown", "8086", "80286", "80386", "80486","8087", "80287", "80387", "80487", // 11 "Pentium(R) brand", "Pentium(R) Pro", "Pentium(R) II", "Pentium(R) processor with MMX(TM) technology", "Celeron(TM)", "Pentium(R) II Xeon(TM)", "Pentium(R) III", "M1 Family", "M2 Family", //24 "K5 Family", "K6 Family", "K6-2", "K6-3", "AMD Athlon(TM) Processor Family", "AMD(R) Duron(TM) Processor", "AMD29000 Family", //31 "K6-2+", "Power PC Family", "Power PC 601", "Power PC 603", "Power PC 603+", "Power PC 604", "Power PC 620", "Power PC X704", "Power PC 750", // 48

		"Alpha Family", "Alpha 21064", "Alpha 21066", "Alpha 21164", "Alpha 21164PC", "Alpha 21164a", "Alpha 21264", "Alpha 21364", // 64 "MIPS Family", "MIPS R4000", "MIPS R4200", "MIPS R4400", "MIPS R4600", "MIPS R10000", // 80 "SPARC Family", "SuperSPARC","microSPARC II", "microSPARC IIep", "UltraSPARC", "UltraSPARC II", "UltraSPARC IIi", "UltraSPARC III", "UltraSPARC IIIi", // 96 "68040", "68xxx Family", "68000", "68010", "68020", "68030", // 112 "Hobbit Family", "Crusoe(TM) TM5000 Family", "Crusoe(TM) TM3000 Family", "Weitek", "Itanium(TM) Processor", // 144 "PA-RISC Family", "PA-RISC 8500", "PA-RISC 8000", "PA-RISC 7300LC", "PA-RISC 7200", "PA-RISC 7100LC", "PA-RISC 7100", // 160 "V30 Family", "Pentium(R) III Xeon(TM)", "Pentium(R) III Processor with Intel(R) SpeedStep(TM) ""Technology", "Pentium(R) 4", "Intel(R) Xeon(TM)", // 180 "AS400 Family", "Intel(R) Xeon(TM) processor MP", "AMD AthlonXP(TM) Family", "AMD AthlonMP(TM) Family", "Intel(R) Itanium(R) 2", // 190 "K7", // 200 "IBM390 Family", "G4", "G5", // 250 "i860", "i960", "SH-3", "SH-4", "ARM", "StrongARM", // 300 "6x86", "MediaGX", "MII", "WinChip", "DSP", "Video Processor"}, For PA-RISC servers, this property is set to “144”, to indicate “PA RISC Processor Family”.
Uint32 CurrentClockSpeed	Inherited from CIM_Processor	The clock speed of the processor, in MHz.
Uint16 DataWidth	Inherited from CIM_Processor	Width of the data-bus of the processor.
Uint16 LoadPercentage	Inherited from CIM_Processor	Loading of the processor, averaged over one minute, in percentage.
Uint16 SpuId	Inherited from HPUX_Processor	This is an ID of the processor as seen by the OS.
Unit16 EnabledState	Inherited from CIM_LogicalDevice	EnabledState is an integer enumeration that indicates the enabled and disabled states of an element. It can also indicate the transitions between these requested states.

		For example, shutting down (value=4) and starting (value=10) are transient states between enabled and disabled. The following text briefly summarizes the various enabled and disabled states: Enabled (2) indicates that the element is or could be executing commands, will process any queued commands, and queues new requests. Disabled (3) indicates that the element will not execute commands and will drop any new requests. Shutting Down (4) indicates that the element is in the process of going to a Disabled state. Not Applicable (5) indicates the element does not support being enabled or disabled. Enabled but Offline (6) indicates that the element might be completing commands, and will drop any new requests. Test (7) indicates that the element is in a test state. Deferred (8) indicates that the element might be completing commands, but will queue any new requests. Quiesce (9) indicates that the element is enabled but in a restricted mode. Starting (10) indicates that the element is in the process of going to an Enabled state. New requests are queued.
Unit16 RequestedState	Inherited from CIM_LogicalDevice	RequestedState is an integer enumeration that indicates the last requested or desired state for the element, irrespective of the mechanism through which it was requested. The actual state of the element is represented by EnabledState. This property is provided to compare the last requested and current enabled or disabled states. Note that when EnabledState is set to 5 ("Not Applicable"), then this property has no meaning. Refer to the EnabledState property description for explanations of the values in the RequestedState enumeration. "Unknown" (0) indicates the last requested state for the element is unknown. Note that the value "No Change" (5) has been deprecated in lieu of indicating the last requested state is "Unknown" (0). If the last requested or desired state is unknown, RequestedState should have the value "Unknown" (0), but may have the value "No Change" (5).Offline (6) indicates that the element has been requested to transition to the Enabled but Offline EnabledState. It should be noted that there are two new values in RequestedState that build on the statuses of EnabledState. These are "Reboot" (10) and "Reset" (11). Reboot refers to doing a "Shut Down" and then moving to an "Enabled" state. Reset indicates that the element is first "Disabled" and then "Enabled". The distinction between requesting "Shut Down" and "Disabled" should also be noted. Shut Down requests an orderly transition to the Disabled state, and might involve removing power, to completely erase any existing state. The Disabled state requests an immediate disabling of the element, such that it will not execute or accept any commands or processing requests. This property is set as the result of a method invocation (such as Start or StopService on CIM_Service), or can be overridden and defined as WRITEable in a subclass. The method approach is considered superior to a WRITEable property, because it allows an explicit invocation of the operation and the return of a result code. If knowledge of the last RequestedState is not supported for the EnabledLogicalElement, the property shall be NULL or have the value 12 "Not Applicable".
Unit16 EnabledDefault	Inherited from CIM_LogicalDevice	An enumerated value indicating an administrator's default or startup configuration for the Enabled State of an element. By default, the element is "Enabled" (value=2).

String OtherIdentifyingInfo	Inherited from CIM_LogicalDevice	OtherIdentifyingInfo captures data, in addition to DeviceID information, that could be used to identify a LogicalDevice. For example, you could use this property to hold the operating system's user-friendly name for the Device.
String IdentifyingDescriptions	Inherited from CIM_LogicalDevice	An array of free-form strings providing explanations and details behind the entries in the OtherIdentifyingInfo array. Note that each entry of this array is related to the entry in OtherIdentifyingInfo that is located at the same index.
String LID	Inherited from HP_processor	The Hardware Physical Address (HPA) (PA platform) of the processor or the contents of the processor's Local ID (LID) register (Itanium(R)-based platforms).
Unit16 PSetId	Inherited from HP_processor	ID of the PSet to which the processor belongs.
String HardwarePath	Inherited from HP_processor	HardwarePath of the CPU returned by the HP-UX OS

Table 2: HP_ProcessorChip properties

Table 2 describes the properties of the HP_ProcessorChip class. It has three columns. The first is the property name (including type and units), the second is the property inheritance (indicating which class or superclass defines the property), and the third is the property's value and data source. Each row describes a property.

Property name	Property inheritance	Property value (and data source)
string Caption	Inherited from CIM_ManagedElement.	This value is always returned as "Processor-Module".
string Description	Inherited from CIM_ManagedElement.	For PA-RISC servers, this string is set to "This is a PA RISC Processor, with the following details: ", followed by location-details for the processor. Location details include (where available): 1. Cabinet Number 2. Card Cage Number 3. Backplane Number 4. Cell Slot number 5. Slot number Tails:
string ElementName	Inherited from CIM_ManagedElement	This string is set to "PA RISC Processor-Module".
String Name	Inherited from CIM_ManagedSystemElement	For PA-RISC servers, this string is set to "PA RISC Processor-Module". For HP Integrity servers, this string is set to "Intel ® Itanium ® 2 Processor-Module".
Uint16 OperationalStatus	Inherited from CIM_ManagedSystemElement.	The Value-Map associated with this property (as per the CIM 2.9 Schema Specification) is as follows: ValueMap {"0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14", "15", "16", "17"}, Values {"Unknown", "Other", "OK", "Degraded", "Stressed", "Predictive Failure", "Error", "Non-Recoverable Error", "Starting", "Stopping", "Stopped",

		"In Service", "No Contact", "Lost Communication", "Aborted", "Dormant", "Supporting Entity in Error", "Completed"} The OperationalStatus array contains multiple values, to indicate the different aspects of the CPU's status. 1. OpStatus[0] indicates summary health-status of the processor. a. If there's nothing wrong with the processor, this is set to 2 (OK). b. In case of errors (see description for OpStatus[2] below), it reflects the value of OpStatus[2]. c. If one of the (logical) processors hosted on the processor-module has been deactivated because of errors, the value is set to 6 (Error). d. In case there is no error, but DeconfigurationStatus of the processor could not be determined, this value is set to 0 (Unknown). 2. OpStatus[1] indicates the "deconfiguration-status" of the processor. a. If the processor chip is configured, the value is 2(OK). b. If the processor chip is marked for deconfiguration, the value is 9 (Stopping). c. If the processor chip is deconfigured, the value is 10 (Stopped). d. If the processor chip is marked for reconfiguration, the value is 8 (Starting). e. If the deconfiguration-status could not be determined, the value is set to 0 (Unknown). 3. OpStatus[2] indicates the "error-status" of the processor. a. In case there are no errors associated with the processor, the value is set to 2 (OK). b. If there are serious errors on the processor, the value is set to 6 (Error). c. If the processor might fail in the future, the value is set to 5 (Predictive Failure).
string StatusDescriptions[]	Inherited from CIM_ManagedSystemElement	This contains string descriptions for the status values returned in the Operational Status array described above. Each value in the StatusDescriptions array corresponds to the (localized) verbose status description for the value at the same index in the OperationalStatus array. The Strings describing the different values in the OperationalStatus Array are as follows: 1. StatDesc[0]: This is the summary description of the status of the processor. a. If there's nothing wrong with the processor, the value will be "Processor is OK". b. If the processor chip has errors, the value will reflect StatDesc[2], explained below. c. If one of the (logical) processors hosted on the

		processor module has errors, the value is set to "At least one processor hosted on this processor module has been deactivated because of errors." d. If the processor has no errors, but the deconfiguration status of the processor could not be determined, the value will be set to "Processor is in an UKNOWN state." 2. StatDesc[1] indicates the deconfiguration-status of the processor. a. If the processor is configured, the value is "Processor is currently Configured." b. If the processor is marked for deconfiguration, the value is "The processor-module is currently configured, and will be deconfigured in the next reboot." c. If the processor is deconfigured, the value will be " The processor-module is currently deconfigured." d. If the processor is marked for configuration, the value will be "The processor-module is currently deconfigured, and will be configured in the next reboot." e. If the deconfiguration-status can't be determined, the value will be "DeconfigurationStatus of the processor could not be determined." 3. StatDesc[2] indicates the Error-status of the processor. a. If the processor has no errors, the value will be "Processor-Module has no errors." b. If the processor has errors, the value will be "Processor-Module has errors." c. If the processor will fail in the future, the value will be set to "Processor-Module is currently ok, but it might fail in the future."
String Tag [Key]	Inherited from CIM_PhysicalElement	This string will be set to a unique value, indicating the physical location of the processor chip.
String CreationClassName [Key]	Inherited from CIM_PhysicalElement	The name of the subclass being instantiated, i.e. "HP_ProcessorChip".
String Model	Inherited from CIM_PhysicalElement	This contains the model string, identifying the model of the processor chip. For HP PA-RISC servers, this contains a string of the format: "PA-RISC XXXX processor (H-Version YYYY)", where XXXX: identifies the PA-RISC processor-model, e.g. 8600, 8700, etc. YYYY: identifies the H-Version of the server platform (in Hexadecimal representation). E.g. For an rp5400 server, the string could be: "PA-RISC 8700 processor (H-Version 0x5df)"
String SerialNumber	Inherited from CIM_PhysicalElement	This contains the serial-number of the processor chip.

Uint16 ArchitectureRevision	Inherited from HP_ProcessorChip	The Value-Map of this property is as follows: ValueMap {"0", "1", "2", "3", "4", "5"}, Values {"Unknown", "Other", "PARISC 1.0", "PARISC 1.1", "PARISC 2.0", "Itanium Architecture"}] The Value of this property indicates the Architecture of the processor: 0 for Unknown 1 for Other 2 for PA RISC 1.0 3 for PA RISC 1.1 4 for PA RISC 2.0
String FirmwareRevision	Inherited from HP_ProcessorChip	Firmware Recipe Number, identifying the processor's Firmware Revision.
uint64 DeconfigurationState	Inherited from HP_ProcessorChip	Configuration state of the Processor-Module. Whether it is configured, deconfigured etc
uint16 NumberOfCores	Inherited from HP_ProcessorChip	The number of processor-cores available on the processor chip. For instance, for dual-core processors, such as PA-RISC 8800, the value would be 2.
uint16 NumberOfCoresInOS	Inherited from HP_ProcessorChip	The number of processor-cores from the current processor chip that are associated with the OS instance. For instance, for a dual-core processor chip with only one core assigned to the OS instance, this value will be 1.
uint16 ThreadsPerCore	Inherited from HP_ProcessorChip	The number of logical processor-threads (e.g. hyperthreads) associated with any core on the processor chip. For instance, if hyperthreading/symmetric-multithreading is enabled, and there are 2 threads for every core, the value will be 2.
uint16 PotentialThreadsPerCore	Inherited from HP_ProcessorChip	The maximum number of logical processor-threads that can potentially be associated with any core on the processor chip. For instance, on a processor that supports hyperthreading/symmetric-multithreading, with 2 threads per processor-core, this value will be 2, irrespective of whether hyperthreading/symmetric-multithreading is enabled.
String PartNumber	Inherited from CIM_PhysicalElement	The part number assigned by the organization that is responsible for producing or manufacturing the PhysicalElement
String Manufacturer	Inherited from CIM_PhysicalElement	Not supported
Datetime ManufactureDate	Inherited from CIM_PhysicalElement	Not supported
String SKU	Inherited from CIM_PhysicalElement	The stock-keeping unit number for this PhysicalElement.
RemovalConditions	Inherited from CIM_PhysicalElement	Not supported

Table 3: HP_ProcessorLocation and HPUX_ProcessorLocation properties

Table 3 describes the properties of the HP_ProcessorLocation and HPUX_ProcessorLocation classes. It has three columns. The first is the property name (including type and units), the second is the property inheritance (indicating which class or superclass defines the property), and the third is the property's value and data source. Each row describes a property.

Property name	Property inheritance	Property value (and data source)
string Caption	Inherited from CIM_ManagedElement.	This value is always returned as "Processor-Slot".
string Description	Inherited from CIM_ManagedElement.	For HP PA-RISC servers, this string is set to "This is the

		location of an PA RISC Processor-Module, with the following details: ", followed by location-details for the processor. Location details include (where available): • Cabinet Number • Card Cage Number • Backplane Number • Cell Slot number • Slot number
string ElementName	Inherited from CIM_ManagedElement	The value is always returned as "Processor-Slot".
String Name [Key]	Inherited from CIM_Location	The location is returned as a string, of the form "CabinetNumber = <Cabinet#> : CardCageNumber = <CardCage#> : BackPlaneNumber = <BackPlane#> : CellSlotNumber = <CellSlot#> : SlotNumber = <Slot#> : " Note that the SPU# is the number assigned to the processor by that instance of the OS. For instance, on a (PA RISC) Keystone platform, could read as follows: "CabinetNumber = 0 : CellSlotNumber = 1 : SlotNumber = 0".
String PhysicalPosition [Key]	Inherited from CIM_Location	A string indicating (uniquely) the position of the processor.
String CellNumber	Inherited from HP_ProcessorLocation	The number of the cell to which the processor belongs.
String SlotNumber	Inherirted from HP_ProcessorLocation	The number of the slot in which the processor rests.
PhysicalLocationLevels	HP_MemoryLocation	Not supported
PhysicalLocationValues	HP_MemoryLocation	Not supported
IsEmpty	HP_MemoryLocation	Whether the memory location is occupied or not.
CabinetNumber	Inherited from HP_ProcessorLocation	The cabinet number.

Table 4: HP_RealizesProcessor properties

Table 4 describes the properties of the HP_RealizesProcessor association class (associating HP_ProcessorChip and HP_Processor). It has three columns. The first is the property name (including type and units), the second is the property inheritance (indicating which class or superclass defines the property), and the third is the property's value and data source. Each row describes a property.

Property name	Property inheritance	Property value (and data source)
HP_ProcessorChip ref Antecedent	Property of HP_RealizesProcessor	Object path of the HP_ProcessorChip Instance.
HP_Processor ref Dependent	Property of HP_RealizesProcessor	Object path of the HP_Processor Instance.

Table 5: HP_ProcessorChipInLocation properties

Table 5 describes the properties of the HP_ProcessorChipInLocation association class (associating HP_ProcessorChip and HP_ProcessorLocation). It has three columns. The first is the property name (including type and units), the second is the property inheritance (indicating which class or superclass defines the property), and the third is the property's value and data source. Each row describes a property.

Property name	Property inheritance	Property value (and data source)
HP_ProcessorChip ref Element	Property of HP_ProcessorChipInLocation	Object path of the HP_ProcessorChip Instance.
HP_ProcessorLocation ref PhysicalLocation	Property of HP_ProcessorChipInLocation	Object path of the HP_ProcessorLocation Instance.

Table 6: HP_ProcessorCollection supported properties. (Properties that are not supported are not mentioned.)

This class represents the processor-subsystem, i.e. the collection of processors (CPUs) in the computer system.

The getInstance() method is not supported for this association class.

Property name	Property inheritance	Property value (and data source)
String InstanceID (Key)	Inherited from CIM_SystemSpecificCollection	Hewlett-Packard:diags.sfm:<CreationClassName>:<LocalID> CreationClassName reflects the collection class name. LocalID is always 0, as we are creating only 1 instance of collection class.
String Caption	Inherited from HP_GroupSystemSpecificCollection	"HP_ProcessorCollection"
Uint16[] GroupOperationalStatus	Inherited from HP_GroupSystemSpecificCollection	ValueMap {"0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14", "15", "16", "17"}, Values {"Unknown", "Other", "OK", "Degraded", "Stressed", "Predictive Failure", "Error", "Non-Recoverable Error", "Starting", "Stopping", "Stopped", "In Service", "No Contact", "Lost Communication", "Aborted", "Dormant", "Supporting Entity in Error", "Completed"},
String[] GroupStatusDescriptions	Inherited from HP_GroupSystemSpecificCollection	Depending on the corresponding GroupOperationalStatus value, this will be one of the following "All member devices are OK." "At least one member device is degraded." "At least one member device is Stressed." "At least one member device has Unknown Status."

Table 7: HP_ProcessorGroupHostedCollection supported properties. (Properties that are not supported are not mentioned.)

Instances of this class associate the Processor Collection with Computer System that contains it.

The getInstance() method is not supported for this association class.

Property name	Property inheritance	Property value (and data source)
CIM_ComputerSystem ref Antecedent	Property of HP_GroupHostedCollection	The reference to the CIM_ComputerSystem.
CIM_SystemSpecificCollection ref Dependent	Property of HP_GroupHostedCollection	The reference to the processor-collection of the system.

Table 8: HP_MemberOfProcessorCollection supported properties. (Properties that are not supported are not mentioned.)

This class associates the Processor/ProcessorChip instances to the ProcessorCollection to which they belong. These instances contribute to the GroupOperationalStatus of the processor-subsystem.

Property name	Property inheritance	Property value (and data source)
HP_ProcessorCollection REF Collection	Property of HP_GroupHostedCollection	Object path of HP_ProcessorCollection.
HP_Processor REF Member	Overridden by HP_GroupHostedCollection	Object path of HP_Processor

Table 9: intrinsic methods for all the CIM classes supported by CPU Instance Provider

Table 9 describes the intrinsic methods supported by this provider. It has three columns. The first is the method name, the second is a description of the provider's actions based on invoking that method, and the third is a list of any exceptions that could result from invoking the method. Each row describes a method.

Method name	description	exceptions thrown
enumerateInstances	Returns all instances of class with values of supported properties. (See tables above.)	CIMOperationFailedException
enumerateInstanceNames	Returns object path of all instances of class.	CIMOperationFailedException
getInstance	Returns an instance that matches the keys with values of supported properties. (See tables above.)	CIMOperationFailedException and CIMObjectNotFoundException
modifyInstance	This operation is not supported by the CPU Instance Provider. This is indicated to the client, via exceptions.	CIMNotSupportedException
deleteInstance	This operation is not supported by the CPU Instance Provider. This is indicated to the client, via exceptions.	CIMNotSupportedException
createInstance	This operation is not supported by the CPU Instance Provider. This is indicated to the client, via exceptions.	CIMNotSupportedException

Indications generated by this Provider	This Provider does not currently generate any indications.

Links to more information

WBEM information

For a CIM tutorial, go to http://www.wbemsolutions.com/tutorials/CIM/.

The System Fault Management Administrator's Guide is available at http://docs.hp.com/en/diag.

For additional information on HP products and services, visit us at http://www.hp.com.

For the location of the nearest sales office, call:

United States: +1 800 637 7740

Canada: +1 905 206 4725

Japan: +81 3 3331 6111

Latin America: +1 305 267 4220

Australia/New Zealand: +61 3 9272 2895

Asia Pacific: +8522 599 7777

Europe/Africa/Middle East: +41 22 780 81 11

For more information, contact any of our worldwide sales offices or HP Channel Partners (in the U.S., call 1 800 637 7740).

2/2008