hp WBEM solutions

hp technical data sheet

CPU Instance Provider

provider overview

description

The CPU Instance Provider is a Web Based Enterprise Management (WBEM) Instance Provider that fetches CPU-related information from HP PA-RISC Servers and HP Integrity Servers, running HPUX. This provider is compliant with the Common Information Model (CIM) 2.7.2 Schema, proposed by the Distributed Management Task Force (DMTF). The provider requires HP WBEM Services for HP-UX Version 2.0 installed on the system.

The CPU Instance Provider allows any client program, compliant with the CIM 2.7.2 Schema, to query for information about the managed system's processors.

The CPU Instance Provider implements the Processor-related CIM classes, proposed in the DMTF CIM 2.7.2 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 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) captures information about the physical location of the processor-chip 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.

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).

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 HP PA-RISC Server containing 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 HPUX kernel)
- 1 instance of HP_ProcessorChip (representing the single processor-chip FRU, i.e. the PA-RISC Processor-Chip)
- 1 instance of HPUX_ProcessorLocation (representing the slot occupied by the PA-RISC

processor-chip above)

- 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)).

The situation with IPF Processors 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 CPU Instance Provider is not a CIM Method Provider, and does not support extrinsic method invocation on instances on any of the MOF classes mentioned above. The invocation of any of these methods will result in a CIM_ERR_NOT_SUPPORTED exception.

requirements

The Provider requires

1. HP WBEM Services for hp-ux version A.02.00.09 or later.
2. OnlineDiag containing EMS V4.20 or later.
3. PHKL_32653, Cumulative diag2 driver and vPars enablement

release history

Starting with the March 2006 release, System Fault Management (SFM) will be available as part of the OE media.

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.

Please note that 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.

setting up this provider

installing this provider

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

Ensure the dependencies mentioned under the requirement section above are met.

Use swinstall to install the product: "Swinstall –s Fully_Qualified_Depot_Name SysFaultMgmt"

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 files, containing the definitions of the HP-specific MOF classes, (namely HPProcessor.mof & HP_ProcessorStatus.mof) will be available in /opt/sfm/schemas/mof. This directory will also include the provider registration file, namely SFMProvidersR.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, which is used for sending notifications to the CPU Instance Provider (e.g. the command to notify updation to the configuration file is "sfmconfig –c <filename>").

- The /opt/sfm/conf/ directory will contain the (XML) configuration files of the SysFaultMgmt Product.
- The /opt/sfm/msgcat/ 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 /opt/sfm/log/ directory will contain log files generated during the execution of the CPU Instance Provider.

The platforms supported by the CPU Instance Provider are listed in the Supported Platforms List.

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 – /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> <!-- Possible Values are INFORMATIONAL, WARNING, ERROR, CRITICAL, STOPLOGGING -->
      <Target> /opt/sfm/log/sfm.log </Target>
      <FileSize> 20480 </FileSize> <!-- sets the max. file size in KB. Min allowed value 2KB, Max allowed value , 1048576 KB (1 GB) -->
      <NBackupFiles> 3 </NBackupFiles> <!-- Number of files to roll over. Min allowed value 1, Max allowed value 10 -->
   </LoggerConfig>
</SFMConfig>
```

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

1. Edit the configuration file /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

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 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 path "/abc/def/" should already exist, and should be writeable by root-user.

2. Run /opt/sfm/bin/fmControl program, to specify the changed configuration file. i.e.

   $ /opt/sfm/bin/fmControl /opt/sfm/conf/FMLoggerConfig.xml

Note that the complete path of the configuration file must be provided to the fmControl 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 2.7.2 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 servers, this string is set to "This is a PA RISC Processor, with the following details: ", followed by location-details for the processor. For HP Integrity Servers, this string is set to "This is an Intel ® Itanium ® 2 Processor, with the following details: ", followed by the 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 (for PA-RISC Servers) or LID (for HP Integrity Servers).
string ElementName	Inherited from CIM_ManagedElement	For PA-RISC Servers, this string is set to "PA RISC Processor". For HP Integrity Servers, this string is set to "Intel ® Itanium ® 2 Processor".
String Name	Inherited from CIM_ManagedSystemElement	For PA-RISC Servers, this string is set to "PA RISC Processor". For HP Integrity Servers, this string is set to "Intel ® Itanium ® 2 Processor".
uint16 OperationalStatus []	Inherited from CIM_ManagedSystemElement	The Value-Map associated with this property (as per the CIM 2.7.2 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. The values returned in different scenarios are as follows: 1. If the processor is Configured and Active, the first element is 2 (for "OK").

		2. If the processor is Configured and Inactive, then the first element is 2 (for "OK") and the second element is 15(for "Dormant") 3. If the processor is Marked for configuration, then the first element is set to 2 (for "OK") and the second element is set to 8 (for "Starting") 4. If the processor is Marked for Deconfiguration and Active, then the first element is set to 2 (for "OK") and the second element is set to 9 (for "Stopping"). 5. If the processor is Marked for Deconfiguration and Inactive, then the first element is set to 2(for "OK"), the second element is set to 15 (for "Dormant") and the third element is set to 9 (for "Stopping"). 6. If the processor has errors, then the first element of this array is set to 6 (for "Error"). Note: Error information will be provided by consuming indications (of LPMC's) from the EMS Wrapper Provider. 7. If either the Deconfiguration status, or the Active status of the processor is unknown, or not retrievable, this array will contain a single element 0, indicating an "Unknown" status
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. The Description string corresponding to a value of "2" in the OperationalStatus array is: "Processor is OK". 2. The Description string corresponding to a value of "15" in the OperationalStatus array is: "Processor is idle. There are no processes scheduled on this processor." 3. The Description string corresponding to a value of "8" in the OperationalStatus array is: "Processor is currently deconfigured, and will be configured in the next reboot." 4. The Description string corresponding to a value of "9" in the OperationalStatus array is: "Processor is currently configured, and will be deconfigured in the next reboot." 5. The Description string corresponding to a value of "6" in the OperationalStatus array is: "Processor is in Error". 6. The Description string corresponding to a value of "0" in the OperationalStatus array is: "Processor is in an UNKNOWN state". For example, consider that the processor is Marked for Deconfiguration, and Active. The OperationalStatus array will be set such that: 1. OperationalStatus[0] = 2 2. OperationalStatus[1] = 9

		Then, the StatusDescriptions array will correspondingly be set such that: 1. StatusDescriptions[0] = “Processor is OK” 2. StatusDescriptions[1] = “Processor is currently configured, and will be deconfigured in the next reboot.”
string SystemCreationClassName [Key]	Inherited from CIM_LogicalDevice	Fixed string “PG_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 PA-RISC Servers, this is set to the Hard Physical Address of the processor. For HP Integrity Servers, this is set to the LID of the processor.
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”. For HP Integrity Servers, this property is set to “184”, to indicate “Intel ® Itanium ® 2”.
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.

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. For HP Integrity Servers, this string is set to “This is an Intel ® Itanium ® 2 Processor, with the following details: “, followed by the 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
string ElementName	Inherited from CIM_ManagedElement	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”.
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 CIM 2.7.2 Schema specifications) 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"} If the processor is Configured, the first element is 2 (for “OK”). If the processor is Marked for configuration, then the first element is set to 2 (for “OK”) and the second element is set to 8 (for “Starting”) If the processor is Marked for Deconfiguration and Active, then the first element is set to 2 (for “OK”) and the second element is set to 9 (for “Stopping”). If the processor has errors, then the first element of this array is set to 6 (for “Error”). Note: Error information will be provided by consuming indications (of LPMC’s) from the EMS Wrapper Provider. If the Deconfiguration status of the processor is unknown, or not retrievable, this array will contain a single element 0, indicating an “Unknown” status

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: The Description string corresponding to a value of "2" in the OperationalStatus array is: "Processor-Module is OK". The Description string corresponding to a value of "8" in the OperationalStatus array is: "The Processor-Module is currently deconfigured, and will be configured in the next reboot." The Description string corresponding to a value of "9" in the OperationalStatus array is: "The Processor-Module is currently configured, and will be deconfigured in the next reboot." The Description string corresponding to a value of "6" in the OperationalStatus array is: "Processor-Module has Error". The Description string corresponding to a value of "0" in the OperationalStatus array is: "Processor-Module is in an UNKNOWN state". For example, consider that the processor-chip is Marked for Deconfiguration. The OperationalStatus array will be set such that: OperationalStatus[0] = 2 OperationalStatus[1] = 9 Then, the StatusDescriptions array will correspondingly be set such that: StatusDescriptions[0] = "Processor-Module is OK" StatusDescriptions[1] = "The Processor-Module is currently configured, and will be deconfigured in the next reboot."
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)" For HP Integrity Servers, this contains a string of the format:

		"Intel (R) Itanium (R) 2 Processor (Family F Model M Stepping S)" where F: identifies the family of the processor (E.g. 31 for Itanium 2) M: indicates the model of the processor-chip S: indicates the stepping string of the processor-chip. E.g. For an rx7620 Server, the string could be: "Intel Itanium 2 (R) Processor (Family 31 Model 1 Stepping B1)"
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 5 for Intel ® Itanium ® 2
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.

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 “Location of Processor-Module”.
string Description	Inherited from CIM_ManagedElement.	For 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. For HP Integrity Servers, this string is set to “This is the location of an Intel ® Itanium ® 2 Processor, with the following details: “, followed by the 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	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”.
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.

table 4: HP_RealizesProcessor properties

Table 3 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 3 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: intrinsic methods for all the CIM classes supported by CPU Instance Provider

Table 6 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.)	
enumerateInstanceNames	Returns object path of all instances of class.	
getInstance	Returns an instance that matches the keys with values of supported properties. (See tables above.)	
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.dmtf.org/education/cimtutorial.php.

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).

9/2003