

# Dell EMC OpenManage CIM Reference Guide

Version 10.1.0.0

## Notes, cautions, and warnings

 **NOTE:** A NOTE indicates important information that helps you make better use of your product.

 **CAUTION:** A CAUTION indicates either potential damage to hardware or loss of data and tells you how to avoid the problem.

 **WARNING:** A WARNING indicates a potential for property damage, personal injury, or death.

|   |           |
|---|-----------|
| <b>Chapter 1: Introduction.....</b>               | <b>6</b>  |
| Server Administrator.....                         | 6         |
| Documenting CIM Classes and Their Properties..... | 6         |
| Base Classes.....                                 | 7         |
| Parent Classes.....                               | 7         |
| Classes That Describe Relationships.....          | 7         |
| Dell-Defined Classes.....                         | 7         |
| Common Properties of Classes.....                 | 8         |
| Other Documents You May Need.....                 | 9         |
| Typographical Conventions.....                    | 10        |
| <br>  |           |
| <b>Chapter 2: CIM_Physical Element.....</b>       | <b>11</b> |
| CIM_PhysicalElement.....                          | 11        |
| CIM_PhysicalPackage.....                          | 12        |
| CIM_PhysicalFrame.....                            | 13        |
| CIM_Chassis.....                                  | 14        |
| DELL_Chassis.....                                 | 14        |
| CIM_PhysicalComponent.....                        | 16        |
| CIM_Chip.....                                     | 16        |
| CIM_PhysicalMemory.....                           | 17        |
| CIM_PhysicalConnector.....                        | 20        |
| CIM_Slot.....                                     | 21        |
| <br>  |           |
| <b>Chapter 3: CIM_LogicalElement.....</b>         | <b>24</b> |
| CIM_LogicalElement.....                           | 25        |
| CIM_System.....                                   | 26        |
| CIM_ComputerSystem.....                           | 26        |
| DELL_System.....                                  | 27        |
| CIM_LogicalDevice.....                            | 27        |
| CIM_FRU.....                                      | 28        |
| CIM_Sensor.....                                   | 28        |
| CIM_DiscreteSensor.....                           | 29        |
| CIM_NumericSensor.....                            | 30        |
| CIM_TemperatureSensor.....                        | 32        |
| CIM_CurrentSensor.....                            | 33        |
| CIM_VoltageSensor.....                            | 33        |
| CIM_Tachometer.....                               | 34        |
| CIM_WatchDog.....                                 | 34        |
| CIM_CoolingDevice.....                            | 35        |
| CIM_Fan.....                                      | 36        |
| CIM_UserDevice.....                               | 36        |
| CIM_PointingDevice.....                           | 37        |
| CIM_Keyboard.....                                 | 38        |
| CIM_PowerSupply.....                              | 38        |

|   |           |
|---|-----------|
| CIM_Controller.....                         | 39        |
| CIM_ParallelController.....                 | 40        |
| CIM_SerialController.....                   | 41        |
| CIM_PCIController.....                      | 41        |
| CIM_PCIDevice.....                          | 42        |
| CIM_PCIBridge.....                          | 43        |
| CIM_Processor.....                          | 44        |
| CIM_StorageExtent.....                      | 51        |
| CIM_Memory.....                             | 51        |
| CIM_CacheMemory.....                        | 52        |
| DELL_SoftwareFeature.....                   | 53        |
| CIM_BIOSElement.....                        | 54        |
| CIM_SoftwareFeature.....                    | 54        |
| DELL_SoftwareFeature.....                   | 55        |
| CIM_SystemResource.....                     | 55        |
| CIM_IRQ.....                                | 56        |
| CIM_MemoryMappedIO.....                     | 57        |
| CIM_DMA.....                                | 58        |
| CIM_RedundancyGroup.....                    | 59        |
| CIM_ExtraCapacityGroup.....                 | 59        |
| DELL_PSRedundancyGroup.....                 | 60        |
| DELL_FanRedundancyGroup.....                | 60        |
| CIM_EnabledLogicalElement.....              | 61        |
| CIM_ServiceAccessPoint.....                 | 61        |
| CIM_RemoteServiceAccessPoint.....           | 61        |
| DELL_RemoteServiceAccessPort.....           | 62        |
| <b>Chapter 4: Dell-Defined Classes.....</b> | <b>64</b> |
| DELL_PostLog.....                           | 64        |
| DELL_BIOSExtensions.....                    | 65        |
| DELL_BIOSSettings.....                      | 65        |
| DELL_SDCardDevice.....                      | 66        |
| DELL_NetworkPort.....                       | 66        |
| DELL_PowerConsumptionAmpsSensor.....        | 69        |
| DELL_PowerConsumptionWattsSensor.....       | 70        |
| DELL_PowerConsumptionData.....              | 70        |
| DCIM_OEM_DataAccessModule.....              | 71        |
| DCIM_RegisteredProfile.....                 | 72        |
| <b>Chapter 5: CIM_Dependency.....</b>       | <b>73</b> |
| DELL_FanSensor.....                         | 73        |
| CIM_PackageTempSensor.....                  | 74        |
| CIM_PackageVoltSensor.....                  | 74        |
| CIM_PackageCurrentSensor.....               | 75        |
| CIM_PackageFanSensor.....                   | 75        |
| CIM_PackagePowerSupplySensor.....           | 76        |
| DELL_PackagePSRedundancy.....               | 76        |
| DELL_PSRedundancy.....                      | 76        |
| DELL_AssociatedSupplyPCAmps.....            | 77        |


|                                   |    |
|-----------------------------------|----|
| DELL_AssociatedSystemPCWatts..... | 77 |
| AssociatedSystemPCData.....       | 78 |
| DELL_PowerProfileData.....        | 78 |

# Introduction

This reference guide documents the OpenManage Server Administrator Common Information Model (CIM) provider contained in the Management Object File (MOF) **dccim32.mof**.

CIM provides a conceptual model for describing manageable objects in a systems management environment. CIM is a modeling tool rather than a programming language. CIM provides the structure for organizing objects into a model of a managed environment. For modeling a managed environment, CIM makes available a set of abstract and concrete classes of objects. These classes model the basic characteristics of systems, networks, and applications, as well as groupings of management-related data.

For more information about CIM, see the Distributed Management Task Force (DMTF) website at **dmtf.org** and the Microsoft website at **microsoft.com**.

 **NOTE:** This document includes sections or data that is not applicable to the PowerEdge MX740x and PowerEdge MX840c.

## Topics:

- [Server Administrator](#)
- [Documenting CIM Classes and Their Properties](#)
- [Common Properties of Classes](#)
- [Other Documents You May Need](#)
- [Typographical Conventions](#)

## Server Administrator

Server Administrator provides a suite of systems management information for keeping track of your networked systems. In addition to providing systems management agents that are independent of the management console, Server Administrator supports these systems management standards: CIM and Simple Network Management Protocol (SNMP).

In addition to supporting systems management industry standards, Server Administrator provides additional systems management information about the specific components of your Dell EMC device.

## Documenting CIM Classes and Their Properties

The Dell CIM provider extends support to Dell-specific software and hardware components. The Dell MOF defines the classes for the Dell CIM provider. All of the supported classes and properties in the MOF are documented in this guide.

The following subsections define some of the basic building blocks of CIM classes that are used in describing the dccim32 provider name. These subsections also explain how the elements used in describing these classes are organized. This section does not document the entire CIM schema, but only those classes and properties supported by the dccim32 provider. The list of properties for each supported class varies greatly.

The property values being presented could be NULL or empty string on some systems, although in general, some non-empty values can be expected. Key properties (listed below) always carry non-empty values. It is recommended that you use only the following properties as key attributes:

- **CIM\_PhysicalElement:** CreationClassName, Tag
- **CIM\_System:** CreationClassName, Name
- **CIM\_LogicalDevice:** SystemCreationClassName, SystemName, CreationClassName, DeviceID
- **CIM\_Dependency:** Antecedent, Dependent
- **CIM\_SoftwareElement:** Name, Version, SoftwareElementState, SoftwareElementID, TargetOperatingSystem
- **CIM\_SoftwareFeature:** IdentifyingNumber, ProductName, Vendor, Version, Name
- **CIM\_IRQ:** CSCreationClassName, CSName, CreationClassName, IRQNumber
- **CIM\_MemoryMappedIO:** CSCreationClassName, CSName, CreationClassName, StartingAddress

- CIM\_DMA: CSCreationClassName, CSName, CreationClassName, DMAChannel
- CIM\_RedundancyGroup: CreationClassName, Name
- DELL\_EsmLog: RecordNumber
- DELL\_PostLog: RecordNumber
- DELL\_BIOSExtensions: systemBIOSCharacteristics
- DELL\_BIOSSettings: DisplayName
- CIM\_ServiceAccessPoint: SystemCreationClassName, SystemName, CreationClassName, Name

## Base Classes

The classes listed in the Server Administrator CIM provider class hierarchy do not have a parent property. These base classes do not derive from another class. The base classes are:

- CIM\_ManagedSystemElement
- CIM\_Dependency
- DELL\_EsmLog
- DELL\_PostLog

The `CIM_ManagedSystemElement` class is the base class for the system element hierarchy from which all other CIM classes are derived. As a result, `CIM_ManagedSystemElement` has no parent. Examples of managed system elements include software components such as files, devices such as hard drives and controllers, and physical subcomponents of devices such as chip sets and cards. For the `CIM_ManagedSystemElement` properties, see [Caption](#), `CreationClassName`, `Description`, `Name`, and `Status` in [Common Properties of Classes](#)

The Dell-defined classes are not defined in the official schema by the DMTF, the industry group that defines the standards for CIM, and hence do not have parent classes. `CIM_Dependency` does not have a parent class because it is a relationship or association between two managed system elements.

## Parent Classes

Most classes in the `dcim32` provider document both a *Class Name* and a *Parent Class* property. The parent class is the class from which any given class inherits its core properties. For example, the `CIM_Controller` class has the `CIM_LogicalDevice` class as its parent, and has various types of controllers (`CIM_ParallelController`, `CIM_SerialController`) as its children.

## Classes That Describe Relationships

Classes that derive from `CIM_Dependency` have `CIM_Dependency` as their parent class, but they are documented in terms of *antecedent* and *dependent* elements in a relationship rather than in terms of common properties. Consider the following relationship between two `CIM_ManagedSystemElements`:

**Table 1. Classes That Describe Relationships**

| Element           | Description                           |
|-------------------|---------------------------------------|
| <b>Antecedent</b> | <code>CIM_PackageCurrentSensor</code> |
| <b>Dependent</b>  | <code>CIM_PhysicalPackage</code>      |

The `CIM_PackageCurrentSensor` class monitors an entire physical package, such as all the components contained in a given system chassis. The `CIM_PhysicalPackage` class is dependent on the `CIM_PackageCurrentSensor` class for this monitoring function.

## Dell-Defined Classes

Server Administrator has extended some CIM classes and has created new classes to assist in managing systems and their components. In this document, the illustrations of the classes created and populated by Dell are represented by an orange circle icon.

# Common Properties of Classes

Many classes have properties such as *Caption*, *Description*, and *CreationClassName*. [Common Properties of Classes](#) defines properties that have the same meaning in every class that has this property and are defined more than once in this guide.

**Table 2. Common Properties of Classes**

| Property                  | Description  | Data Type |
|---------------------------|--|-----------|
| Caption                   | Describes the object using a short textual description (one-line string).  | string    |
| CreationClassName         | Indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.                               | string    |
| CSCreationClassName       | Indicates the computer system's creation class name.   | string    |
| CSName                    | Indicates the computer system's name.  | string    |
| CurrentReading            | Indicates the actual current value indicated by the sensor in amperes.   | sint32    |
| Description               | Provides a textual description of the object.  | string    |
| LowerThresholdNonCritical | If current reading is between lower threshold noncritical and upper threshold noncritical, the current state is normal. See Figure 3-2.  | sint32    |
| LowerThresholdCritical    | If the current reading is between upper threshold critical and upper threshold fatal, the current state is critical. See Figure 3-2.   | sint32    |
| IsLinear                  | Indicates that the sensor is linear over its dynamic range.  | Boolean   |
| Manufacturer              | Provides the name of the organization responsible for producing the CIM_PhysicalElement or CIM_SoftwareElement. This may be the entity from whom the element is purchased, but not necessarily. Purchase information is contained in the vendor property of CIM_Product. | string    |
| Name                      | Defines the label by which the object is known. When subclassed, the <i>Name</i> property can be overridden to be a <i>Key</i> property.   | string    |
| Status                    | Provides a string indicating the status of the component. Status values include:<br>Operational Status Values:   | string    |

**Table 2. Common Properties of Classes (continued)**

| Property                  | Description   | Data Type |
|---------------------------|---|-----------|
|                           | <ul style="list-style-type: none"> <li>• <i>OK</i> indicates that the object is functioning normally.</li> <li>• <i>Degraded</i> means that the item is functioning, but not optimally.</li> <li>• <i>Stressed</i> indicates that the element is functioning, but needs attention. Examples of <i>Stressed</i> states are overloaded, overheated, and so on.</li> </ul> <p><b>Nonoperational Status Values:</b></p> <ul style="list-style-type: none"> <li>• <i>Non-recover</i> means that a nonrecoverable error has occurred.</li> <li>• <i>Error</i> means that an element has encountered an operational condition that is severe as compared to its normal mode of operation.</li> </ul> |           |
| SystemCreationClassName   | Indicates the system's creation class name.   | string    |
| UnitModifier              | Provides the unit multiplier for the values returned by this sensor. All the values returned by this sensor are represented in units of 10 raised to the power of the unit modifier. If the unit modifier is -6, then the units of the values returned are microvolts. The units apply to all numeric properties of the sensor, unless explicitly overridden by the units' qualifier.   | sint32    |
| UpperThresholdCritical    | If the current reading is between upper threshold critical and upper threshold fatal, the current status is critical. See Figure 3-2.   | sint32    |
| UpperThresholdNonCritical | If the current reading is between lower threshold noncritical and lower threshold critical, the current status is noncritical. See Figure 3-2.  | sint32    |
| Version                   | Version should be in the form <major>.<minor>.<revision> or <major>.<minor><letter><revision>; for example, 1.2.3 or 1.2a3.   | string    |

## Other Documents You May Need


Besides this *Dell EMC OpenManage Server Administrator CIM Reference Guide*, you can find the following documents on the Dell Support website at [dell.com/support/manuals](http://dell.com/support/manuals):

- *Dell EMC OpenManage Server Administrator User's Guide* documents the features, installation, and uninstallation of Server Administrator.
- *Dell EMC OpenManage Server Administrator Installation Guide* contains instructions to help you install Dell EMC OpenManage Server Administrator.
- *Dell EMC OpenManage Management Station Software Installation Guide* contains instructions to help you install Dell EMC OpenManage management station software that includes Baseboard Management Utility, iDRAC Tools, and Active Directory Snap-In.

- *Dell EMC OpenManage Server Administrator Command Line Interface User's Guide* explains how to perform tasks using the text-based command line interface.
- *Dell EMC OpenManage Server Administrator Messages Reference Guide* lists the messages that you can receive on your systems management console or on your operating system's event viewer. This guide explains the text, severity, and cause of each message that the Server Administrator issues.
- *Dell EMC OpenManage Server Administrator SNMP Reference Guide* documents the SNMP management information base (MIB). The SNMP MIB defines variables that cover the capabilities of Server Administrator systems management agents.
- The *Glossary* for information on terms used in this document.

## Typographical Conventions

The following example shows how most of the classes in the Dell CIM provider are documented. [CIM\\_DMA Properties](#) shows a partial class description for the DELL\_DMA class.

 **NOTE:** For a full class description, see [CIM\\_DMA Properties](#).

*Class Name* appears in Courier typeface and provides the string that names the class in the MOF.

*Parent Class* appears in Courier typeface and provides the name of the class from which the present class is derived.

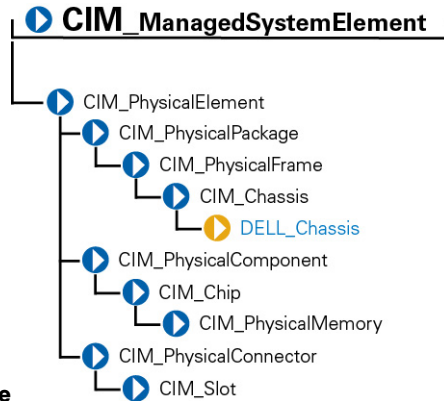
*Property* denotes the name of the attribute that is being defined for this class.

*Description* includes text that defines the property.

*Data Type* stipulates the format that the values of this property must take. Common data types include Boolean, string, and various types of integer. Boolean indicates that the property must be expressed as one of two alternatives.

# CIM\_PhysicalElement

`CIM_PhysicalElement` is a CIM-defined class. The `CIM_PhysicalElement` class contains the subclasses shown in Figure



**CIM\_PhysicalElement Class Structure**

## Topics:

- [CIM\\_PhysicalElement](#)
- [CIM\\_PhysicalPackage](#)
- [CIM\\_PhysicalFrame](#)
- [CIM\\_Chassis](#)
- [DELL\\_Chassis](#)
- [CIM\\_PhysicalComponent](#)
- [CIM\\_Chip](#)
- [CIM\\_PhysicalMemory](#)
- [CIM\\_PhysicalConnector](#)
- [CIM\\_Slot](#)

## CIM\_PhysicalElement



Subclasses of the `CIM_PhysicalElement` class listed in Table below define any component of a system that has a distinct physical identity. Physical elements are tangible managed system elements (usually actual hardware items) that have a physical manifestation of some sort. By contrast, processes, files, and logical devices are not classified as physical elements. A managed system element is not necessarily a discrete component. A single card (which is a type of physical element) can host more than one logical device.

One card, for example, could implement both a modem and a local area network (LAN) adapter. In this case, the card would be represented by a single physical element associated with multiple logical devices.

The following table lists the CIM Class Name, Parent Class and describes the CIM physical element properties along with the data type.

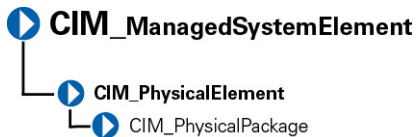
**Class Name:** `CIM_PhysicalElement`

**Parent Class:** `CIM_ManagedSystemElement`

**Table 3. CIM\_PhysicalElement Properties**

| Property          | Description   | Data Type |
|-------------------|---|-----------|
| CreationClassName | See <a href="#">Common Properties of Classes</a>  |           |
| Manufacturer      | See <a href="#">Common Properties of Classes</a>  |           |
| Model             | The name by which the physical element is generally known.  | string    |
| SerialNumber      | A manufacturer-allocated number used to identify the physical element.  | string    |
| Tag               | Uniquely identifies the physical element and serves as the element's key. The Tag property can contain information such as asset tag or serial number data. The key for a physical element is placed very high in the object hierarchy in order to identify the hardware/entity independently, regardless of the physical placement in or on cabinets, adapters, and so on. For example, a hotswappable or removable component can be taken from its containing (scoping) package and be temporarily unused. The object still continues to exist and may even be inserted into a different scoping container. Therefore, the key for the physical element is an arbitrary string that is defined independently of any placement or location-oriented hierarchy. | string    |

## CIM\_PhysicalPackage



The `CIM_PhysicalPackage` class listed in Table below represents physical elements that contain or host other components. Examples are a rack enclosure or an adapter card with multiple functions.

The following table lists the `CIM_PhysicalPackage` properties that represent the physical elements that contain or host other components.

**Class Name:** `CIM_PhysicalPackage`

**Parent Class:** `CIM_PhysicalElement`

**Table 4. CIM\_PhysicalPackage Properties**

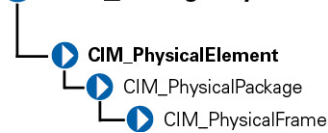
| Property    | Description   | Data Type |
|-------------|---|-----------|
| Removable   | A <code>CIM_PhysicalPackage</code> is removable if it is designed to be taken in and out of the physical container in which it is normally found without impairing the function of the overall package. | Boolean   |
| Replaceable | A <code>CIM_PhysicalPackage</code> is replaceable if it is possible to substitute a physically different element for the original element.  | Boolean   |

**Table 4. CIM\_PhysicalPackage Properties (continued)**

| Property | Description   | Data Type |
|----------|---|-----------|
|          | as in a field replaceable unit (FRU). For example, some computer systems allow the microprocessor to be upgraded to one of a higher clock rating. In this case, the microprocessor is said to be replaceable. |           |

## CIM\_PhysicalFrame

### ▶ CIM\_ManagedSystemElement



The `CIM_PhysicalFrame` class described in Table properties contains other frame enclosures such as racks and chassis. Properties like **VisibleAlarm** or **AudibleAlarm**, and data related to security breaches are also members of this class.

The following table describes the class, data type, and properties of `CIM_Physical Frame`.

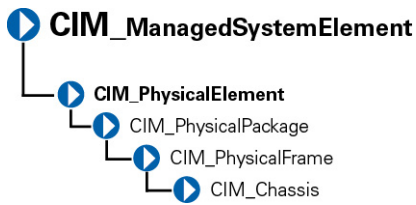
**Class Name:** `CIM_PhysicalFrame`

**Parent Class:** `CIM_PhysicalPackage`

**Table 5. CIM\_Physical Frame Properties**

| Property               | Description  | Data Type |
|------------------------|--|-----------|
| LockPresent            | Indicates whether the frame is protected with a lock.  | Boolean   |
| AudibleAlarm           | Indicates whether the frame is equipped with an audible alarm.   | Boolean   |
| VisibleAlarm           | Indicates that the equipment includes a visible alarm.   | Boolean   |
| SecurityBreach         | An enumerated, integer-valued property indicating that a physical breach of the frame is in progress. Values for the <b>SecurityBreach</b> property are:<br><b>1</b> - Other <b>2</b> - Unknown <b>3</b> - No breach <b>4</b> - Breach attempted <b>5</b> - Breach successful  | uint16    |
| SecurityBreachDriveBay | An enumerated, integer-valued property indicating that a physical breach of the frame is in progress. Values for the <b>SecurityBreachDriveBay</b> property are:<br><b>1</b> - Other <b>2</b> - Unknown <b>3</b> - No breach <b>4</b> - Breach attempted <b>5</b> - Breach successful <b>6</b> - Drive Bay open <b>7</b> - Drive Bay open extended | uint16    |
| IsLocked               | Indicates that the frame is currently locked.  | Boolean   |

# CIM\_Chassis



The `CIM_Chassis` class described in Table below represents the physical elements that enclose physical elements such as power supplies, fans, and processors.

The following table describes the class, data type, and properties of the `CIM_Chassis` Parent class.

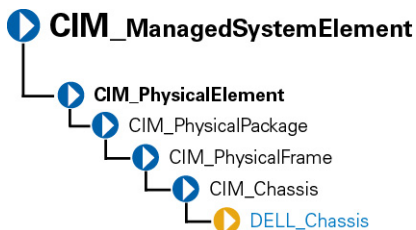
**Class Name:** `CIM_Chassis`

**Parent Class:** `CIM_PhysicalFrame`

**Table 6. CIM\_Chassis Parent Properties**

| Property     | Description   | Data Type |
|--------------|---|-----------|
| ChassisTypes | Values for the <b>ChassisTypes</b> property are: <ol style="list-style-type: none"> <li>1. Other</li> <li>2. Unknown</li> <li>3. Mini-tower</li> <li>4. Tower</li> <li>5. Space-saving</li> <li>6. Main system chassis</li> <li>7. Expansion chassis</li> <li>8. Subchassis</li> <li>9. Space-saving</li> <li>10. Main system chassis</li> <li>11. Expansion chassis</li> <li>12. Subchassis</li> <li>13. Bus expansion chassis</li> <li>14. Peripheral chassis</li> <li>15. Storage chassis</li> <li>16. Rack-mount chassis</li> </ol> | uint16    |

# DELL\_Chassis



The `DELL_Chassis` class explained in Table below defines the identifying and status properties of the chassis. `DELL_Chassis` inherits from CIM-defined classes, but is populated by Dell properties.

The following table describes the properties and data type of the DELL Chassis class.

**Class Name:** `DELL_Chassis`

**Parent Class:** `CIM_Chassis`

**Table 7. DELL\_Chassis Properties**

| Property                | Description   | Data Type |
|-------------------------|---|-----------|
| AssetTag                | Indicates the container <b>AssetTag</b> string. This asset tag string is assigned by the system administrator.  | string    |
| SystemClass             | Refers to the system type that is installed and running the instrumentation. Values for the <b>SystemClass</b> property are:<br><b>1</b> - Other <b>2</b> - Unknown <b>3</b> - Workstation <b>4</b> - Server <b>5</b> - Desktop <b>6</b> - Portable<br><b>7</b> - Net PC  | uint16    |
| SystemID                | Indicates the system identifier code.   | uint16    |
| LogFormat               | Defines whether the event log data is unicode formatted or binary (raw). Values for the event <b>LogFormat</b> property are:<br><b>1</b> - Formatted (event log only) <b>2</b> - Unformatted<br><b>3</b> - Events_and_POST_Formatted (both the event log and the power-on self-test (POST) log are unicode formatted) | uint16    |
| FanStatus               | Indicates the global status of fan sensors.   | string    |
| TempStatus              | Indicates the global status of temperature sensors.   | string    |
| VoltStatus              | Indicates the global status of voltage sensors.   | string    |
| AmpStatus               | Indicates the global status of current sensors.   | string    |
| PsStatus                | Indicates the global status of power supplies.  | string    |
| MemStatus               | Indicates the global status of memory devices.  | string    |
| ProcStatus              | Indicates the global status of processor devices.   | string    |
| FanRedStatus            | Indicates the global status of the cooling unit.  | string    |
| PsRedStatus             | Indicates the global status of the power unit.  | string    |
| IsDefaultThrSupported   | Indicates whether resetting default thresholds are supported.   | Boolean   |
| ChassisSystemProperties | Indicates chassis characteristics, such as energy smart and so on.  | uint16    |
| ChassisSystemRevision   | Indicates the chassis revision.   | uint16    |
| EsmLogStatus            | Indicates the global status of ESM log.   | string    |

**Table 7. DELL\_Chassis Properties (continued)**

| Property                  | Description                                       | Data Type |
|---------------------------|---|-----------|
| MemoryRedStatus           | Indicates the global status of memory redundancy. | string    |
| ChassisExpressServiceCode | Indicates the chassis express service code.       | string    |
| ChassisNodeID             | Chassis Node ID                                   | string    |

## CIM\_PhysicalComponent

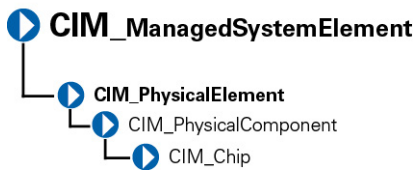


The CIM\_PhysicalComponent class listed in Table below represents any low-level or basic component within a package. A component object either cannot or does not need to be broken down into its constituent parts. For example, an application specific integrated circuit (ASIC) cannot be broken down into smaller discrete parts.

**Table 8. CIM\_PhysicalComponent Properties**

| Properties    |                       |
|---------------|-----------------------|
| Class Name:   | CIM_PhysicalComponent |
| Parent Class: | CIM_PhysicalElement   |

## CIM\_Chip



The CIM\_Chip class listed in [Chip Properties](#) represents any type of integrated circuit hardware, including ASICs, processors, memory chips, and so on.

The following table describes the properties and data type of the CIM Chip class.

**Class Name:** CIM\_Chip

**Parent Class:** CIM\_PhysicalComponent

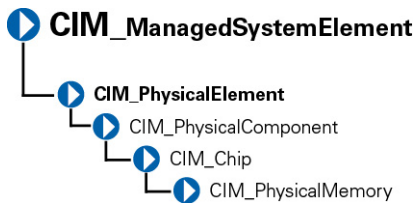
**Table 9. CIM\_Chip Properties**

| Property   | Description  | Data Type |
|------------|--|-----------|
| FormFactor | <b>0</b> — Unknown<br><b>1</b> — Other<br><b>2</b> — SIP<br><b>3</b> — DIP<br><b>4</b> — ZIP<br><b>5</b> — SOJ | uint16    |

**Table 9. CIM\_Chip Properties**

| Property | Description  | Data Type |
|----------|--|-----------|
|          | <b>6</b> — Proprietary<br><b>7</b> — SIMM<br><b>8</b> — DIMM<br><b>9</b> — TSOP<br><b>10</b> — PGA<br><b>11</b> — RIMM<br><b>12</b> — SODIMM<br><b>13</b> — SRIMM<br><b>14</b> — SMD<br><b>15</b> — SSMP<br><b>16</b> — QFP<br><b>17</b> — TQFP<br><b>18</b> — SOIC<br><b>19</b> — LCC<br><b>20</b> — PLCC<br><b>21</b> — BGA<br><b>22</b> — FPBGA<br><b>23</b> — LGA<br><b>24</b> — FB-DIMM |           |

## CIM\_PhysicalMemory



The `CIM_PhysicalMemory` class listed in [PhysicalMemoryProperties](#) is a subclass of `CIM_Chip`, representing low-level memory devices, such as SIMMs, DIMMs, and so on.

The following table describes the properties and data type of the CIM PhysicalMemory class.

**Class Name:** `CIM_PhysicalMemory`

**Parent Class:** `CIM_Chip`

**Table 10. CIM\_PhysicalMemory Properties**

| Property   | Description   | Data Type |
|------------|---|-----------|
| FormFactor | See <a href="#">Chip Properties</a>   | uint16    |
| MemoryType | Indicates the type of physical memory. Values for the MemoryType property are:<br><br><b>0</b> - Unknown<br><b>1</b> - Other<br><b>2</b> - DRAM | uint16    |

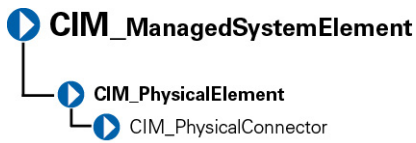
**Table 10. CIM\_PhysicalMemory Properties (continued)**

| Property         | Description  | Data Type |
|------------------|--|-----------|
|                  | <p><b>3</b> - Synchronous DRAM</p> <p><b>4</b> - Cache DRAM</p> <p><b>5</b> - EDO</p> <p><b>6</b> - EDRAM</p> <p><b>7</b> - VRAM</p> <p><b>8</b> - SRAM</p> <p><b>9</b> - RAM</p> <p><b>10</b> - ROM</p> <p><b>11</b> — Flash</p> <p><b>12</b> - EEPROM</p> <p><b>13</b> - FEPRAM</p> <p><b>14</b> - EPROM</p> <p><b>15</b> - CDRAM</p> <p><b>16</b> - 3DRAM</p> <p><b>17</b> - SDRAM</p> <p><b>18</b> - SGRAM</p> <p><b>19</b> - RDRAM</p> <p><b>20</b> - DDR</p> <p><b>21</b> - DDR2</p> <p><b>22</b> - DDR2 FB-DIMM</p> <p><b>24</b> - DDR3</p> <p><b>25</b> - FBD2</p> <p><b>26</b> - DDR4</p> |           |
| MemoryTechnology | <p>Defines the Memory Technology type of the Memory Device.</p> <p>The technology values are:</p> <p>01h - Other</p> <p>02h - Unknown</p> <p>03h - DRAM</p> <p>04h - NVDIMM-N</p> <p>05h - NVDIMM-F</p> <p>06h - NVDIMM-P</p> <p>07h - Intel persistent memory</p>   | uint32    |
| TotalWidth       | <p>Indicates the total width, in bits, of the physical memory, including check or error correction bits. If there are no error correction bits, the value in this property should match that specified for the <b>DataWidth</b> property.</p>  | uint16    |
| DataWidth        | <p>Indicates the data width, in bits, of the physical memory. A data width of 0 and a total width of 8 would indicate that the</p>   | uint16    |

**Table 10. CIM\_PhysicalMemory Properties (continued)**

| Property                     | Description  | Data Type |
|------------------------------|--|-----------|
|                              | memory is solely used to provide error correction bits.  |           |
| Speed                        | Indicates the speed of the physical memory, in nanoseconds.  | uint32    |
| Rank                         | The Rank values of DIMM are:<br><b>0</b> - Unknown<br><b>1</b> - Single<br><b>2</b> - Dual<br><b>4</b> - Quad<br><b>8</b> - Octal<br><b>16</b> - Hexa  | uint32    |
| SpeedAsString                | Indicates the accurate speed of the physical memory, in string format (with units).  | string    |
| Capacity                     | Indicates the total capacity of this physical memory, in bytes.  | uint64    |
| BankLabel                    | A string identifying the physically labeled bank where the memory is located, for example, "Bank 0" or "Bank A."   | string    |
| PositionInRow                | Specifies the position of the physical memory in a "row." For example, if it takes two 8-bit memory devices to form a 16-bit row, then a value of 2 means that this memory is the second device. 0 is an invalid value for this property.                          | uint32    |
| InterleavePosition           | Indicates the position of this physical memory in an interleave. 0 indicates noninterleaved. 1 indicates the first position, 2 the second position, and so on. For example, in a 2:1 interleave, a value of 1 indicates that the memory is in the "even" position. | uint32    |
| Manufacturer                 | Indicates the manufacturer of the physical memory.   | string    |
| NonVolatileSize              | The size of the Non-volatile portion of the memory device in Mega Bytes (MB)   | uint64    |
| VolatileSize                 | The size of the Volatile portion of the memory device in Mega Bytes (MB)   | uint64    |
| CacheSize                    | The size of the Cache portion of the memory device in Mega Bytes (MB)  | uint64    |
| RemainingRatedWriteEndurance | The Remaining life of persistent memory device in Percentage.  | uint32    |
| SerialNumber                 | Indicates the serial number of the physical memory.  | string    |

# CIM\_PhysicalConnector



The `CIM_PhysicalConnector` class listed in [Physical Connector Properties](#) includes physical elements such as plugs, jacks, or buses that connect physical elements. Any object that can be used to connect and transmit signals or power between two or more physical elements is a member of this class. For example, slots and D-shell connectors are types of physical connectors. See [Connector Type Values](#) for a list of valid connector type values.

The following table describes the properties and data type of the CIM PhysicalConnector class.

**Class Name:** `CIM_PhysicalConnector`

**Parent Class:** `CIM_PhysicalElement`

**Table 11. CIM\_PhysicalConnector Properties**

| Property        | Description  | Data Type |
|-----------------|--|-----------|
| ConnectorPinout | A free-form string describing the pin configuration and signal usage of a physical connector.  | string    |
| ConnectorType   | An array of integers defining the type of physical connector. An array is specified to allow the description of "combinations" of connector information. For example, one array entry could specify RS-232, another DB-25, and a third entry could define the connector as male. See <a href="#">Connector Type Values</a> for the values of the ConnectorType property. | uint16    |

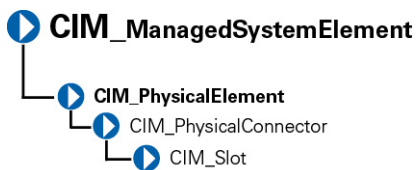
**Table 12. Connector Type Values**

| Connector Type Values               |                     |                              |                                  |
|-------------------------------------|---------------------|------------------------------|----------------------------------|
| 0 - Unknown                         | 30 - unused         | 60 - Micro-DIN               | 90 - On board IDE Connector      |
| 1 - Other                           | 31 - unused         | 61 - PS/2                    | 91 - On board floppy             |
| 2 - Male                            | 32 - IEEE-48        | 62 - Infrared                | 92 - 9 Pin dual inline           |
| 3 - Female                          | 33 - AUI            | 63 - unused                  | 93 - 25 Pin dual inline          |
| 4- Shielded                         | 34 - UTP Category 3 | 64 - Access bus              | 94 - 50 Pin dual inline          |
| 5 - Unshielded                      | 35 - UTP Category 4 | 65 - unused                  | 95 - 68 Pin dual inline          |
| 6 - SCSI (A) High-Density (50 pins) | 36 - UTP Category 5 | 66 - Centronics              | 96 - On board sound connector    |
| 7 - SCSI (A) Low-Density (50 pins)  | 37 - BNC            | 67 - Mini-Centronics         | 97 - Mini-jack                   |
| 8 - SCSI (P) High-Density (68 pins) | 38 - RJ11           | 68 - Mini-Centronics Type-14 | 98 - PCI-X                       |
| 9 - SCSI SCA-I (80 pins)            | 39 - RJ45           | 69 - Mini-Centronics Type-20 | 99 - Sbus IEEE 1396-1993 32-bit  |
| 10 - SCSI SCA-II (80 pins)          | 40 - Fiber MIC      | 70 - Mini-Centronics Type-26 | 100 - Sbus IEEE 1396-1993 64-bit |

**Table 12. Connector Type Values (continued)**

| Connector Type Values              |                     |                                      |                  |
|------------------------------------|---------------------|--------------------------------------|------------------|
| 11 - Fibre Channel (DB-9 Copper)   | 41 - unused         | 71 - Bus mouse                       | 101 - unused     |
| 12 - Fibre Channel (Fiber Optical) | 42 - unused         | 72 - ADB                             | 102 - GIO        |
| 13 - Fibre Channel SCAII (40 pins) | 43 - PCI            | 73 - AGP                             | 103 - XIO        |
| 14 - Fibre Channel SCAII (20 pins) | 44 - ISA            | 74 - VME bus                         | 104 - HIO        |
| 15 - Fibre Channel BNC             | 45 - unused         | 75 - VME64                           | 105 - NGIO       |
| 16 - ATA 3-1/2 inch (40 pins)      | 46 - VESA           | 76 - Proprietary                     | 106 - PMC        |
| 17 - ATA 2-1/2 inch (44 pins)      | 47 - unused         | 77 - Proprietary processor card slot | 107 - MTRJ       |
| 18 - ATA-2                         | 48 - unused         | 78 - Proprietary memory card slot    | 108 - VF-45      |
| 19 - ATA-3                         | 49 - unused         | 79 - Proprietary I/O riser slot      | 109 - Future I/O |
| 20 - ATA/66                        | 50 - unused         | 80 - PCI-66 MHz                      | 110 - SC         |
| 21 - DB-9                          | 51 - unused         | 81 - AGP2X                           | 111 - SG         |
| 22 - DB-15                         | 52 - unused         | 82 - AGP4X                           | 112 - Electrical |
| 23 - DB-25                         | 53 - USB            | 83 - PC-98                           | 113 - Optical    |
| 24 - DB-36                         | 54 - IEEE 1394      | 84 - PC-98-Hireso                    | 114 - Ribbon     |
| 25 - RS-232C                       | 55 - HIPPI          | 85 - PC-H98                          | 115 - GLM        |
| 26 - RS-422                        | 56 - HSSDC (6 pins) | 86 - PC-98Note                       | 116 - 1x9        |
| 27 - RS-423                        | 57 - GBIC           | 87 - PC-98Full                       | 117 - Mini SG    |
| 28 - RS-485                        | 58 - DIN            | 88 - SSA SCSI                        | 118 - LC         |
| 29 - RS-449                        | 59 - Mini-DIN       | 89 - Circular                        | 119 - HSSC       |

## CIM\_Slot



The `CIM_Slot` class that is listed in Table below represents connectors into which packages are inserted. For example, a physical package that is a hard drive can be inserted into a small computer system interface-single connector attachment (SCSI-SCA) slot. As another example, a card can be inserted into a 16-, 32-, or 64-bit expansion slot on a host board.

The following table describes the properties and data type of the CIM Slot class.

**Class Name:** `class CIM_Slot`

**Parent Class:** `CIM_PhysicalConnector`

**Table 13. CIM\_Slot Properties**

| Property      | Description                               | Data Type |
|---------------|---|-----------|
| ConnectorType | See <a href="#">Connector Type Values</a> | uint16    |

**Table 13. CIM\_Slot Properties (continued)**

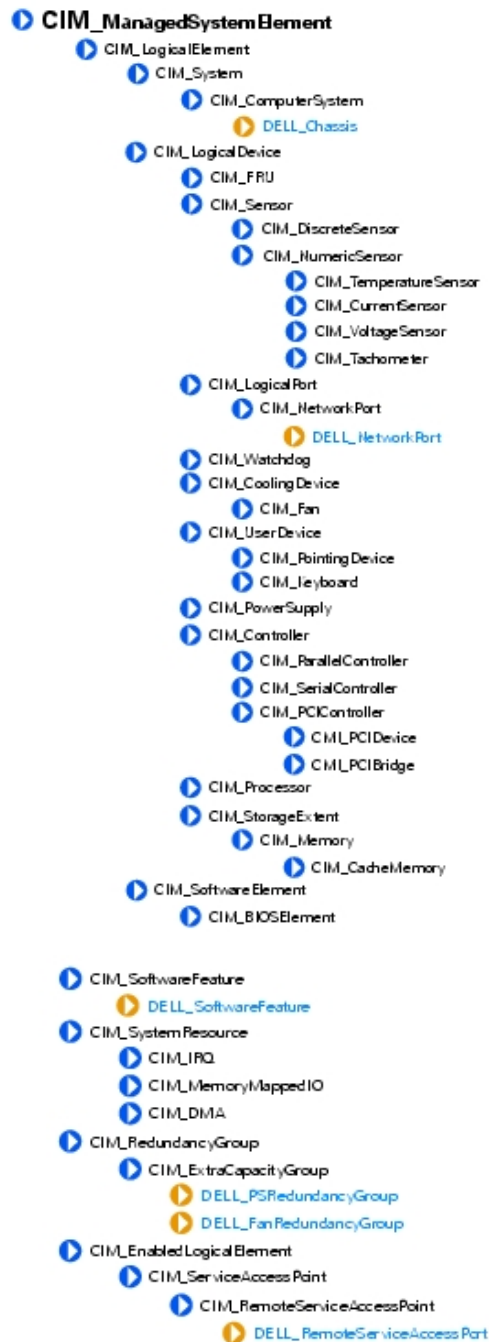
| Property        | Description  | Data Type |
|-----------------|--|-----------|
| SupportsHotPlug | Indicates whether the slot supports hot-plug adapter cards.  | Boolean   |
| MaxDataWidth    | <p>Indicates the maximum bus width in bits of adapter cards that can be inserted into this slot. Values for the MaxDataWidth property are as follows:</p> <ul style="list-style-type: none"> <li><b>0</b> - Unknown</li> <li><b>1</b> - Other</li> <li><b>8</b> - bits</li> <li><b>16</b> - bits</li> <li><b>32</b> - bits</li> <li><b>64</b> - bits</li> <li><b>128</b> - bits</li> </ul>   | uint16    |
| SystemSlotType  | <p>Indicates the type of system slot. Values for the <b>SystemSlotType</b> property are as follows:</p> <ul style="list-style-type: none"> <li><b>1</b> - Other</li> <li><b>2</b> - Unknown</li> <li><b>3</b> - ISA</li> <li><b>4</b> - MCA</li> <li><b>5</b> - EISA</li> <li><b>6</b> - PCI</li> <li><b>7</b> - PCMCIA</li> <li><b>8</b> - VL-VESA</li> <li><b>9</b> - Proprietary</li> <li><b>10</b> - Processor card Slot</li> <li><b>11</b> - Proprietary memory card slot</li> <li><b>12</b> - I/O Riser card slot</li> <li><b>13</b> - NuBus</li> <li><b>14</b> - PCI - 66MHz capable</li> <li><b>15</b> - AGP</li> <li><b>16</b> - AGP 2X</li> <li><b>17</b> - AGP 4X</li> <li><b>18</b> - PCI-X</li> <li><b>19</b> - AGP 8X</li> <li><b>20</b> - M.2 Socket 1-DP (Mechanical Key A)</li> <li><b>21</b> - M.2 Socket 1-SD (Mechanical Key E)</li> <li><b>22</b> - M.2 Socket 2 (Mechanical Key B)</li> <li><b>23</b> - M.2 Socket 3 (Mechanical Key M)</li> <li><b>24</b> - MXM Type I</li> <li><b>25</b> - MXM Type II</li> <li><b>26</b> - MXM Type III (standard connector)</li> </ul> |           |

**Table 13. CIM\_Slot Properties (continued)**

| Property       | Description  | Data Type |
|----------------|--|-----------|
|                | <p><b>27</b> - MXM Type III (HE connector)<br/> <b>28</b> - MXM Type IV<br/> <b>29</b> - MXM 3.0 Type A<br/> <b>30</b> - MXM 3.0 Type B<br/> <b>31</b> - PCI Express Gen 2 SFF-8639<br/> <b>32</b> - PCI Express Gen 3 SFF-8639<br/> <b>33</b> - PCI Express Mini 52-pin with bottom-side keep-outs<br/> <b>34</b> - PCI Express Mini 52-pin without bottom-side keep-outs<br/> <b>35</b> - PCI Express Mini 76-pin</p>  |           |
| SystemSlotType | <p><b>160</b> - PC-98/C20<br/> <b>161</b> - PC-98/C24<br/> <b>162</b> - PC-98/E<br/> <b>163</b> - PC-98/Local bus<br/> <b>164</b> - PC-98/Card<br/> <b>165</b> - PCI Express<br/> <b>166</b> - PCI Express x1<br/> <b>167</b> - PCI Express x2<br/> <b>168</b> - PCI Express x4<br/> <b>169</b> - PCI Express x8<br/> <b>170</b> - PCI Express x16<br/> <b>171</b> - PCI Express Gen 2<br/> <b>172</b> - PCI Express Gen 2 x1<br/> <b>173</b> - PCI Express Gen 2 x2<br/> <b>174</b> - PCI Express Gen 2 x4<br/> <b>175</b> - PCI Express Gen 2 x8<br/> <b>176</b> - PCI Express Gen 2 x16<br/> <b>177</b> - PCI Express Gen 3<br/> <b>178</b> - PCI Express Gen 3 x1<br/> <b>179</b> - PCI Express Gen 3 x2<br/> <b>180</b> - PCI Express Gen 3 x4<br/> <b>181</b> - PCI Express Gen 3 x8<br/> <b>182</b> - PCI Express Gen 3 x16<br/> <b>184</b> - PCI Express Gen 4<br/> <b>185</b> - PCI Express Gen 4 x1<br/> <b>186</b> - PCI Express Gen 4 x2<br/> <b>187</b> - PCI Express Gen 4 x4<br/> <b>188</b> - PCI Express Gen 4 x8<br/> <b>189</b> - PCI Express Gen 4 x16</p> |           |

# CIM\_LogicalElement

CIM\_LogicalElement is a CIM-defined class containing the subclasses described in below Figure.



## Topics:

- [CIM\\_LogicalElement](#)
- [CIM\\_System](#)
- [CIM\\_ComputerSystem](#)
- [DELL\\_System](#)

- CIM\_LogicalDevice
- CIM\_FRU
- CIM\_Sensor
- CIM\_DiscreteSensor
- CIM\_NumericSensor
- CIM\_TemperatureSensor
- CIM\_CurrentSensor
- CIM\_VoltageSensor
- CIM\_Tachometer
- CIM\_WatchDog
- CIM\_CoolingDevice
- CIM\_Fan
- CIM\_UserDevice
- CIM\_PointingDevice
- CIM\_Keyboard
- CIM\_PowerSupply
- CIM\_Controller
- CIM\_ParallelController
- CIM\_SerialController
- CIM\_PCIController
- CIM\_PCIDevice
- CIM\_PCIBridge
- CIM\_Processor
- CIM\_StorageExtent
- CIM\_Memory
- CIM\_CacheMemory
- DELL\_SoftwareFeature
- CIM\_BIOSElement
- CIM\_SoftwareFeature
- DELL\_SoftwareFeature
- CIM\_SystemResource
- CIM\_IRQ
- CIM\_MemoryMappedIO
- CIM\_DMA
- CIM\_RedundancyGroup
- CIM\_ExtraCapacityGroup
- DELL\_PSRedundancyGroup
- DELL\_FanRedundancyGroup
- CIM\_EnabledLogicalElement
- CIM\_ServiceAccessPoint
- CIM\_RemoteServiceAccessPoint
- DELL\_RemoteServiceAccessPort

## CIM\_LogicalElement



Table properties list the following characteristics for members of the `CIM_LogicalElement` class:

- Represent abstractions used to manage and coordinate aspects of a physical environment such as files, processes, systems, system capabilities, and network components in the form of logical devices
- Represent devices, where devices are abstractions of hardware entities that may or may not be realized in physical hardware

**Table 14. CIM\_LogicalElement Properties**

| Properties           |                          |
|----------------------|--------------------------|
| <b>Class Name:</b>   | CIM_LogicalElement       |
| <b>Parent Class:</b> | CIM_ManagedSystemElement |

## CIM\_System



The CIM\_System class described in Table below defines a collection of managed system elements that operates as a functional whole. An instance of the CIM\_System class contains a well-defined list of components that work together to perform a specific function.

The following table describes the properties and data type of the CIM System class.

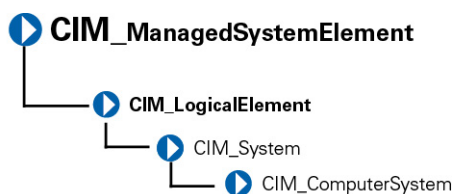
**Class Name:** CIM\_System

**Parent Class:** CIM\_LogicalElement

**Table 15. CIM\_System Properties**

| Property            | Description  | Data Type |
|---------------------|--|-----------|
| CreationClassName   | See <a href="#">Common Properties of Classes</a>   | string    |
| Name                | Indicates the name of a specific system, such as a particular storage system or server.  | string    |
| PrimaryOwnerContact | Provides information about how the primary system owner can be reached, for example, a phone number or e-mail address.   | string    |
| PrimaryOwnerName    | Indicates the name of the primary system owner.  | string    |
| Roles               | An array of strings that specifies the roles this system plays in the IT environment. For example, for an instance of a network system, the <b>Roles</b> property might contain the string "storage system." | string    |

## CIM\_ComputerSystem

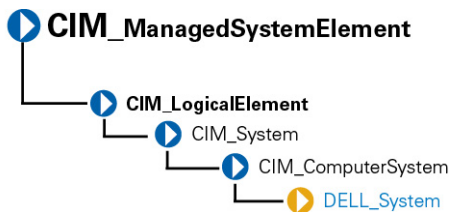


The CIM\_ComputerSystem class described in Table below contains some or all of the following CIM\_ManagedSystemElements: file system, operating system, processor, and memory (volatile and/or nonvolatile storage). For properties, see [CIM\\_System Properties](#).

**Table 16. CIM\_ComputerSystem Properties**

| Properties           |                    |
|----------------------|--------------------|
| <b>Class Name:</b>   | CIM_ComputerSystem |
| <b>Parent Class:</b> | CIM_System         |

## DELL\_System



The DELL\_System class described in Table below is the set of all Dell instrumented systems, including server, and storage systems. For properties, see [CIM\\_System Properties](#)

**Table 17. DELL\_System Properties**

| Properties           |                    |
|----------------------|--------------------|
| <b>Class Name:</b>   | DELL_System        |
| <b>Parent Class:</b> | CIM_ComputerSystem |

## CIM\_LogicalDevice



The CIM\_LogicalDevice class described below Table models a hardware entity that may be realized in physical hardware. CIM\_LogicalDevice includes any characteristics of a logical device that manages its operation or configuration. An example of a logical device is a temperature sensor’s reading of the actual temperature.

The following table describes the properties and data type of the CIM Logical Device class.

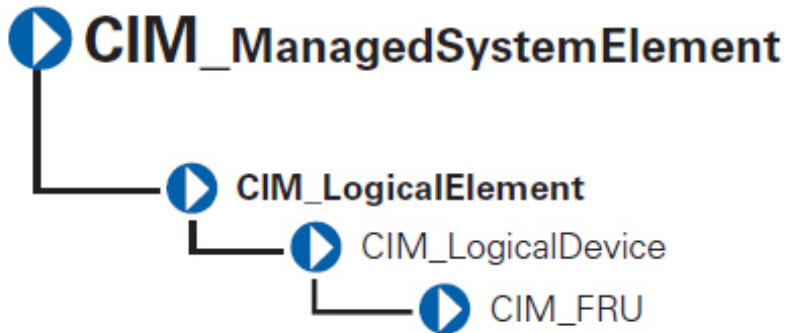
**Class Name:** CIM\_LogicalDevice

**Parent Class:** CIM\_LogicalElement

**Table 18. CIM\_Logical Device Properties**

| Property                | Description   | Data Type |
|-------------------------|---|-----------|
| SystemCreationClassName | See <a href="#">Common Properties of Classes</a>  | string    |
| SystemName              | Indicates the scoping system’s name.  | string    |
| CreationClassName       | See <a href="#">Common Properties of Classes</a>  | string    |
| DeviceID                | Identifies an address or other identifying information to uniquely name the logical device. | string    |

# CIM\_FRU



The `CIM_FRU` class described in [FRU Properties](#) contains manufacturing information related to the Field Replaceable Units (FRU) of a system such as a system planar or I/O riser card.

The following table describes the properties and data type of the CIM FRU class.

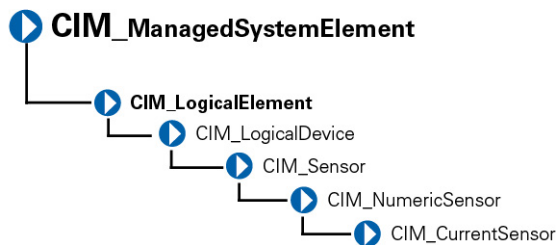
**Class Name:** `CIM_FRU`

**Parent Class:** `CIM_LogicalDevice`

**Table 19. CIM\_FRU Properties**

| Property                 | Description  | Data Type |
|--------------------------|--|-----------|
| FRUInformationState      | Indicates the state and availability of FRU information. | uint 16   |
| FRUDeviceName            | Indicates the device name of the FRU.                    | string    |
| FRUManufacturingDateName | Indicates the manufacturing date of the FRU in ticks.    | datetime  |
| FRUManufacturerName      | Indicates the name of the manufacturer.                  | string    |
| FRUPartNumberName        | Indicates the FRU part number.                           | string    |
| FRUSerialNumberName      | Indicates the FRU serial number.                         | string    |
| FRURevisionName          | Indicates the FRU revision number.                       | string    |

# CIM\_Sensor



The `CIM_Sensor` class described in Table below contains hardware devices capable of measuring the characteristics of some physical property, for example, the temperature or voltage characteristics of a computer system

The following table describes the properties and data type of the CIM Sensor class.

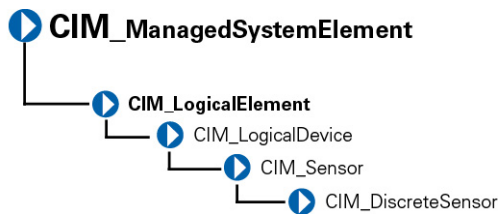
**Class Name:** `CIM_Sensor`

**Parent Class:** `CIM_LogicalDevice`

**Table 20. CIM\_Sensor Properties**

| Property        | Description   | Data Type |
|-----------------|---|-----------|
| SensorType      | Indicates the type of the sensor, for example, voltage or temperature sensor.<br>Values for the SensorType property are:<br><b>0</b> - Unknown<br><b>1</b> - Other<br><b>2</b> - Temperature sensors measure the environmental temperature.<br><b>3</b> - Voltage sensors measure electrical voltage.<br><b>4</b> - Current sensors measure current readings.<br><b>5</b> - Tachometers measure speed/ revolutions of a device. For example, a fan device can have an associated tachometer that measures its speed.<br><b>6</b> - Batteries maintain the time and date and save the system's BIOS configuration when the system is turned off. | uint16    |
| OtherSensorType | Description Indicates the type of sensor when the SensorType property is set to <b>Other</b> .  | string    |
| PossibleStates  | Enumerates the string outputs of the sensor. For example, a NumericSensor can report states based on threshold readings.  | string    |
| CurrentState    | Indicates the current state of the sensor. This value is always one of the Possible States.   | string    |
| PollingInterval | Indicates the polling interval, in nanoseconds, that the sensor hardware or instrumentation uses to determine the current state of the sensor.  | uint64    |

## CIM\_DiscreteSensor



The `CIM_DiscreteSensor` class described in Table below has a set of legal string values that it can report. The `CIM_DiscreteSensor` always has a "current reading" that corresponds to one of the enumerated values.

The following table describes the properties and data type of the CIM Discrete Sensor class.

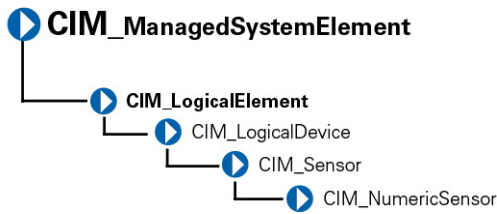
**Class Name:** `CIM_DiscreteSensor`

**Parent Class:** `CIM_Sensor`

**Table 21. CIM\_DiscreteSensor Properties**

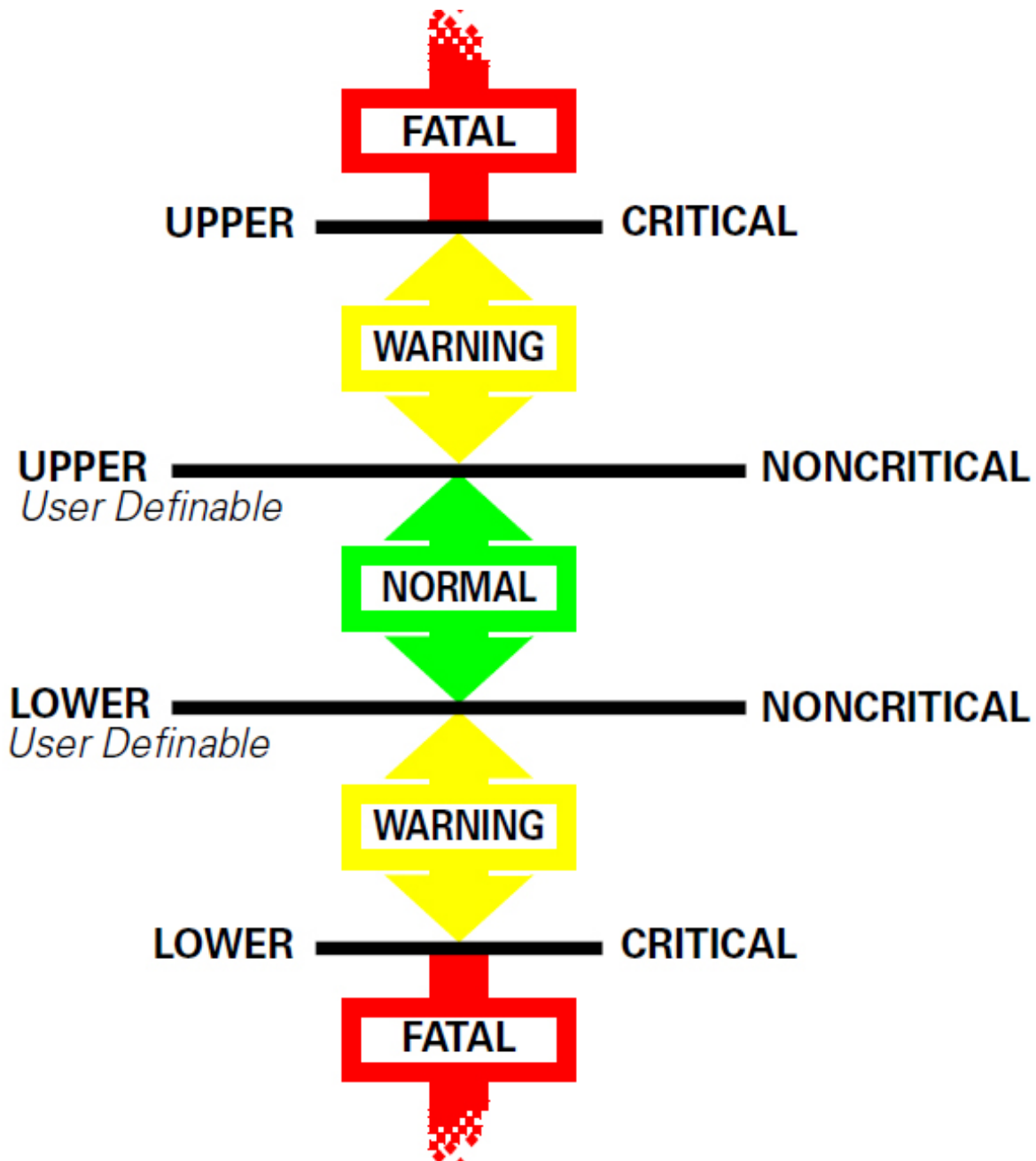
| Property       | Description   | Data Type |
|----------------|---|-----------|
| CurrentReading | See <a href="#">Common Properties of Classes</a>                  | sint32    |
| PossibleValues | Enumerates the string outputs that can be reported by the sensor. | sint32    |

## CIM\_NumericSensor



The `CIM_NumericSensor` class described in [NumericSensor Properties](#) returns numeric settings and may also support threshold settings. Figure below shows the relationship among upper and lower critical and upper and lower non-critical threshold values. The normal range falls between upper and lower non-critical thresholds.

**Figure: Ranges for Threshold Values**



The following table describes the properties and data type of the CIM Numeric Sensor class.

**Class Name:** CIM\_NumericSensor

**Parent Class:** CIM\_Sensor

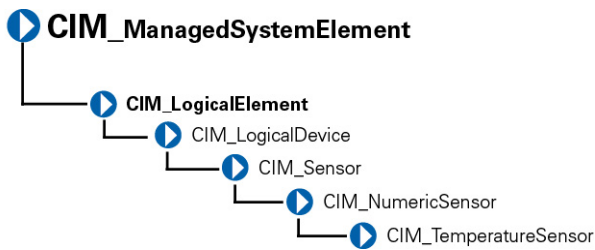
**Table 22. CIM\_NumericSensor Properties**

| Property                  | Description                                      | Data Type |
|---------------------------|--|-----------|
| UnitModifier              | See <a href="#">Common Properties of Classes</a> | sint32    |
| CurrentReading            | See <a href="#">Common Properties of Classes</a> | sint32    |
| IsLinear                  | See <a href="#">Common Properties of Classes</a> | Boolean   |
| LowerThresholdNonCritical | See <a href="#">Common Properties of Classes</a> | sint32    |
| UpperThresholdNonCritical | See <a href="#">Common Properties of Classes</a> | sint32    |
| LowerThresholdCritical    | See <a href="#">Common Properties of Classes</a> | sint32    |

**Table 22. CIM\_NumericSensor Properties (continued)**

| Property               | Description  | Data Type |
|------------------------|--|-----------|
| UpperThresholdCritical | See <a href="#">Common Properties of Classes</a>   | sint32    |
| SupportedThresholds    | An array representing the thresholds supported by this sensor. The supported values are as follows:<br>1 - LowerThresholdNonCritical<br>2 - UpperThresholdNonCritical<br>3 - LowerThresholdCritical<br>4 - UpperThresholdCritical                          | uint16    |
| EnabledThresholds      | An array representing the thresholds that are currently enabled for this sensor.<br>Enabled threshold values are as follows:<br>1 - LowerThresholdNonCritical<br>2 - UpperThresholdNonCritical<br>3 - LowerThresholdCritical<br>4 - UpperThresholdCritical | uint16    |
| SettableThresholds     | An array representing the writable thresholds supported by this sensor.<br>Settable threshold values are:<br>1 - LowerThresholdNonCritical<br>2 - UpperThresholdNonCritical  | uint16    |

## CIM\_TemperatureSensor



The `CIM_TemperatureSensor` class described in Table below contains sensors that sample ambient temperature and return a value in degrees celsius.

The following table describes the properties and data type of the CIM Temperature Sensor class.

**Class Name:** `CIM_TemperatureSensor`

**Parent Class:** `CIM_NumericSensor`

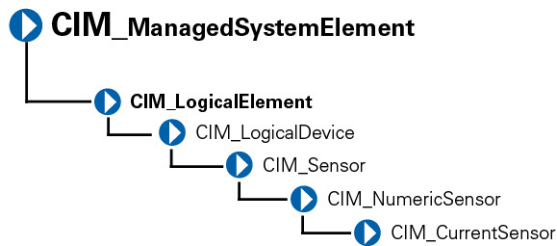
**Table 23. CIM\_TemperatureSensor Properties**

| Property                  | Description                                      | Data Type |
|---------------------------|--|-----------|
| UnitModifier              | See <a href="#">Common Properties of Classes</a> | sint32    |
| CurrentReading            | See <a href="#">Common Properties of Classes</a> | sint32    |
| IsLinear                  | See <a href="#">Common Properties of Classes</a> | Boolean   |
| LowerThresholdNonCritical | See <a href="#">Common Properties of Classes</a> | sint32    |

**Table 23. CIM\_TemperatureSensor Properties (continued)**

| Property                  | Description                                      | Data Type |
|---------------------------|--|-----------|
| UpperThresholdNonCritical | See <a href="#">Common Properties of Classes</a> | sint32    |
| LowerThresholdCritical    | See <a href="#">Common Properties of Classes</a> | sint32    |
| UpperThresholdCritical    | See <a href="#">Common Properties of Classes</a> | sint32    |

## CIM\_CurrentSensor



The `CIM_CurrentSensor` class described in Properties Table below contains sensors that measure amperage and return a value in amperes and watts.

The following table describes the properties and data type of the CIM Current Sensor class.

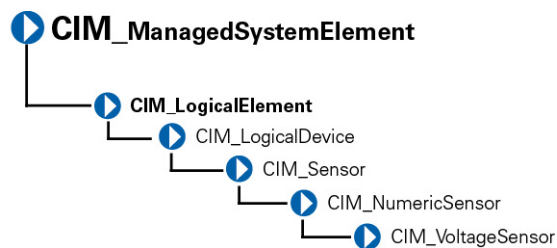
**Class Name:** `CIM_CurrentSensor`

**Parent Class:** `CIM_NumericSensor`

**Table 24. CIM\_CurrentSensor Properties**

| Property                  | Description                                      | Data Type |
|---------------------------|--|-----------|
| UnitModifier              | See <a href="#">Common Properties of Classes</a> | sint32    |
| CurrentReading            | See <a href="#">Common Properties of Classes</a> | sint32    |
| IsLinear                  | See <a href="#">Common Properties of Classes</a> | Boolean   |
| LowerThresholdNonCritical | See <a href="#">Common Properties of Classes</a> | sint32    |
| UpperThresholdNonCritical | See <a href="#">Common Properties of Classes</a> | sint32    |
| LowerThresholdCritical    | See <a href="#">Common Properties of Classes</a> | sint32    |
| UpperThresholdCritical    | See <a href="#">Common Properties of Classes</a> | sint32    |

## CIM\_VoltageSensor



The `CIM_VoltageSensor` class described in Table below contains sensors that measure voltage and return a value in volts.

The following table describes the properties and data type of the CIM Voltage Sensor class.

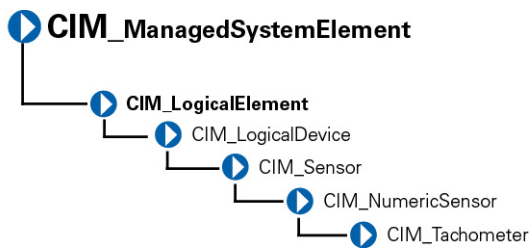
**Class Name:** `CIM_VoltageSensor`

**Parent Class:** `CIM_NumericSensor`

**Table 25. CIM\_VoltageSensor Properties**

| Property                  | Description                                      | Data Type |
|---------------------------|--|-----------|
| UnitModifier              | See <a href="#">Common Properties of Classes</a> | sint32    |
| CurrentReading            | See <a href="#">Common Properties of Classes</a> | sint32    |
| IsLinear                  | See <a href="#">Common Properties of Classes</a> | Boolean   |
| LowerThresholdNonCritical | See <a href="#">Common Properties of Classes</a> | sint32    |
| UpperThresholdNonCritical | See <a href="#">Common Properties of Classes</a> | sint32    |
| LowerThresholdCritical    | See <a href="#">Common Properties of Classes</a> | sint32    |
| UpperThresholdCritical    | See <a href="#">Common Properties of Classes</a> | sint32    |

## CIM\_Tachometer



The `CIM_Tachometer` class described in Table below contains devices that measure revolutions per minute (RPM) of a fan and return the value in RPMs.

The following table describes the properties and data type of the CIM Tachometer class.

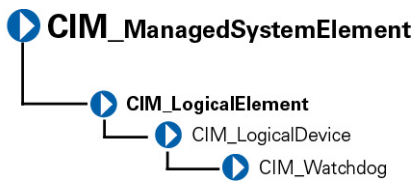
**Class Name:** `CIM_Tachometer`

**Parent Class:** `CIM_NumericSensor`

**Table 26. CIM\_Tachometer Properties**

| Property                  | Description                                      | Data Type |
|---------------------------|--|-----------|
| SensorType                | See <a href="#">Common Properties of Classes</a> | uint16    |
| UnitModifier              | See <a href="#">Common Properties of Classes</a> | sint32    |
| CurrentReading            | See <a href="#">Common Properties of Classes</a> | sint32    |
| IsLinear                  | See <a href="#">Common Properties of Classes</a> | Boolean   |
| LowerThresholdNonCritical | See <a href="#">Common Properties of Classes</a> | sint32    |
| UpperThresholdNonCritical | See <a href="#">Common Properties of Classes</a> | sint32    |

## CIM\_WatchDog



The `CIM_WatchDog` class described in Table below represents a timer that is implemented in system hardware. The watchdog feature allows the hardware to monitor the state of the operating system, BIOS, or a software component installed on the system. If the monitored component fails to rearm the timer before its expiration, the hardware assumes that the system is in

a critical state and could reset the system. This feature can also be used as an application watchdog timer for a mission-critical application. In this case, the application would assume responsibility for rearming the timer before expiration.

The following table describes the properties and data type of the CIM WatchDog class.

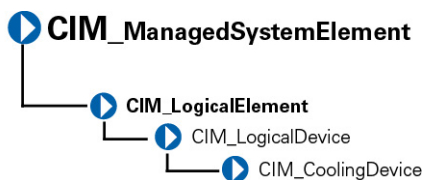
**Class Name:** CIM\_WatchDog

**Parent Class:** CIM\_LogicalDevice

**Table 27. CIM\_WatchDog Properties**

| Property                    | Description  | Data Type |
|-----------------------------|--|-----------|
| MonitoredEntity             | Indicates the entity that is currently being monitored by the watchdog feature. This property is used to identify the module that is responsible for rearming the watchdog at periodic intervals. Values for the <b>MonitoredEntity</b> property are:<br><br><b>1</b> - Unknown<br><b>2</b> - Other<br><b>3</b> - Operating System | uint16    |
| MonitoredEntity Description | A string describing additional textual information about the monitored entity.   | string    |
| TimeoutInterval             | Indicates the time-out interval used by the watchdog, in microseconds.   | uint32    |
| TimerResolution             | Indicates the resolution of the watchdog timer. For example, if this value is 100, then the timer can expire anytime between -100 microseconds and +100 microseconds.  | uint32    |

## CIM\_CoolingDevice



The CIM\_CoolingDevice class described in [CIM\\_CoolingDevice](#) contains a set of devices that work to keep the ambient internal temperature of the system at a safe value.

The following table describes the properties and data type of the CIM Cooling Device class.

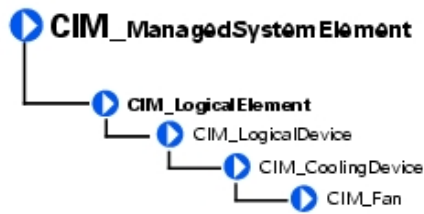
**Class Name:** CIM\_CoolingDevice

**Parent Class:** CIM\_LogicalDevice

**Table 28. CIM\_CoolingDevice Properties**

| Property      | Description   | Data Type |
|---------------|---|-----------|
| ActiveCooling | Specifies whether the device provides active (as opposed to passive) cooling. | Boolean   |

# CIM\_Fan



The `CIM_Fan` class described in below Table contains a set of devices that work to keep the ambient internal temperature of the system at a safe value by circulating air.

The following table describes the properties and data type of the `CIM_Fan` class.

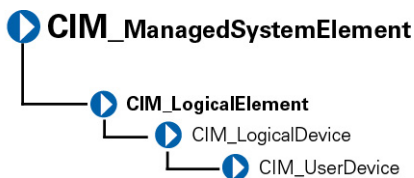
**Class Name:** `CIM_Fan`

**Parent Class:** `CIM_CoolingDevice`

**Table 29. CIM\_Fan Properties**

| Property      | Description  | Data Type |
|---------------|--|-----------|
| VariableSpeed | Specifies if the fan supports variable speeds.   | Boolean   |
| DesiredSpeed  | Indicates the currently requested fan speed, defined in RPM. When the VariableSpeed value is TRUE, the fan supports variable speeds. When a variable speed fan is supported (VariableSpeed is TRUE), the actual speed is determined using a sensor (CIM_Tachometer) that is associated with the fan. | uint64    |

# CIM\_UserDevice



The `CIM_UserDevice` class described in Table below contains logical devices that allow a system's users to input or view data. Classes derived from `CIM_UserDevice` include `CIM_Keyboard` and `CIM_PointingDevice`.

The following table describes the properties and data type of the `CIM User Device` class.

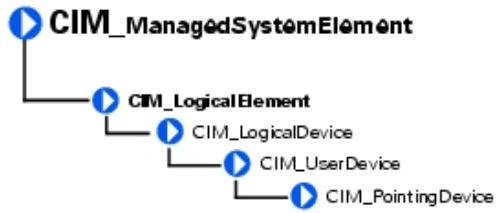
**Class Name:** `CIM_UserDevice`

**Parent Class:** `CIM_LogicalDevice`

**Table 30. CIM\_UserDevice Properties**

| Property | Description   | Data Type |
|----------|---|-----------|
| IsLocked | Indicates if the device is locked, preventing user input or output. | Boolean   |

# CIM\_PointingDevice



The CIM\_PointingDevice class described in Table below includes those devices used to point to regions of a display. Examples of such devices are a mouse or a trackball.

The following table describes the properties and data type of the CIM Pointing Device class.

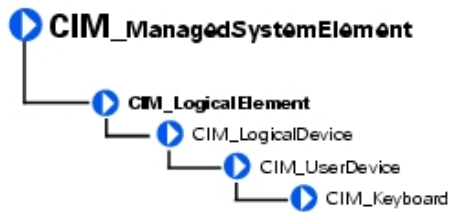
**Class Name:** CIM\_PointingDevice

**Parent Class:** CIM\_UserDevice

**Table 31. CIM\_PointingDevice Properties**

| Property        | Description   | Data Type |
|-----------------|---|-----------|
| PointingType    | Indicates the type of pointing device. Values for the <b>PointingType</b> property are:<br><b>1</b> — Other<br><b>2</b> — Unknown<br><b>3</b> — Mouse<br><b>4</b> — Trackball<br><b>5</b> — Trackpoint<br><b>6</b> — Glidepoint<br><b>7</b> — Touch pad<br><b>8</b> — Touch screen<br><b>9</b> — Mouse — optical sensor | uint16    |
| NumberOfButtons | Indicates the number of buttons. If the CIM_PointingDevice has no buttons, a value of 0 is returned.  | uint8     |
| Handedness      | Integer indicating if the CIM_PointingDevice is configured for right- or left-handed operation. Values for the <b>Handedness</b> property are as follows:<br><b>0</b> — Unknown<br><b>1</b> — Not applicable<br><b>2</b> — Right-handed operation<br><b>3</b> — Left-handed operation                                   | uint16    |

# CIM\_Keyboard



The CIM\_Keyboard class described in Table below includes devices that allow users to enter data.

The following table describes the properties and data type of the CIM Keyboard class.

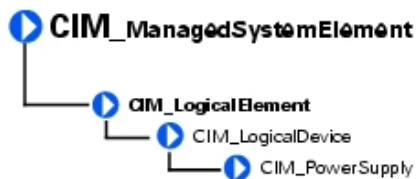
**Class Name:** CIM\_Keyboard

**Parent Class:** CIM\_UserDevice

**Table 32. CIM\_Keyboard Properties**

| Property             | Description   | Data Type |
|----------------------|---|-----------|
| NumberOfFunctionKeys | Indicates the number of function keys on the keyboard.  | uint16    |
| Layout               | A free-form string indicating the format and layout of the keyboard.  | string    |
| Password             | An integer indicating if a hardware-level password is enabled at the keyboard, preventing local input. Values for the <b>Password</b> property are:<br><br>1 — Other<br>2 — Unknown<br>3 — Disabled<br>4 — Enabled<br>5 — Not implemented | uint16    |

# CIM\_PowerSupply



The CIM\_PowerSupply class described in Table below contains devices that provide current and voltage for the operation of the system and its components

The following table describes the properties and data type of the CIM Power Supply CLASS.

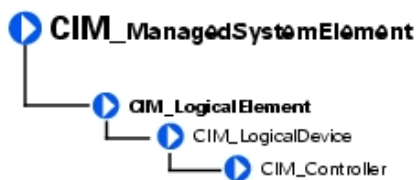
**Class Name:** CIM\_PowerSupply

**Parent Class:** CIM\_LogicalDevice

**Table 33. CIM\_PowerSupply Properties**

| Property               | Description   | Data Type |
|------------------------|---|-----------|
| IsSwitchingSupply      | Indicates that the power supply is a switching power supply and not a linear power supply.  | Boolean   |
| Range1InputVoltageLow  | Indicates the low voltage in millivolts of input voltage range 1 for this power supply. A value of 0 denotes unknown.   | uint32    |
| Range1InputVoltageHigh | Indicates the high voltage in millivolts of input voltage range 1 for this power supply. A value of 0 denotes unknown.  | uint32    |
| ActiveInputVoltage     | Indicates which input voltage range is currently in use. Range 1, range 2, or both can be specified using the values 3, 4, or 5, respectively. If the supply is not drawing power, a value of 6 (neither) can be specified. This information is necessary in the case of an uninterruptible power supply (UPS), a subclass of power supply. Values for the <b>ActiveInputVoltage</b> property are:<br><br><b>1</b> — Other<br><b>2</b> — Unknown<br><b>3</b> — Range 1<br><b>4</b> — Range 2<br><b>5</b> — Both range 1 and range 2<br><b>6</b> — Neither range 1 nor range 2 | uint16    |
| TotalOutputPower       | Defines the maximum sustained output wattage of the power supply (in tenths of Watts) 0 denotes 'unknown'.  | uint32    |
| InputRatedPower        | Defines the rated input wattage of the power supply (in tenths of Watts).   | uint32    |
| PMCapable              | Indicates the Power Monitoring capability.  | Boolean   |

## CIM\_Controller



The `CIM_Controller` class described in below Table properties, groups miscellaneous control-related devices. Examples of controllers are small computer system interface (SCSI) controllers, Universal Serial Bus (USB) controllers, and serial controllers.

The following table describes the properties and data type of the CIM Controller class.

**Class Name:** `CIM_Controller`

**Parent Class:** `CIM_LogicalDevice`

**Table 34. CIM\_Controller Properties**

| Property          | Description   | Data Type |
|-------------------|---|-----------|
| ProtocolSupported | The protocol used by the controller to access controlled devices. Values for the <b>ProtocolSupported</b> property are:<br><br><b>1</b> — Other<br><b>2</b> — Unknown<br><b>3</b> — PCI<br><b>4</b> — Parallel protocol | uint16    |

## CIM\_ParallelController



The CIM\_ParallelController class described in below contains a set of objects that control parallel devices. Parallel controllers transfer 8 or 16 bits of data at a time to the devices they control, for example, a parallel port controlling a printer.

The following table describes the properties and data type of the CIM Parallel Controller class.

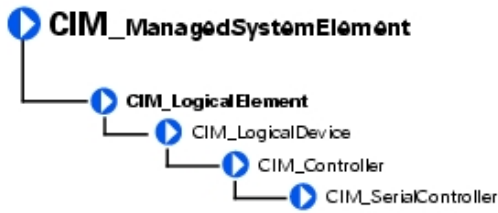
**Class Name:** CIM\_ParallelController

**Parent Class:** CIM\_Controller

**Table 35. CIM\_ParallelController Properties**

| Property   | Description   | Data Type |
|------------|---|-----------|
| DMASupport | Set to TRUE if the parallel controller supports DMA.  | Boolean   |
| Security   | An enumeration indicating the operational security for the controller. Values for the <b>Security</b> property are:<br><br><b>1</b> — Other<br><b>2</b> — Unknown<br><b>3</b> — None<br><b>4</b> — External interface locked out<br><b>5</b> — External interface enabled<br><b>6</b> — Boot bypass | uint16    |

# CIM\_SerialController



The `CIM_SerialController` class described in Table below contains controllers that transfer data one bit at a time to the devices they control, for example, a serial port controlling a modem.

The following table describes the properties and data type of the CIM Serial Controller class.

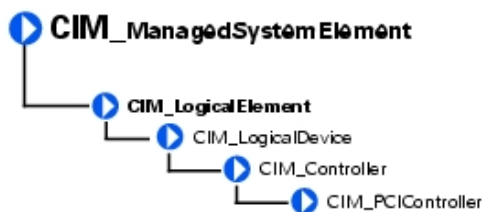
**Class Name:** `CIM_SerialController`

**Parent Class:** `CIM_Controller`

**Table 36. CIM\_SerialController Properties**

| Property    | Description   | Data Type |
|-------------|---|-----------|
| MaxBaudRate | Indicates the maximum baud rate in bits per second supported by the serial controller.  | uint32    |
| Security    | An enumeration indicating the operational security for the controller. Values for the <b>Security</b> property are:<br><br>1 — Other<br>2 — Unknown<br>3 — None<br>4 — External interface locked out<br>5 — External interface enabled<br>6 — Boot bypass | uint16    |

# CIM\_PCIController



The `CIM_PCIController` class described in Table below contains a set of devices that follow the Peripheral Component Interconnect (PCI) protocol defined by the Personal Computer Memory Card International Association (PCMCIA). The PCI protocol defines how data is transferred between devices. The `CIM_PCIController` class contains PCI adapters and bridges.

The following table describes the properties and data type of the CIM PCI Controller class.

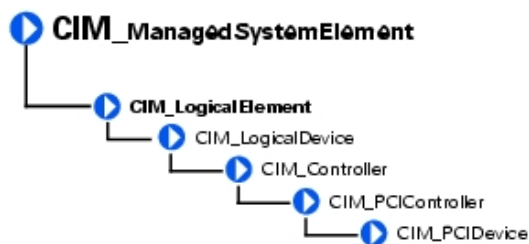
**Class Name:** `CIM_PCIController`

**Parent Class:** `CIM_Controller`

**Table 37. CIM\_PCIController Properties**

| Property        | Description   | Data Type |
|-----------------|---|-----------|
| CommandRegister | <p>The current contents of the register that provide basic control over the device’s ability to respond to, and/or perform PCI accesses. The data in the capabilities array is gathered from the PCI status register and the PCI capabilities list as defined in the PCI specification.</p> <p>Values for the <b>CommandRegister</b> property are:</p> <ul style="list-style-type: none"> <li><b>0</b> — Unknown</li> <li><b>1</b> — Other</li> <li><b>2</b> — Supports 66 MHz</li> <li><b>3</b> — Supports user-definable features</li> <li><b>4</b> — Supports fast back-to-back transactions</li> <li><b>5</b> — PCI-X capable</li> <li><b>6</b> — PCI power management supported</li> <li><b>7</b> — Message signaled interrupts supported</li> <li><b>8</b> — Parity error recovery capable</li> <li><b>9</b> — AGP supported</li> <li><b>10</b> — Vital product data supported</li> <li><b>11</b> — Provides slot identification</li> <li><b>12</b> — Hot swap supported</li> </ul> | uint16    |

## CIM\_PCIDevice



The CIM\_PCIDevice class shown in Table below describes the capabilities and management of a PCI device controller on an adapter card.

The following table describes the properties and data type of the CIM PCI Device class.

**Class Name:** CIM\_PCIDevice

**Parent Class:** CIM\_PCIController

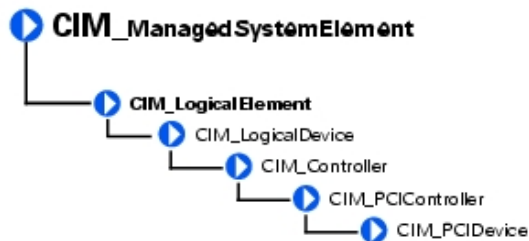
**Table 38. CIM\_PCIDevice Properties**

| Property    | Description   | Data Type |
|-------------|---|-----------|
| BaseAddress | Identifies an array of up to six double-word base memory addresses. | uint32    |

**Table 38. CIM\_PCIDevice Properties (continued)**

| Property                | Description  | Data Type |
|-------------------------|--|-----------|
| SubsystemID             | Identifies a subsystem identifier code.  | uint16    |
| SubsystemVendorID       | Identifies a subsystem vendor ID. ID information is reported from a PCI device via protocol-specific requests. This information is also present in the CIM_PhysicalElement class (the manufacturer property) for hardware, and the CIM_Product class (the vendor property) for information related to product acquisition. | uint16    |
| ExpansionROMBaseAddress | Identifies a double-word expansion ROM base memory address.  | uint32    |

## CIM\_PCIBridge



The CIM\_PCIBridge class described in [PCIBridge Properties](#) describes the capabilities and management of a PCI controller providing bridge-to-bridge capability. An example is a PCI to Industry-Standard Architecture (ISA) bus bridge.

The following table describes the properties and data type of the CIM PCI Bridge class.

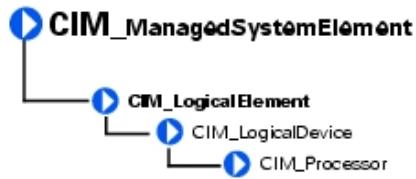
**Class Name:** CIM\_PCIBridge

**Parent Class:** CIM\_PCIController

**Table 39. CIM\_PCIBridge Properties**

| Property    | Description  | Data Type |
|-------------|--|-----------|
| BaseAddress | Identifies an array of double-word base memory addresses.  | uint32    |
| BridgeType  | Indicates the type of bridge. A bridge is PCI to <value>, except for the Host, which is a host-to-PCI bridge. Values for the <b>BridgeType</b> property are as follows:<br><br><b>0</b> — Host<br><b>1</b> — ISA<br><b>128</b> — Other | uint16    |
| BaseAddress | Identifies an array of double-word base memory addresses.  | uint32    |

# CIM\_Processor



The `CIM_Processor` class described in Table below contains devices that interpret and execute commands, for example, the Intel Xeon microprocessor.

The following table describes the properties and data type of the CIM Processor class.

**Class Name:** `CIM_Processor`

**Parent Class:** `CIM_LogicalDevice`

**Table 40. CIM\_Processor Properties**

| Property      | Description   | Data Type |
|---------------|---|-----------|
| Role          | A string describing the role of the microprocessor, for example, central microprocessor or math processor.  | string    |
| UpgradeMethod | Provides microprocessor socket information including data on how this microprocessor can be upgraded (if upgrades are supported). This property is an integer enumeration. Values for the UpgradeMethod property are as follows:<br><br><b>1</b> - Other<br><b>2</b> - Unknown<br><b>3</b> - Daughter board<br><b>4</b> - ZIF socket<br><b>5</b> - Replacement/piggy back<br><b>6</b> - None<br><b>7</b> - LIF socket<br><b>8</b> - Slot 1<br><b>9</b> - Slot 2<br><b>10</b> - 370-pin socket<br><b>19</b> - Socket mPGA604<br><b>20</b> - Socket LGA771<br><b>21</b> - Socket LGA775<br><b>22</b> - Socket S1<br><b>23</b> - Socket AM2<br><b>24</b> - Socket F (1207)<br><b>25</b> — Socket LGA1366 | uint16    |
| MaxClockSpeed | Indicates the maximum speed (in MHz) of this microprocessor.  | uint32    |
| Core count    | Indicates the number of core processors detected.   | uint16    |

**Table 40. CIM\_Processor Properties (continued)**

| Property                | Description   | Data Type |
|-------------------------|---|-----------|
| CoreEnabledCount        | Indicates the number of core processors enabled.  | uint16    |
| CurrentClockSpeed       | Indicates the current speed (in MHz) of this microprocessor.  | uint32    |
| DataWidth               | Indicates the processor data width in bits.   | uint16    |
| AddressWidth            | Indicates the processor address width in bits.  | uint16    |
| Stepping                | Indicates the revision level of the processor within the microprocessor family.   | string    |
| UniqueID                | Identifies a globally unique identifier for the microprocessor. This identifier may only be unique within a microprocessor family.  | string    |
| Brand                   | Indicates the brand name of the processor.  | string    |
| Model                   | Indicates the model name of the processor.  | string    |
| ExtendedCharacteristics | Indicates the extended capabilities of the processor. This attribute is a bit field. The following are the definitions of a bit when set to one:<br><br>Bit 0 — Virtualization Technology (VT) supported<br><br>Bit 1 — Demand-Based Switching (DBS) supported<br><br>Bit 2 — eXecute Disable (XD) supported<br><br>Bit 3 — Hyper Threading (HT) supported        | uint16    |
| ExtendedStates          | Indicates the setting of the extended capabilities of the processor. This attribute is a bit field. The following are the definitions of a bit when set to one:<br><br>Bit 0 — Virtualization Technology (VT) enabled<br><br>Bit 1 — Demand-Based Switching (DBS) enabled<br><br>Bit 2 — eXecute Disable (XD) enabled<br><br>Bit 3 — Hyper Threading (HT) enabled | uint16    |
| CPUStatus               | Indicates the current status of the microprocessor.<br><br>For example, it may be disabled by the user through the BIOS or disabled due to a POST error. Values for the CPUStatus property are as follows:<br><br><b>0</b> - Unknown<br><br><b>1</b> - Microprocessor enabled   | uint16    |

**Table 40. CIM\_Processor Properties (continued)**

| Property | Description   | Data Type |
|----------|---|-----------|
|          | <p><b>2</b> - Microprocessor disabled by user through BIOS setup</p> <p><b>3</b> - Microprocessor disabled by BIOS (POST error)</p> <p><b>4</b> - Microprocessor is idle</p> <p><b>5</b> - Other</p>  |           |
| Family   | <p>Refers to the processor family type. Values for the <b>Family</b> property are as follows:</p> <p><b>1</b> - Other</p> <p><b>2</b> - Unknown</p> <p><b>3</b> - 8086</p> <p><b>4</b> - 80286</p> <p><b>5</b> - 80386</p> <p><b>6</b> - 80486</p> <p><b>7</b> - 8087</p> <p><b>8</b> - 80287</p> <p><b>9</b> - 80387</p> <p><b>10</b> - 80487</p> <p><b>11</b> - Pentium Brand</p> <p><b>12</b> - Pentium Pro</p> <p><b>13</b> - Pentium II</p> <p><b>14</b> - Pentium processor with MMX technology</p> <p><b>15</b> - Celeron</p> <p><b>16</b> - Pentium II Xeon</p> <p><b>17</b> - Pentium III</p> <p><b>18</b> - M1 family</p> <p><b>19</b> - M2 family</p> <p><b>24</b> — AMD Duron processor</p> <p><b>25</b> — K5 family</p> <p><b>26</b> - K6 family</p> <p><b>27</b> - K6 -2</p> <p><b>28</b> - K6-3</p> <p><b>29</b> - AMD Athlon processor family</p> <p><b>30</b> - AMD29000 family</p> <p><b>31</b> - K6-2+</p> <p><b>32</b> - Power PC family</p> <p><b>33</b> - Power PC 601</p> <p><b>34</b> - Power PC 603</p> <p><b>35</b> - Power PC 603+</p> <p><b>36</b> - Power PC 604</p> | uint16    |

**Table 40. CIM\_Processor Properties (continued)**

| Property | Description   | Data Type |
|----------|---|-----------|
|          | <p><b>37</b> - Power PC 620</p> <p><b>38</b>- Power PC X704</p> <p><b>39</b>- Power PC 750</p> <p><b>40</b> - Intel Core Duo processor</p> <p><b>41</b> - Intel Core Duo mobile processor</p> <p><b>42</b> - Intel Core Solo mobile processor</p> <p><b>43</b> - Intel Atom processor</p> <p><b>48</b> - Alpha family</p> <p><b>49</b> - Alpha 21064</p> <p><b>50</b> - Alpha 21066</p> <p><b>51</b> - Alpha 21164</p> <p><b>52</b> - Alpha 21164</p> <p><b>53</b> - Alpha 21164a</p> <p><b>54</b> - Alpha 21264</p> <p><b>55</b> - Alpha 21364</p> <p><b>60</b> - AMD Opteron 4100 Series processor</p> <p><b>64</b> - MIPS family</p> <p><b>65</b> - MIPS R4000</p> <p><b>66</b> - MIPS R4200</p> <p><b>67</b> - MIPS R4400</p> <p><b>68</b> - MIPS R4600</p> <p><b>69</b> - MIPS R10000</p> <p><b>80</b> - SPARC family</p> <p><b>81</b> - SuperSPARC</p> <p><b>82</b> - microSPARC II</p> <p><b>83</b> - microSPARC IIep</p> <p><b>84</b> - UltraSPARC</p> <p><b>85</b> - UltraSPARC II</p> <p><b>86</b> - UltraSPARC III</p> <p><b>87</b> - UltraSPARC III</p> <p><b>88</b> - UltraSPARC IIIi</p> <p><b>96</b> - 68040</p> <p><b>97</b> - 68xxx family</p> <p><b>98</b> - 68000</p> <p><b>99</b> - 68010</p> <p><b>100</b> - 68020</p> <p><b>101</b> - 68030</p> <p><b>112</b> - Hobbit family</p> <p><b>120</b> - Crusoe 5000 family</p> <p><b>121</b> - Crusoe 3000 family</p> |           |

**Table 40. CIM\_Processor Properties (continued)**

| Property | Description   | Data Type |
|----------|---|-----------|
|          | <p><b>122</b> - Efficeon 8000 family</p> <p><b>128</b> - Weitek</p> <p><b>130</b> - Itanium processor</p> <p><b>131</b> - AMD Athlon 64 processor family</p> <p><b>132</b> - AMD Opteron processor family</p> <p><b>133</b> - AMD Sempron processor family</p> <p><b>134</b> - AMD Turion 64 Mobile technology</p> <p><b>135</b> - Dual-Core AMD Opteron processor family</p> <p><b>136</b> - AMD Athlon 64 X2 Dual-Core processor family</p> <p><b>137</b> - AMD Turion 64 X2 Mobile technology</p> <p><b>138</b> - Quad-Core AMD Opteron processor family</p> <p><b>139</b> - Third-Generation AMD Opteron processor family</p> <p><b>140</b> - AMD Phenom FX Quad-Core processor family</p> <p><b>141</b> - AMD Phenom X4 Quad-Core processor family</p> <p><b>142</b> - AMD Phenom X2 Dual-Core processor family</p> <p><b>143</b> - AMD Athlon X2 Dual-Core processor family</p> <p><b>144</b> - PA-RISC family</p> <p><b>145</b> - PA-RISC 8500</p> <p><b>146</b> - PA-RISC 8000</p> <p><b>147</b> - PA-RISC 7300LC</p> <p><b>148</b> - PA-RISC 7200</p> <p><b>149</b> - PA-RISC 7100LC</p> <p><b>150</b> - PA-RISC 7100</p> <p><b>160</b> - V30 family</p> <p><b>161</b> - Quad-Core Intel Xeon processor 3200 Series</p> <p><b>162</b> - Dual-Core Intel Xeon processor 3000 Series</p> <p><b>163</b> - Quad-Core Intel Xeon processor 5300 Series</p> <p><b>164</b> - Dual-Core Intel Xeon processor 5100 Series</p> <p><b>165</b> - Dual-Core Intel Xeon processor 5000 Series</p> <p><b>166</b> - Dual-Core Intel Xeon processor LV</p> <p><b>167</b> - Dual-Core Intel Xeon processor ULV</p> |           |

**Table 40. CIM\_Processor Properties (continued)**

| Property | Description  | Data Type |
|----------|--|-----------|
|          | <p><b>168</b> - Dual-Core Intel Xeon processor 7100 Series</p> <p><b>169</b> - Quad-Core Intel Xeon processor 5400 Series</p> <p><b>170</b> - Quad-Core Intel Xeon processor</p> <p><b>171</b>- Dual-Core Intel Xeon processor 5200 Series</p> <p><b>172</b>- Dual-Core Intel Xeon processor 7200 Series</p> <p><b>173</b> - Quad-Core Intel Xeon processor 7300 Series</p> <p><b>174</b>- Quad-Core Intel Xeon processor 7400 Series</p> <p><b>175</b>- Multi-Core Intel Xeon processor 7400 Series</p> <p><b>176</b> - Pentium III Xeon</p> <p><b>177</b> - Pentium III Processor with Intel SpeedStep</p> <p><b>178</b>- Technology</p> <p><b>179</b> - Pentium 4</p> <p><b>180</b> - Intel Xeon</p> <p><b>181</b>- AS400 family</p> <p><b>182</b> - Intel Xeon Processor MP</p> <p><b>183</b> - AMD Athlon XP family</p> <p><b>184</b> - AMD Athlon MP family</p> <p><b>185</b> - Intel Itanium 2</p> <p><b>186</b>- Intel Pentium M processor</p> <p><b>187</b> - Intel Celeron D processor</p> <p><b>188</b> - Intel Pentium D processor</p> <p><b>189</b> - Intel Pentium Extreme Edition processor</p> <p><b>190</b>- Intel Core 2 processor</p> <p><b>192</b> - Intel Core 2 Solo processor</p> <p><b>193</b> - Intel Core 2 Extreme processor</p> <p><b>194</b>- Intel Core 2 Quad processor</p> <p><b>195</b> - Intel Core 2 Extreme mobile processor</p> <p><b>196</b>- Intel Core 2 Duo mobile processor</p> <p><b>197</b>- Intel Core 2 Solo mobile processor</p> <p><b>198</b> - Intel Core i7 processor</p> <p><b>199</b> - Dual-Core Intel Celeron processor</p> <p><b>200</b> - S/390 and zSeries family</p> <p><b>201</b>- ESA/390 G4</p> <p><b>202</b>- ESA/390 G5</p> |           |

**Table 40. CIM\_Processor Properties (continued)**

| Property | Description  | Data Type |
|----------|--|-----------|
|          | <p><b>203</b>- ESA/390 G6</p> <p><b>204</b> - z/Architecture base</p> <p><b>206</b> - CEh 206 Intel Core i3 processor</p> <p><b>214</b> - Multi-Core Intel Xeon processor</p> <p><b>215</b> - Dual-Core Intel Xeon processor 3xxx Series</p> <p><b>216</b> - Quad-Core Intel Xeon processor 3xxx Series</p> <p><b>217</b> - D9h 217 VIA Nano processor family</p> <p><b>218</b> - Dual-Core Intel Xeon processor 5xxx Series</p> <p><b>219</b> - Quad-Core Intel Xeon processor 5xxx Series</p> <p><b>221</b> - Dual-Core Intel Xeon processor 7xxx Series</p> <p><b>222</b>- Dual-Core Intel Xeon processor 7xxx Series</p> <p><b>223</b> Multi-Core Intel Xeon processor 7xxx Series</p> <p><b>224</b> - E0h 224 Multi-Core Intel Xeon processor 3400 Series</p> <p><b>230</b> - Embedded AMD Opteron Quad-Core processor family</p> <p><b>231</b> - AMD Phenom Triple-Core processor family</p> <p><b>232</b> - AMD Turion Ultra Dual-Core Mobile processor family</p> <p><b>233</b> - AMD Turion Dual-Core Mobile processor family</p> <p><b>234</b> - AMD Turion Dual-Core Mobile processor family</p> <p><b>235</b>- AMD Sempron SI processor family</p> <p><b>238</b> - AMD Opteron Six-Core processor family</p> <p><b>250</b> - i860</p> <p><b>251</b> - i960</p> <p><b>260</b> - SH-3</p> <p><b>261</b> - SH-4</p> <p><b>280</b> - ARM</p> <p><b>281</b> - StrongARM</p> <p><b>300</b> - 6x86</p> <p><b>301</b> - MediaGX</p> <p><b>302</b> - MII</p> <p><b>320</b> - WinChip</p> <p><b>350</b>- DSP</p> |           |

**Table 40. CIM\_Processor Properties (continued)**

| Property | Description           | Data Type |
|----------|-----------------------|-----------|
|          | 500 - Video processor |           |

## CIM\_StorageExtent

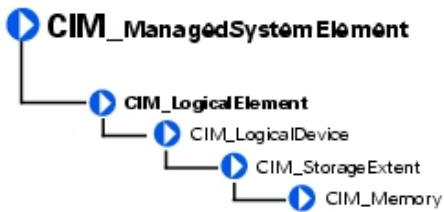


The CIM\_StorageExtent identified in Table below contains devices that manage data storage, for example, hard drives or microprocessor memory.

**Table 41. CIM\_StorageExtent Properties**

| Properties    |                   |
|---------------|-------------------|
| Class Name:   | CIM_StorageExtent |
| Parent Class: | CIM_LogicalDevice |

## CIM\_Memory

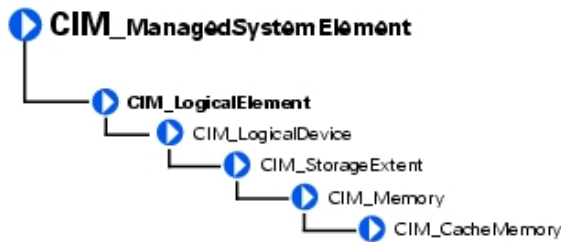


The CIM\_Memory class identified in Table below describes the capabilities and management of storage extent devices, for example, cache memory or system memory.

**Table 42. CIM\_Memory Properties**

| Properties    |                   |
|---------------|-------------------|
| Class Name:   | CIM_Memory        |
| Parent Class: | CIM_StorageExtent |

# CIM\_CacheMemory



The CIM\_CacheMemory class described in [CacheMemoryProperties](#) describes the capabilities and management of cache memory. Cache memory allows a microprocessor to access data and instructions faster than normal system memory

The following table describes the property, description, and data type details for CIM CacheMemory.

**Class Name:** CIM\_CacheMemory

**Parent Class:** CIM\_Memory

**Table 43. CIM\_CacheMemory Properties**

| Property    | Description   | Data Type |
|-------------|---|-----------|
| Level       | Defines if this is the primary, secondary, or tertiary cache. Values for the <b>Level</b> property are as follows: <ol style="list-style-type: none"> <li>1- Other</li> <li>2- Unknown</li> <li>3- Primary</li> <li>4 - Secondary</li> <li>5- Tertiary</li> <li>6- Not applicable</li> </ol>  | uint16    |
| WritePolicy | Defines if this cache is a write-back or write-through cache or if this information varies with address or is defined individually for each input/output (I/O). Values for the <b>WritePolicy</b> property are as follows: <ol style="list-style-type: none"> <li>1- Other</li> <li>2- Unknown</li> <li>3- Write-back</li> <li>4 - Write-through</li> <li>5- Varies with address</li> <li>6- Determination per I/O</li> </ol> | uint16    |
| CacheType   | Defines if this cache is for instruction caching, data caching, or both (unified). Values for the <b>CacheType</b> property are as follows: <ol style="list-style-type: none"> <li>1- Other</li> <li>2- Unknown</li> <li>3- Instruction</li> <li>4- Data</li> </ol>   | uint16    |

**Table 43. CIM\_CacheMemory Properties (continued)**

| Property   | Description  | Data Type |
|------------|--|-----------|
|            | 5- Unified   |           |
| LineSize   | Indicates the size, in bytes, of a single cache bucket or line.  | uint32    |
| ReadPolicy | Defines the policy used by the cache for handling read requests. Values for the <b>ReadPolicy</b> property are as follows:<br>1- Other<br>2- Unknown<br>3- Read<br>4- Read-ahead<br>5- Read and read-ahead<br>6- Determination per I/O | uint16    |

## DELL\_SoftwareFeature



The DELL\_SoftwareFeature described in Table below defines the universal resource locator (URL) of the systems management software and the language in which systems management information displays. Defining these properties enables users to manage a system using an Internet browser. You can access Server Administrator using the secure hypertext transfer protocol (https) and a preassigned port number of 1311, or you can specify a port number of your own choice.

The following table describes the property, description, and data type details for DELL SoftwareFeature.

**Class Name:** DELL\_SoftwareFeature

**Parent Class:** CIM\_SoftwareFeature

**Table 44. DELL\_SoftwareFeature Properties**

| Property     | Description  | Data Type |
|--------------|--|-----------|
| OmsaURL      | Defines the URL for Server Administrator.                                  | string    |
| Language     | Sets the language for systems management information.                      | string    |
| AgentVersion | Defines the version information of local CIM agent (same as ISVC version.) | string    |

# CIM\_BIOSElement



The `CIM_BIOSElement` class listed in [BIOSElement Properties](#) describes the BIOS for the system. The BIOS controls the following:

- Communications between the microprocessor and peripheral devices, such as the keyboard and the video adapter.
- Miscellaneous functions, such as system messages.

The following table describes the property, description, and data type details for `CIM_BIOSElement`.

**Class Name:** `CIM_BIOSElement`

**Parent Class:** `CIM_SoftwareElement`

**Table 45. CIM\_BIOSElement Properties**

| Property     | Description   | Data Type |
|--------------|---|-----------|
| Version      | Provides the product version information.   | string    |
| Manufacturer | See <a href="#">Common Properties of Classes</a>  | string    |
| PrimaryBIOS  | Specifies whether a given BIOS is the primary BIOS for the system. When the value = TRUE, the BIOS is the primary BIOS. | Boolean   |

# CIM\_SoftwareFeature



The `CIM_SoftwareFeature` class described in Table properties below defines a particular function or capability of a product or application system. This class is intended to be meaningful to a consumer, or user of a product, rather than to explain how the product is built or packaged. When a software feature can exist on multiple platforms or operating systems (for example, a client component of a three-tiered client/server application might run on Windows Server 2003), a software feature is a collection of all the software elements for these different platforms. The users of the model must be aware of this situation because typically they are interested in a sub-collection of the software elements required for a particular platform.

The following table describes the property, description, and data type details for `CIM SoftwareFeature`.

**Class Name:** `CIM_SoftwareFeature`

**Parent Class:** `CIM_LogicalElement`

**Table 46. CIM\_SoftwareFeature Properties**

| Property          | Description   | Data Type |
|-------------------|---|-----------|
| IdentifyingNumber | Provides product identification such as a serial number of the software | string    |

**Table 46. CIM\_SoftwareFeature Properties (continued)**

| Property    | Description   | Data Type |
|-------------|---|-----------|
| ProductName | Identifies the commonly used product name.  | string    |
| Vendor      | Identifies the name of the product's supplier. Corresponds to the vendor property in the product object in the DMTF solution exchange standard. | string    |
| Version     | Identifies the product version information. Corresponds to the version property in the product object in the DMTF solution exchange standard.   | string    |
| Name        | Defines the label by which the object is known to the users. This label is a user-defined name that uniquely identifies the element.            | string    |

## DELL\_SoftwareFeature



The `DELL_SoftwareFeature` described in Table below defines the universal resource locator (URL) of the systems management software and the language in which systems management information displays. Defining these properties enables users to manage a system using an Internet browser. You can access Server Administrator using the secure hypertext transfer protocol (https) and a preassigned port number of 1311, or you can specify a port number of your own choice.

The following table describes the property, description, and data type details for DELL SoftwareFeature.

**Class Name:** `DELL_SoftwareFeature`

**Parent Class:** `CIM_SoftwareFeature`

**Table 47. DELL\_SoftwareFeature Properties**

| Property     | Description  | Data Type |
|--------------|--|-----------|
| OmsaURL      | Defines the URL for Server Administrator.                                  | string    |
| Language     | Sets the language for systems management information.                      | string    |
| AgentVersion | Defines the version information of local CIM agent (same as ISVC version.) | string    |

## CIM\_SystemResource



The CIM\_SystemResource class described in Table below provides access to system resources from an operating system. System resources consist of interrupt requests (IRQs) and direct memory access (DMA) capabilities.

**Table 48. CIM\_SystemResource Properties**

| Properties           |                    |
|----------------------|--------------------|
| <b>Class Name:</b>   | CIM_SystemResource |
| <b>Parent Class:</b> | CIM_LogicalElement |

## CIM\_IRQ



The CIM\_IRQ class described in Properties Table below , contains IRQ information. An IRQ is a signal that data is about to be sent to or received by a peripheral device. The signal travels by an IRQ line to the microprocessor. Each peripheral connection must be assigned an IRQ number. For example, the first serial port in the computer (COM1) is assigned to IRQ4 by default.

The following table describes the property, description, and data type details for CIM IRQ.

**Class Name:** CIM\_IRQ

**Parent Class:** CIM\_SystemResource

**Table 49. CIM\_IRQ Properties**

| Property            | Description  | Data Type |
|---------------------|--|-----------|
| CSCreationClassName | See <a href="#">Common Properties of Classes</a>   | string    |
| CSName              | See <a href="#">Common Properties of Classes</a>   | string    |
| CreationClassName   | See <a href="#">Common Properties of Classes</a>   | string    |
| IRQNumber           | Identifies the interrupt request number.   | uint32    |
| Availability        | Indicates the availability of the IRQ.<br>Values for the Availability property are as follows:<br><br><ol style="list-style-type: none"> <li>1- Other</li> <li>2- Unknown</li> <li>3- Available</li> <li>4 - In use/not available</li> <li>5- In use and available</li> </ol>  | uint16    |
| TriggerLevel        | Indicates if the interrupt is triggered by the hardware signal going high or low.<br>Values for the TriggerLevel property are as follows:<br><br><ol style="list-style-type: none"> <li>1- Other</li> <li>2- Unknown</li> <li>3- Active low</li> <li>4- Active high</li> </ol> | uint16    |
| TriggerType         | Indicates if edge (value=4) or level triggered (value=3) interrupts occur.   | uint16    |

**Table 49. CIM\_IRQ Properties (continued)**

| Property  | Description   | Data Type |
|-----------|---|-----------|
|           | <b>1-</b> Other<br><b>2-</b> Unknown<br><b>3-</b> Level<br><b>4-</b> Edge   |           |
| Shareable | Indicates if the IRQ can be shared. A value of TRUE indicates that the IRQ can be shared.   | Boolean   |
| Hardware  | Indicates if the interrupt is hardware- or software-based. (A value of TRUE indicates that the interrupt is hardware based.) On a personal computer, a hardware IRQ is a physical wire to a programmable interrupt controller (PIC) chip set through which the microprocessor can be notified of time critical events. Some IRQ lines are reserved for standard devices such as the keyboard, diskette drive, and the system clock. A software interrupt is a programmatic mechanism to allow an application to get the attention of the processor. | Boolean   |

## CIM\_MemoryMappedIO



The CIM\_MemoryMappedIO class described in properties Table below addresses both memory and port I/O resources for personal computer architecture memory mapped I/O.

The following table describes the property, description, and data type details for the CIM MemoryMappedIO.

**Class Name:** CIM\_MemoryMappedIO

**Parent Class:** CIM\_SystemResource

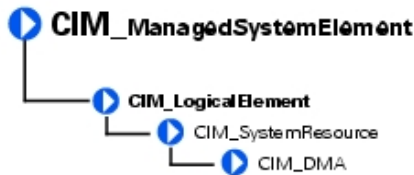
**Table 50. CIM\_MemoryMappedIO Properties**

| Property            | Description   | Data Type |
|---------------------|---|-----------|
| CSCreationClassName | See <a href="#">Common Properties of Classes</a>      | string    |
| CSName              | See <a href="#">Common Properties of Classes</a>      | string    |
| CreationClassName   | See <a href="#">Common Properties of Classes</a>      | string    |
| StartingAddress     | Identifies the starting address of memory mapped I/O. | uint64    |
| EndingAddress       | Identifies the ending address of memory mapped I/O.   | uint64    |

**Table 50. CIM\_MemoryMappedIO Properties (continued)**

| Property       | Description   | Data Type |
|----------------|---|-----------|
| MappedResource | Indicates the type of memory mapped I/O. MappedResource defines if memory or I/O is mapped, and for I/O, if the mapping is to a memory or a port space. Memory mapped I/O values are as follows:<br><br><ol style="list-style-type: none"> <li>1- Other</li> <li>2- Mapped memory</li> <li>3 - I/O mapped to memory space</li> <li>4- I/O mapped to port space</li> </ol> | uint16    |

## CIM\_DMA



The CIM\_DMA class described in [DMA Properties](#) contains DMA information. A DMA channel allows certain types of data transfer between RAM and a device to bypass the microprocessor.

The following table describes the property, description, and data type details for CIM DMA.

**Class Name:** CIM\_DMA

**Parent Class:** CIM\_SystemResource

**Table 51. CIM\_DMA Properties**

| Property            | Description  | Data Type |
|---------------------|--|-----------|
| CSCreationClassName | See <a href="#">Common Properties of Classes</a>   | string    |
| CSName.             | See <a href="#">Common Properties of Classes</a>   | string    |
| CreationClassName   | See <a href="#">Common Properties of Classes</a>   | string    |
| DMAChannel          | A part of the object’s key value, the DMA channel number.  | uint32    |
| Availability        | Availability of the DMA. <b>Availability</b> values are defined as follows:<br><br><ul style="list-style-type: none"> <li>• 1 - Other</li> <li>• 2 - Unknown</li> <li>• 3 - Available</li> <li>• 4 - In Use/Not Available</li> <li>• 5 - In Use and Available/Shareable</li> </ul> | uint16    |

# CIM\_RedundancyGroup



The CIM\_RedundancyGroup class described in Table properties below is a set of components that provide more instances of a critical component than are required for the system’s operation. The extra components are used in case of critical component failure. For example, multiple power supplies allow a working power supply to take over when another power supply has failed.

The following table describes the property, description, and data type details for the CIM RedundancyGroup.

**Class Name:** CIM\_RedundancyGroup

**Parent Class:** CIM\_LogicalElement

**Table 52. CIM\_RedundancyGroup Properties**

| Property          | Description  | Data Type |
|-------------------|--|-----------|
| CreationClassName | See <a href="#">Common Properties of Classes</a>   | string    |
| Name              | Serves as the key for the redundancy group’s instance in an enterprise environment.  | string    |
| RedundancyStatus  | Provides information on the state of the redundancy group. Values for the <b>RedundancyStatus</b> property are as follows:<br><br><b>0</b> - Unknown<br><b>1</b> - Other<br><b>2</b> - Fully redundant. Fully redundant - all of the configured redundancy is still available.<br><b>3</b> - Degraded redundancy. Degraded redundancy - that some failures have been experienced but some reduced amount of redundancy is still available.<br><b>4</b> - Redundancy lost. Redundancy lost - that a sufficient number of failures have occurred so that no redundancy is available and the next failure experienced causes overall failure. | uint16    |

# CIM\_ExtraCapacityGroup



The CIM\_ExtraCapacityGroup class described in below properties Table applies to systems that have more capability and components than are required for normal operation, for example, systems that have extra fans or power supplies.

The following table describes the property, description, and data type details for the CIM ExtraCapacityGroup.

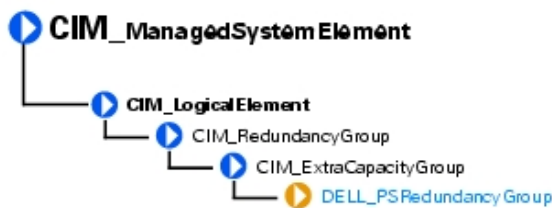
**Class Name:** CIM\_ExtraCapacityGroup

**Parent Class:** CIM\_RedundancyGroup

**Table 53. CIM\_ExtraCapacityGroup Properties**

| Property        | Description  | Data Type |
|-----------------|--|-----------|
| MinNumberNeeded | Specifies the smallest number of elements that must be operational in order to have redundancy. For example, in an N+1 redundancy relationship, the MinNumberNeeded property should be set to N. | uint32    |

## DELL\_PSRedundancyGroup

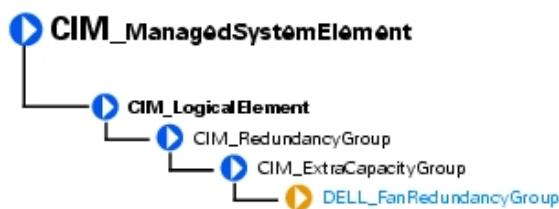


The DELL\_PSRedundancyGroup described in Table below is a Dell-specific extension of the CIM\_ExtraCapacityGroup class. The DELL\_PSRedundancyGroup class defines what constitutes power supply redundancy in a system.

**Table 54. DELL\_PSRedundancyGroup Properties**

| Properties           |                        |
|----------------------|------------------------|
| <b>Class Name:</b>   | DELL_PSRedundancyGroup |
| <b>Parent Class:</b> | CIM_ExtraCapacityGroup |

## DELL\_FanRedundancyGroup



The DELL\_FanRedundancyGroup described in Table below defines what constitutes fan redundancy in a system.

**Table 55. DELL\_FanRedundancyGroup Properties**

| Properties           |                         |
|----------------------|-------------------------|
| <b>Class Name:</b>   | DELL_FanRedundancyGroup |
| <b>Parent Class:</b> | CIM_ExtraCapacityGroup  |

# CIM\_EnabledLogicalElement



The `CIM_EnabledLogicalElement` class described in Table below extends the `CIM_LogicalElement` class to abstract the concept of an element that is enabled or disabled, such as a `LogicalDevice` or `ServiceAccessPoint`.

**Table 56. CIM\_EnabledLogicalElement Properties**

| Properties           |   |
|----------------------|---|
| <b>Class Name:</b>   | <code>CIM_EnabledLogicalElementGroup</code> |
| <b>Parent Class:</b> | <code>CIM_LogicalElementGroup</code>        |

# CIM\_ServiceAccessPoint

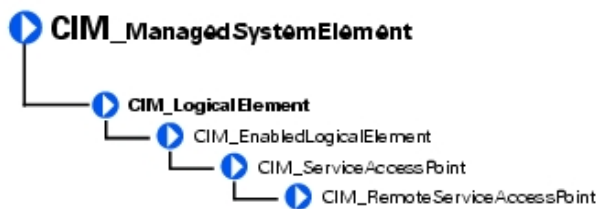


The `CIM_ServiceAccessPoint` class described in Table below represents the ability to utilize or invoke a service. Access points indicate that a service is available to other entities for use.

**Table 57. CIM\_ServiceAccessPoint Properties**

| Properties           |  |
|----------------------|--|
| <b>Class Name:</b>   | <code>CIM_ServiceAccessPoint</code>    |
| <b>Parent Class:</b> | <code>CIM_EnabledLogicalElement</code> |

# CIM\_RemoteServiceAccessPoint



The `CIM_RemoteServiceAccessPoint` class identified in Table below describes the accessing and addressing of information for a remote connection that is known to a *local* network element. This information is contained in the *local* network element since this is the context in which it is *remote*. The relevance of the remote service access point and information on its use are described by subclassing or associating to the `CIM_RemoteServiceAccessPoint` class.

The following table describes the property, description, and data type details for the `CIM RemoteServiceAccessPoint`.

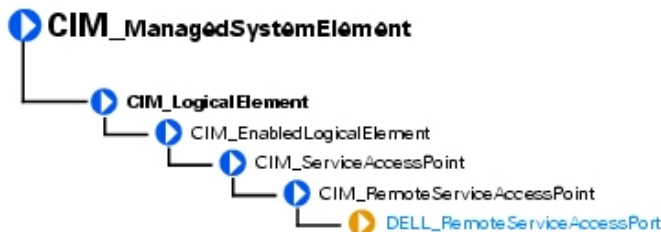
**Class Name:** `CIM_RemoteServiceAccessPointGroup`

Parent Class: CIM\_ServiceAccessPointGroup

**Table 58. CIM\_RemoteServiceAccessPoint Properties**

| Property   | Description   | Data Type |
|------------|---|-----------|
| AccessInfo | Describes accessing or addressing of information for a remote connection. This can be a host name, network address, and other similar information.  | string    |
| InfoFormat | Indicates an enumerated integer describing the format and interpretation of the AccessInfo property. This property can have the following values:<br><b>1</b> - Other<br><b>2</b> - Host Name<br><b>3</b> - IPv4 Address<br><b>4</b> - IPv6 Address<br><b>5</b> - IPX Address<br><b>6</b> - DECnet Address<br><b>7</b> - SNA Address<br><b>8</b> - Autonomous System Number<br><b>9</b> - MPLS Label<br><b>10..99</b> - DMTF Reserved<br><b>100</b> - Dial String<br><b>101</b> - Ethernet Address<br><b>102</b> - Token Ring Address<br><b>103</b> - ATM Address<br><b>104</b> - Frame Relay Address<br><b>105..199</b> - DMTF Reserved<br><b>200</b> - URL<br><b>32768..65535</b> - Vendor Specific | uint16    |

## DELL\_RemoteServiceAccessPort



The DELL\_RemoteServiceAccessPort class described in Table below is an extended class of the CIM\_RemoteServiceAccessPoint class. The DELL\_RemoteServiceAccessPort class provides information about Dell implementation-specific attributes.

The following table describes the property, description, and data type details for the DELL RemoteServiceAccessPort.

**Class Name:** DELL\_RemoteServiceAccessPort

Parent Class: CIM\_RemoteServiceAccessPoint

**Table 59. DELL\_RemoteServiceAccessPort Properties**

| Property         | Description   | Data Type |
|------------------|---|-----------|
| PortName         | Displays the name of the service access port.   | string    |
| VersionString    | Indicates the version of the access point service.  | string    |
| RemoteAccessType | Indicated the type of remote access service. This property can have the following values:<br><b>0</b> - BMC<br><b>8</b> - IMC<br><b>9</b> - CMC<br><b>10</b> - iDRAC6<br><b>11</b> - iDRAC6 for modular systems<br><b>13</b> - BMC<br><b>16</b> - iDRAC7<br><b>17</b> - iDRAC7 for modular systems  | uint16    |
| BladeFormFactor  | Type of Blade Form Factor. This property can have the following values:<br><b>0</b> - singleWidthHalfHeight<br><b>1</b> - dualWidthHalfHeight<br><b>2</b> - singleWidthFullHeight<br><b>3</b> - dualWidthFullHeight<br><b>4</b> - singleWidthQuarterHeight<br><b>5</b> - 1UHalfWidth<br><b>6</b> - 1UQuarterWidth<br><b>7</b> - 1UFullWidth<br><b>255</b> - notApplicable | uint16    |

## Dell-Defined Classes

Dell-defined classes are defined and populated by Dell rather than by the Common Information Model (CIM). For information on how the logs are formatted, see [DELL\\_Chassis Properties](#).

### ▶ CIM\_ManagedSystemElement

#### ▶ DELL\_EsmLog

The `DELL_EsmLog` class described in [DELL\\_EsmLog Properties](#) records failure threshold violations collected by Server Administrator's embedded server management (ESM) capabilities.

The following table describes the property, description, and data type details for the Dell Defined Classes.

**Class Name:** `DELL_EsmLog`

**Parent Class:** None

**Table 60. DELL\_EsmLog Properties**

| Property     | Description  | Data Type |
|--------------|--|-----------|
| recordNumber | Provides an index to the ESM table.                                      | uint32    |
| logRecord    | Provides the ESM message content.  | string    |
| eventTime    | Indicates the time that the message is generated.                        | datetime  |
| status       | Indicates the severity of the event that caused the log to be generated. | string    |

### Topics:

- [DELL\\_PostLog](#)
- [DELL\\_BIOSExtensions](#)
- [DELL\\_BIOSSettings](#)
- [DELL\\_SDCardDevice](#)
- [DELL\\_NetworkPort](#)
- [DELL\\_PowerConsumptionAmpsSensor](#)
- [DELL\\_PowerConsumptionWattsSensor](#)
- [DELL\\_PowerConsumptionData](#)
- [DCIM\\_OEM\\_DataAccessModule](#)
- [DCIM\\_RegisteredProfile](#)

## DELL\_PostLog

### ▶ CIM\_ManagedSystemElement

#### ▶ DELL\_PostLog

The `DELL_PostLog` class described in [DELL\\_PostLog Properties](#) is a record of the system's power-on self-test (POST). When you turn on a system, the POST tests various system components, such as random-access memory (RAM), the hard drives, and the keyboard.

**Table 61. DELL\_PostLog Properties**

| Properties           |              |
|----------------------|--------------|
| <b>Class Name:</b>   | DELL_PostLog |
| <b>Parent Class:</b> | None         |

## DELL\_BIOSExtensions

The DELL\_BIOSExtensions identified in [DELL\\_BIOSExtensions Properties](#) contains information related to the specific extension of the data attributes on your system.

The following table describes the property, description, and data type details for DELL BIOSExtensions.

**Class Name:** DELL\_BIOSExtensions

**Parent Class:** CIM\_ManagedSystemElement

**Table 62. DELL\_BIOSExtensions Properties**

| Property                      | Description   | Data Type |
|-------------------------------|---|-----------|
| systemBIOSCharacteristics     | Indicates the characteristics of BIOS on your system.                   | uint64    |
| systemBIOSCharacteristicsExt1 | Indicates the specific extension of the data attributes on your system. | uint8     |
| systemBIOSCharacteristicsExt2 | Indicates the specific extension of the data attributes on your system. | uint8     |

## DELL\_BIOSSettings

The DELL\_BIOSSettings identified in [DELL\\_BIOSSettings Properties](#) contains information related to setting parameters in the Dell System Management BIOS.

The following table describes the property, description, and data type details for DELL BIOSSettings.

**Class Name:** DELL\_BIOSSettings

**Parent Class:** CIM\_ManagedSystemElement

**Table 63. DELL\_BIOSSettings Properties**

| Property              | Description  | Data Type |
|-----------------------|--|-----------|
| DellInstanceID        | Defines the instance ID of this class.   | uint32    |
| TrustedPlatformModule | Enables or Disables the Trusted Platform Module (TPM). Values for the TPM property are:<br>0 - Other<br>1 - Unsupported<br>2 - Off<br>3 - On with BIOS Management<br>4 - On without BIOS Measurement | uint8     |

# DELL\_SDCardDevice

The DELL\_SDCard Devices identified in [DELL\\_SDCardDevice Properties](#) contains information related to the SD card devices. The following table describes the property, description, and data type details for DELL SDCardDevice.

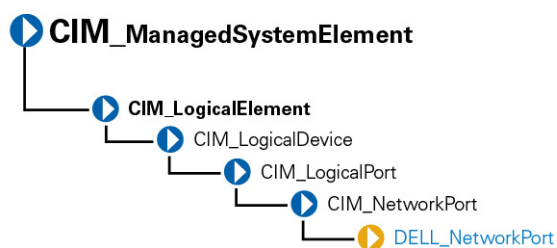
**Class Name:** DELL\_SDCardDevice

**Parent Class:** CIM\_LogicalDevice

**Table 64. DELL SDCardDevice Properties**

| Property         | Description  | Data Type |
|------------------|--|-----------|
| sdType           | An enumerated storage device type. The values for this property are:<br>1 - Other<br>2 - Unknown<br>3 - Hypervisor SD<br>4 - Virtual Flash SD  | uint8     |
| sdCertified      | Indicates the licensing information of SD media. The values for this property are:<br>0 - Unknown<br>1 - Unlicensed<br>2 - Licensed  | uint8     |
| sdCardSizeMB     | Indicates the size of the storage device in MB.  | uint32    |
| sdCardFreeSizeMB | Indicates the available size of SD Media in MB.  | uint32    |
| sdCardState      | Indicates the value of the SD Card. The values for this property are:<br>0 - Present<br>1 and 2 - Reserved<br>3 - Offline Detected<br>4 - Failed Detectez<br>5 - Active<br>6 - Bootable<br>7 - Write Protected |           |

# DELL\_NetworkPort



The `Dell_Network_Port` class described in [DELL\\_NetworkPort Properties](#) represents the Dell-specific features of the network adapters.

The following table describes the property, description, and data type details for DELL NetworkPort Properties.

**Class Name:** `DELL_NetworkPort`

**Parent Class:** `CIM_Network_Port`

**Table 65. DELL NetworkPort Properties**

| Property            | Description  | Data Type |
|---------------------|--|-----------|
| NIC Capabilities    | <p>NIC Capabilities bitmask indicates the capabilities of the NIC.</p> <p>The bitmask for the NIC Capability property are:</p> <p>Bit 0, Value 0 - Reporting NIC capabilities through this attribute is not supported.</p> <p>Bit 0, Value 1 - Reporting NIC capabilities through this attribute is supported.</p> <p>Bit 1, Value 0 - NIC is not TOE capable.</p> <p>Bit 1, Value 1 - NIC is TOE capable.</p> <p>Bit 2, Value 0 - NIC is not iSOE capable.</p> <p>Bit 2, Value 1 - NIC is iSOE capable.</p> <p>Bit 3, Value 0 - NIC is not FCoE capable.</p> <p>Bit 3, Value 1 - NIC is FCoE capable.</p>   | uint 32   |
| NIC TOE Capability  | <p>Defines the TOE capability of the NIC.</p> <p>Values for the NIC TOE Capability property are:</p> <p>0 - NIC/driver does not support querying for capability.</p> <p>1 - NIC/driver supports querying for capability but query returned an error.</p> <p>2 - NIC/driver supports querying for capability and querying indicates that it is capable.</p> <p>4 - NIC/driver supports querying for capability and querying indicates that it is not capable.</p> <p>8 - NIC/driver supports querying for capability but an error prevented querying the NIC/driver.</p> <p>16 - NIC/driver supports querying for capability but NIC/driver did not respond to query.</p> <p><b>NOTE:</b> Boolean value is defined if TOE is enabled (Boolean is <code>TOEEnable</code>).</p> | uint 32   |
| NIC RDMA Capability | <p>Defines the RDMA capability of the NIC.</p> <p>Values for the NIC RDMA Capability property are:</p> <p>0 - NIC/driver does not support querying for capability.</p>   | uint 32   |

**Table 65. DELL NetworkPort Properties (continued)**

| Property             | Description   | Data Type |
|----------------------|---|-----------|
|                      | <p>1 - NIC/driver supports querying for capability but query returned an error.</p> <p>2 - NIC/driver supports querying for capability and querying indicates that it is capable.</p> <p>4 - NIC/driver supports querying for capability and querying indicates that it is not capable.</p> <p>8 - NIC/driver supports querying for capability but an error prevented querying the NIC/driver.</p> <p>16 - NIC/driver supports querying for capability but NIC/driver did not respond to query.</p> <p><b>NOTE:</b> Boolean value is defined if RDMA is enabled (Boolean is RDMAEnable).</p>  |           |
| NIC iSCSI Capability | <p>Defines the iSCSI capability of the NIC.</p> <p>Values for the NIC iSCSI Capability property are:</p> <p>0 - NIC/driver does not support querying for capability.</p> <p>1 - NIC/driver supports querying for capability but query returned an error.</p> <p>2 - NIC/driver supports querying for capability and querying indicates that it is capable.</p> <p>4 - NIC/driver supports querying for capability and querying indicates that it is not capable.</p> <p>8 - NIC/driver supports querying for capability but an error prevented querying the NIC/driver.</p> <p>16 - NIC/driver supports querying for capability but NIC/driver did not respond to query.</p> <p><b>NOTE:</b> Boolean value is defined if iSCSI is enabled (Boolean is iSCSIEnable).</p> | uint 32   |
| NIC Status           | <p>Indicates the status of the NIC or driver.</p> <p>The values for the NIC Status property are:</p> <p>0 - Unknown</p> <p>1 - Connected</p> <p>2 - Disconnected</p> <p>3 - Driver is bad</p> <p>4 - Driver is disabled</p> <p>10 - Hardware is initializing</p>  | uint 32   |

**Table 65. DELL NetworkPort Properties (continued)**

| Property             | Description  | Data Type |
|----------------------|--|-----------|
|                      | 12 - Hardware is closing<br>13 - Hardware is not ready   |           |
| NParEPEnable         | Indicates the mode for NParEP.<br>The values for the NParEPEnable are:<br>0 - Disabled<br>1 - Enabled<br>2 - Unknown | uint 32   |
| BusNumber            | Indicates the PCI bus number.  | uint 8    |
| DeviceNumber         | Indicates the PCI device number.   | uint 8    |
| FunctionNumber       | Indicates the PCI function number.   | uint 8    |
| DriverVersion        | Indicates the NIC driver version.  | string    |
| IPAddressss          | Indicates the NIC IP address.  | string    |
| SubnetMask           | Indicates the NIC subnet mask.   | string    |
| DHCPsServer          | Indicates the DHCP server.   | string    |
| DefaultGateway       | Indicates the default gateway.   | string    |
| CurrentMacAddress    | Indicates the NIC's current MAC address.   | string    |
| OSAdapterDescription | Describes the operating system adapter.  | string    |
| OSProductName        | Describes the product name of the operating system.  | string    |
| ServiceName          | Indicates the service name.  | string    |

## DELL\_PowerConsumptionAmpsSensor

The DELL\_PowerConsumptionAmpsSensor identified in [DELL\\_PowerConsumptionAmpsSensor](#) contains information related to monitoring the power consumption.

The following table describes the property, description, and data type details for DELL\_PowerConsumptionAmpsSensor.

**Class Name:** DELL\_PowerConsumptionAmpsSensor

**Parent Class:** CIM\_Numeric Sensor

**Table 66. DELL\_PowerConsumptionAmpsSensor**

| Property                  | Description                                      | Data Type |
|---------------------------|--|-----------|
| UnitModifier              | See <a href="#">Common Properties of Classes</a> | sint32    |
| CurrentReading            | See <a href="#">Common Properties of Classes</a> | sint32    |
| IsLinear                  | See <a href="#">Common Properties of Classes</a> | Boolean   |
| LowerThresholdNonCritical | See <a href="#">Common Properties of Classes</a> | sint32    |
| UpperThresholdNonCritical | See <a href="#">Common Properties of Classes</a> | sint32    |
| LowerThresholdCritical    | See <a href="#">Common Properties of Classes</a> | sint32    |

**Table 66. DELL\_PowerConsumptionAmpsSensor (continued)**

| Property               | Description                                      | Data Type |
|------------------------|--|-----------|
| UpperThresholdCritical | See <a href="#">Common Properties of Classes</a> | sint32    |

## DELL\_PowerConsumptionWattsSensor

The DELL\_PowerConsumptionWattsSensor identified in [DELL\\_PowerConsumptionWattsSensor](#) contains information related to monitoring the power consumption.

The following table describes the property, description, and data type details for DELL PowerConsumptionWattsSensor.

**Class Name:** DELL\_PowerConsumptionWattsSensor

**Parent Class:** CIM\_Numeric Sensor

**Table 67. DELL PowerConsumptionWattsSensor**

| Property                  | Description                                      | Data Type |
|---------------------------|--|-----------|
| UnitModifier              | See <a href="#">Common Properties of Classes</a> | sint32    |
| CurrentReading            | See <a href="#">Common Properties of Classes</a> | sint32    |
| IsLinear                  | See <a href="#">Common Properties of Classes</a> | Boolean   |
| LowerThresholdNonCritical | See <a href="#">Common Properties of Classes</a> | sint32    |
| UpperThresholdNonCritical | See <a href="#">Common Properties of Classes</a> | sint32    |
| LowerThresholdCritical    | See <a href="#">Common Properties of Classes</a> | sint32    |
| UpperThresholdCritical    | See <a href="#">Common Properties of Classes</a> | sint32    |

## DELL\_PowerConsumptionData

The DELL\_PowerConsumptionData identified in [DELL\\_PowerConsumptionData](#) contains information about the total power consumed from a start time and peak values registered during a time period.

The following table describes the property, description, and data type details for DELL PowerConsumptionData.

**Class Name:** DELL\_PowerConsumptionData

**Parent Class:** CIM\_Logical Device

**Table 68. DELL PowerConsumptionData**

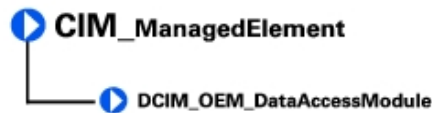
| Property                | Description   | Data Type |
|-------------------------|---|-----------|
| cumulative PowerReading | Indicates the total power consumed from a start time.                       | uint 32   |
| peakAmpReading          | Indicates the time from which the peak amperage reading is being monitored. | uint 16   |
| peakWattReading         | Indicates the time from which the peak watt reading is being monitored.     | uint 16   |
| ResetCounters           | Is the function used to reset the peak readings.                            | uint 32   |

**Table 68. DELL PowerConsumptionData (continued)**

| Property        | Description                                       | Data Type |
|-----------------|---|-----------|
| powerCapSetting | This refers to the user configured power setting. | uint 16   |
| instHeadroom    | This refers to the instantaneous headroom.        | uint 32   |
| peakHeadRoom    | Is the function used to set the power budget.     | uint 32   |

## DCIM\_OEM\_DataAccessModule

The `DCIM_OEM_DataAccessModule` class is derived from the `CIM_ManagedElement` class. This class models hardware information in a proprietary format.



The following table describes the property, description, and data type details for the DCIM OEM DataAccessModule.

**Class Name:** `DCIM_OEM_DataAccessModule`

**Parent Class:** `CIM_ManagedElement`

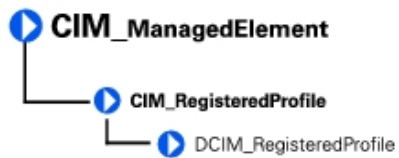
**Table 69. DCIM OEM DataAccessModule**

| Property     | Description  | Data Type |
|--------------|--|-----------|
| InstanceID   | Identifies the instance.   | string    |
| GlobalStatus | Represents the global health status of the system. This property can have the following values:<br><b>0</b> - Other<br><b>1</b> - Unknown<br><b>2</b> - OK<br><b>3</b> - Warning / Non-Critical<br><b>4</b> - Critical<br><b>5</b> - Non-Recoverable<br>.. - Reserved<br>ⓘ <b>NOTE:</b> GlobalStatus property is available only for Linux systems. | sint32    |
| SendCmd      | The SendCmd method is used to invoke proprietary hardware management operation.  | string    |
| iDRACIPv4    | Provides Remote Access controller (iDRAC ) IPv4 address.   | string    |

**Table 69. DCIM OEM DataAccessModule (continued)**

| Property  | Description  | Data Type |
|-----------|--|-----------|
| iDRACIPv6 | Provides Remote Access controller (iDRAC ) IPv6 address. | string    |

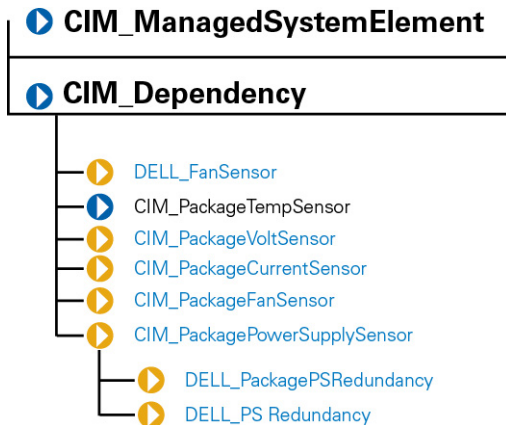
## DCIM\_RegisteredProfile



The `DCIM_RegisteredProfile` class is derived from the `CIM_RegisteredProfile` class. This class advertises the capabilities of `DCIM_OEM_DataAccessModule`.

# CIM\_Dependency

The CIM\_Dependency class is an association used to establish dependency relationships between two managed system elements. The CIM\_Dependency class described in the figure below does not have a parent class because it is a relationship or association between two elements.



Each class derived from the `CIM_Dependency` class has an element called an antecedent that represents the independent object in this association, and another element called a dependent that represents the object that is dependent on the antecedent. For example, consider two managed system elements: `Chassis1` and `PowerSupply3`. `Chassis1` is the antecedent element because a managed power supply would always be either contained in, or grouped with, a chassis.

## Topics:

- [DELL\\_FanSensor](#)
- [CIM\\_PackageTempSensor](#)
- [CIM\\_PackageVoltSensor](#)
- [CIM\\_PackageCurrentSensor](#)
- [CIM\\_PackageFanSensor](#)
- [CIM\\_PackagePowerSupplySensor](#)
- [DELL\\_PackagePSRedundancy](#)
- [DELL\\_PSRedundancy](#)
- [DELL\\_AssociatedSupplyPCAmps](#)
- [DELL\\_AssociatedSystemPCWatts](#)
- [AssociatedSystemPCData](#)
- [DELL\\_PowerProfileData](#)

## DELL\_FanSensor



The `DELL_FanSensor` class described in [DELL\\_FanSensor Properties](#) defines a Dell-specific association between a fan and a sensor. The `CIM_PackageFanSensor` class contains fans that assist in cooling the entire package as opposed to a fan dedicated to cooling only some of the components in the package.

The following table describes the element, description, and parent class details for `DELL_FanSensor`.

**Class Name:** `DELL_FanSensor`

Parent Class: CIM\_Dependency

Table 70. DELL\_FanSensor Properties

| Element    | Description  |
|------------|--|
| Antecedent | CIM_Tachometer refers to the tachometer (fan sensor) that measures the RPM of the fan. |
| Dependent  | CIM_Fan refers to the fan whose revolutions are measured by the tachometer.            |

## CIM\_PackageTempSensor



The CIM\_PackageTempSensor class described in [CIM\\_PackageTempSensor Properties](#) contains temperature sensors that are often installed in a package such as a chassis or a rack to assist in the monitoring of the package in general. This relationship is described by the CIM\_PackageTempSensor association.

The following table describes the element, description, and parent class details for CIM PackageTempSensor.

**Class Name:** CIM\_PackageTempSensor

**Parent Class:** CIM\_Dependency

Table 71. CIM\_PackageTempSensor Properties

| Element    | Description  |
|------------|--|
| Antecedent | CIM_TempSensor refers to the temperature sensor for the package.                         |
| Dependent  | CIM_PhysicalPackage refers to the physical package whose environment is being monitored. |

## CIM\_PackageVoltSensor



The CIM\_PackageVoltSensor class described in [CIM\\_PackageVoltage Properties](#) contains voltage sensors that are often installed in a package such as a chassis or a rack to assist in the monitoring of the package in general. This relationship is described by the CIM\_PackageVoltSensor association.

The following table describes the element, description, and parent class details for CIM PackageVoltage.

**Class Name:** CIM\_PackageVoltSensor

**Parent Class:** CIM\_Dependency

**Table 72. CIM\_PackageVoltage Properties**

| Element    | Description  |
|------------|--|
| Antecedent | CIM_PackageVoltageSensor refers to the voltage sensor for the package.                 |
| Dependent  | CIM_PhysicalPackage refers to the physical package whose voltages are being monitored. |

## CIM\_PackageCurrentSensor

▶ CIM\_ManagedSystemElement

▶ CIM\_Dependency

▶ CIM\_PackageCurrentSensor

The CIM\_PackageCurrentSensor class described in [CIM\\_PackageCurrentSensor Properties](#) contains amperage sensors that are often installed in a package such as a chassis or a rack to assist in the monitoring of the package in general. This relationship is described by the CIM\_PackageCurrentSensor association.

The following table describes the element, description, and parent class details for CIM PackageCurrentSensor.

**Class Name:** CIM\_PackageCurrentSensor

**Parent Class:** CIM\_Dependency

**Table 73. CIM\_PackageCurrentSensor Properties**

| Element    | Description   |
|------------|---|
| Antecedent | CIM_CurrentSensor refers to the amperage sensor for the package.                      |
| Dependent  | CIM_PhysicalPackage refers to the physical package whose amperage is being monitored. |

## CIM\_PackageFanSensor

▶ CIM\_ManagedSystemElement

▶ CIM\_Dependency

▶ CIM\_PackageFanSensor

The CIM\_PackageFanSensor class described in [CIM\\_PackageFanSensor Properties](#) contains fan sensors that monitor the whole package.

The following table describes the element, description, and parent class details for CIM PackageFanSensor Properties.

**Parent Class:** CIM\_Dependency

**Class Name:** CIM\_PackageFanSensor

**Table 74. CIM\_PackageFanSensor Properties**

| Element    | Description  |
|------------|--|
| Antecedent | CIM_Fan refers to the cooling device for the package.                                    |
| Dependent  | CIM_PhysicalPackage refers to the physical package whose environment is being monitored. |

# CIM\_PackagePowerSupplySensor

▶ CIM\_ManagedSystemElement

▶ CIM\_Dependency

▶ CIM\_PackagePowerSupplySensor

The CIM\_PackagePowerSupplySensor class described in [CIM\\_PackagePowerSupplySensor Properties](#) contains power supplies that provide power to the whole package.

The following table describes the element, description, and parent class details for CIM PackagePowerSupplySensor Properties.

**Class Name:** CIM\_PackagePowerSupplySensor

**Parent Class:** CIM\_Dependency

**Table 75. CIM\_PackagePowerSupplySensor Properties**

| Element    | Description   |
|------------|---|
| Antecedent | CIM_PowerSupplySensor refers to the power supply sensor that monitors wattage for the entire package. |
| Dependent  | CIM_PhysicalPackage refers to the package whose wattage is being monitored.                           |

# DELL\_PackagePSRedundancy

▶ CIM\_ManagedSystemElement

▶ CIM\_Dependency

▶ DELL\_PackagePSRedundancy

The DELL\_PackagePSRedundancy class described in [DELL\\_PackagePSRedundancy Properties](#) defines what constitutes a power supply redundancy for an entire package.

The following table describes the element, description, and parent class details for DELL PackagePSRedundancy Properties.

**Class Name:** DELL\_PackagePSRedundancy

**Parent Class:** CIM\_Dependency

**Table 76. DELL\_PackagePSRedundancy Properties**

| Element    | Description  |
|------------|--|
| Antecedent | DELL_PSRedundancyGroup refers to power supplies that deliver wattage for the entire package. |
| Dependent  | CIM_PhysicalPackagerefers to the package to which the wattage is being supplied.             |

# DELL\_PSRedundancy

▶ CIM\_ManagedSystemElement

▶ CIM\_Dependency

▶ DELL\_PSRedundancy

The `DELL_PSRedundancy` class described in [DELL\\_PSRedundancy Properties](#) defines what constitutes a power supply redundancy for Dell systems.

The following table describes the element, description, and parent class details for DELL PSRedundancy Properties.

**Class Name:** `DELL_PSRedundancy`

**Parent Class:** `CIM_Dependency`

**Table 77. DELL\_PSRedundancy Properties**

| Element    | Description  |
|------------|--|
| Antecedent | <code>CIM_PowerSupplySensor</code> refers to the power supply sensor that monitors wattage for the entire package. |
| Dependent  | <code>CIM_PhysicalPackage</code> refers to the package whose wattage is being monitored.                           |

## DELL\_AssociatedSupplyPCAmps

The `DELL_AssociatedSupplyPCAmps` class described in [DELL\\_AssociatedSupplyPCAmps](#) is a `PowerConsumptionAmpsSensor` associated with a `CIM_PowerSupply` which is defined by this class.

The following table describes the property, description, and data type details for DELL AssociatedSupplyPCAmps.

**Class Name:** `DELL_AssociatedSupplyPCAmps`

**Parent Class:** `CIM_Dependency`

**Table 78. DELL\_AssociatedSupplyPCAmps**

| Property   | Description  | Data Type |
|------------|--|-----------|
| Antecedent | Indicates the <code>PowerSupply</code> instance.   | uint 16   |
| Dependent  | Indicates the <code>PowerConsumptionAmpsSensor</code> associated with the <code>CIM_PowerSupply</code> . | uint 16   |

## DELL\_AssociatedSystemPCWatts

The `DELL_AssociatedSystemPCWatts` class described in [DELL\\_AssociatedSystemPCWatts](#) is a `PowerConsumptionWattsSensor` associated with a `Dell_System` which is defined by this class.

The following table describes the property, description, and data type details for DELL AssociatedSystemPCWatts.

**Class Name:** `DELL_AssociatedSystemPCWatts`

**Parent Class:** `CIM_Dependency`

**Table 79. DELL\_AssociatedSystemPCWatts**

| Property   | Description  | Data Type |
|------------|--|-----------|
| Antecedent | Indicates the <code>Dell_System</code> instance.                                   | uint 16   |
| Dependent  | Indicates the <code>PowerConsumptionWattsSensor</code> associated with the system. | uint 16   |

# AssociatedSystemPCData

The AssociatedSystemPCData identified in AssociatedSystemPCData is a PowerConsumptionData associated with a Dell\_System which is defined by this class.

The following table describes the property, description, and data type details for the AssociatedSystemPCData.

**Class Name:** DELL\_AssociatedSupplyPCamps

**Parent Class:** CIM\_Dependency

**Table 80. AssociatedSystemPCData**

| Property   | Description  | Data Type |
|------------|--|-----------|
| Antecedent | Indicates the Dell_System instance.                                  | uint 16   |
| Dependent  | Indicates the PowerConsumptionData associated with the Power Supply. | uint 16   |

# DELL\_PowerProfileData

The DELL\_PowerProfileData identified in DELL\_PowerProfileData contains information related to power profiling and power knob data.

The following table describes the property, description and data type details for DELL PowerProfileData class.

**Class Name:** DELL\_PowerProfileData

**Parent Class:** CIM\_LogicalDevice

**Table 81. DELL\_PowerProfileData**

| Property          | Description  | Data Type |
|-------------------|--|-----------|
| chassisIndex      | Indicates the chassisIndex for this power profile.         | uint 8    |
| supportedProfile  | Indicates the supported profiles.                          | uint 16   |
| profileSetting    | Indicates the Profile setting.                             | uint 16   |
| customCPUCaps     | Indicates the Custom Profile CPU management capability.    | uint 16   |
| customCPUSettings | Indicates the Custom Profile CPU management setting.       | uint 16   |
| customMemCaps     | Indicates the Custom Profile memory management capability. | uint 16   |
| customMemSettings | Indicates the Custom Profile memory management capability. | uint 16   |
| customFanSettings | Indicates the Custom Profile fan management setting.       | uint 16   |