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.

<table>
<thead>
<tr>
<th>Class Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIM_Keyboard</td>
<td>38</td>
</tr>
<tr>
<td>CIM_PowerSupply</td>
<td>39</td>
</tr>
<tr>
<td>CIM_Controller</td>
<td>40</td>
</tr>
<tr>
<td>CIM_ParallelController</td>
<td>40</td>
</tr>
<tr>
<td>CIM_SerialController</td>
<td>41</td>
</tr>
<tr>
<td>CIM_PCIController</td>
<td>42</td>
</tr>
<tr>
<td>CIM_PCIDevice</td>
<td>43</td>
</tr>
<tr>
<td>CIM_PCIEBridge</td>
<td>43</td>
</tr>
<tr>
<td>CIM_Processor</td>
<td>44</td>
</tr>
<tr>
<td>CIM_StorageExtent</td>
<td>52</td>
</tr>
<tr>
<td>CIM_Memory</td>
<td>53</td>
</tr>
<tr>
<td>CIM_CacheMemory</td>
<td>53</td>
</tr>
<tr>
<td>DELL_SoftwareFeature</td>
<td>54</td>
</tr>
<tr>
<td>CIM_BIOSElement</td>
<td>55</td>
</tr>
<tr>
<td>CIM_SoftwareFeature</td>
<td>56</td>
</tr>
<tr>
<td>CIM_SystemResource</td>
<td>57</td>
</tr>
<tr>
<td>CIM_IRQ</td>
<td>57</td>
</tr>
<tr>
<td>CIM_MemoryMappedIOM</td>
<td>59</td>
</tr>
<tr>
<td>CIM_DMA</td>
<td>59</td>
</tr>
<tr>
<td>CIM_RedundancyGroup</td>
<td>60</td>
</tr>
<tr>
<td>CIM_ExtraCapacityGroup</td>
<td>61</td>
</tr>
<tr>
<td>DELL_PSRedundancyGroup</td>
<td>61</td>
</tr>
<tr>
<td>DELL_FanRedundancyGroup</td>
<td>61</td>
</tr>
<tr>
<td>CIM_EnabledLogicalElement</td>
<td>62</td>
</tr>
<tr>
<td>CIM_ServiceAccessPoint</td>
<td>62</td>
</tr>
<tr>
<td>CIM_RemoteServiceAccessPoint</td>
<td>63</td>
</tr>
<tr>
<td>DELL_RemoteServiceAccessPort</td>
<td>64</td>
</tr>
</tbody>
</table>

### 4 Dell-Defined Classes

<table>
<thead>
<tr>
<th>Class Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELL_PostLog</td>
<td>66</td>
</tr>
<tr>
<td>DELL_CMApplication</td>
<td>66</td>
</tr>
<tr>
<td>DELL_CMDevice</td>
<td>66</td>
</tr>
<tr>
<td>DELL_CMDeviceApplication</td>
<td>66</td>
</tr>
<tr>
<td>DELL_CMIInventory</td>
<td>66</td>
</tr>
<tr>
<td>DELL_CMOS</td>
<td>66</td>
</tr>
<tr>
<td>DELL_CMProductInfo</td>
<td>66</td>
</tr>
<tr>
<td>DELL_BIOSExtensions</td>
<td>67</td>
</tr>
<tr>
<td>DELL_BIOSettings</td>
<td>67</td>
</tr>
<tr>
<td>DELL_SDCardDevice</td>
<td>67</td>
</tr>
<tr>
<td>DELL_NetworkPort</td>
<td>67</td>
</tr>
<tr>
<td>DELL_PowerConsumptionAmpsSensor</td>
<td>67</td>
</tr>
<tr>
<td>DELL_PowerConsumptionWattsSensor</td>
<td>67</td>
</tr>
<tr>
<td>DELL_PowerConsumptionData</td>
<td>67</td>
</tr>
<tr>
<td>DCIM_OEM_DataAccessModule</td>
<td>67</td>
</tr>
<tr>
<td>DCIM_RegainedProfile</td>
<td>67</td>
</tr>
</tbody>
</table>

---

4 Dell-Defined Classes

<table>
<thead>
<tr>
<th>Class Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELL_PowerConsumptionWattsSensor</td>
<td>67</td>
</tr>
<tr>
<td>DELL_PowerConsumptionData</td>
<td>67</td>
</tr>
<tr>
<td>DCIM_OEM_DataAccessModule</td>
<td>67</td>
</tr>
<tr>
<td>DCIM_RegainedProfile</td>
<td>67</td>
</tr>
</tbody>
</table>
## 5 CIM_Dependency

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELL_FanSensor</td>
<td>79</td>
</tr>
<tr>
<td>CIM_PackageTempSensor</td>
<td>80</td>
</tr>
<tr>
<td>CIM_PackageVoltSensor</td>
<td>80</td>
</tr>
<tr>
<td>CIM_PackageCurrentSensor</td>
<td>81</td>
</tr>
<tr>
<td>CIM_PackageFanSensor</td>
<td>81</td>
</tr>
<tr>
<td>CIM_PackagePowerSupplySensor</td>
<td>82</td>
</tr>
<tr>
<td>DELL_PackagePSRedundancy</td>
<td>82</td>
</tr>
<tr>
<td>DELL_PSRedundancy</td>
<td>82</td>
</tr>
<tr>
<td>DELL_AssociatedSupplyPCAmps</td>
<td>83</td>
</tr>
<tr>
<td>DELL_AssociatedSystemPCWatts</td>
<td>84</td>
</tr>
<tr>
<td>AssociatedSystemPCData</td>
<td>84</td>
</tr>
<tr>
<td>DELL_PowerProfileData</td>
<td>85</td>
</tr>
</tbody>
</table>
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.

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 system.

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
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**
- **DELL_CMApplication**
- **DELL_CMDevice**
- **DELL_CMDeviceApplications**
- **DELL_CMInventory**
- **DELL_CMOS**
- **DELL_CMProductInfo**

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 the 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 dccim32 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

<table>
<thead>
<tr>
<th>Antecedent</th>
<th>Dependent</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>CIM_PackageCurrentSensor</code></td>
<td><code>CIM_PhysicalPackage</code></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caption</td>
<td>Describes the object using a short textual description (one-line string).</td>
<td>string</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>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.</td>
<td>string</td>
</tr>
<tr>
<td>CSCreationClassName</td>
<td>Indicates the computer system's creation class name.</td>
<td>string</td>
</tr>
<tr>
<td>CSName</td>
<td>Indicates the computer system's name.</td>
<td>string</td>
</tr>
<tr>
<td>CurrentReading</td>
<td>Indicates the actual current value indicated by the sensor in amperes.</td>
<td>sint32</td>
</tr>
<tr>
<td>Description</td>
<td>Provides a textual description of the object.</td>
<td>string</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
<td>Data Type</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>LowerThresholdNonCritical</td>
<td>If current reading is between lower threshold noncritical and upper threshold noncritical, the current state is normal. See Figure 3-2.</td>
<td>sint32</td>
</tr>
<tr>
<td>LowerThresholdCritical</td>
<td>If the current reading is between upper threshold critical and upper threshold fatal, the current state is critical. See Figure 3-2.</td>
<td>sint32</td>
</tr>
<tr>
<td>IsLinear</td>
<td>Indicates that the sensor is linear over its dynamic range.</td>
<td>Boolean</td>
</tr>
<tr>
<td>Manufacturer</td>
<td>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.</td>
<td>string</td>
</tr>
<tr>
<td>Name</td>
<td>Defines the label by which the object is known. When subclassed, the Name property can be overridden to be a Key property.</td>
<td>string</td>
</tr>
<tr>
<td>Status</td>
<td>Provides a string indicating the status of the component. Status values include:</td>
<td>string</td>
</tr>
<tr>
<td></td>
<td><strong>Operational Status Values:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• OK indicates that the object is functioning normally.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Degraded means that the item is functioning, but not optimally.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Stressed indicates that the element is functioning, but needs attention. Examples of Stressed states are overloaded, overheated, and so on.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Nonoperational Status Values:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Non-recover means that a nonrecoverable error has occurred.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Error means that an element has encountered an operational condition that is severe as compared to its normal mode of operation.</td>
<td></td>
</tr>
<tr>
<td>SystemCreationClassName</td>
<td>Indicates the system’s creation class name.</td>
<td>string</td>
</tr>
<tr>
<td>UnitModifier</td>
<td>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.</td>
<td>sint32</td>
</tr>
</tbody>
</table>
### Property Description Data Type

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>UpperThresholdCritical</td>
<td>If the current reading is between upper threshold critical and upper threshold fatal, the current status is critical. See Figure 3-2.</td>
<td>sint32</td>
</tr>
<tr>
<td>UpperThresholdNonCritical</td>
<td>If the current reading is between lower threshold noncritical and lower threshold critical, the current status is noncritical. See Figure 3-2.</td>
<td>sint32</td>
</tr>
<tr>
<td>Version</td>
<td>Version should be in the form &lt;major&gt;.&lt;minor&gt;.&lt;revision&gt; or &lt;major&gt;.&lt;minor&gt;&lt;letter&gt;&lt;revision&gt;; for example, 1.2.3 or 1.2a3.</td>
<td>string</td>
</tr>
</tbody>
</table>

### 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:

- **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.

1. **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_Physical Element**

*CIM_PhysicalElement* is a CIM-defined class. The *CIM_PhysicalElement* class contains the subclasses shown in Figure 1.

**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*

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.
### Table 3. CIM_PhysicalElement Properties

**Class Name:** CIM_PhysicalElement  
**Parent Class:** CIM_ManagedSystemElement

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreationClassName</td>
<td>See Common Properties of Classes</td>
<td></td>
</tr>
<tr>
<td>Manufacturer</td>
<td>See Common Properties of Classes</td>
<td></td>
</tr>
<tr>
<td>Model</td>
<td>The name by which the physical element is generally known.</td>
<td>string</td>
</tr>
<tr>
<td>SerialNumber</td>
<td>A manufacturer-allocated number used to identify the physical element.</td>
<td>string</td>
</tr>
<tr>
<td>Tag</td>
<td>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.</td>
<td>string</td>
</tr>
</tbody>
</table>

---

**CIM_PhysicalPackage**

The **CIM_PhysicalPackage** class listed in Table above represents physical elements that contain or host other components. Examples are a rack enclosure or an adapter card with multiple functions.
Table 4. CIM_PhysicalPackage Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Removable</td>
<td>A CIM_PhysicalPackage 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.</td>
<td>Boolean</td>
</tr>
<tr>
<td>Replaceable</td>
<td>A CIM_PhysicalPackage is replaceable if it is possible to substitute a physically different element for the original element, 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.</td>
<td>Boolean</td>
</tr>
</tbody>
</table>

CIM_PhysicalFrame

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.

Table 5. CIM_PhysicalFrame Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>LockPresent</td>
<td>Indicates whether the frame is protected with a lock.</td>
<td>Boolean</td>
</tr>
<tr>
<td>AudibleAlarm</td>
<td>Indicates whether the frame is equipped with an audible alarm.</td>
<td>Boolean</td>
</tr>
<tr>
<td>VisibleAlarm</td>
<td>Indicates that the equipment includes a visible alarm.</td>
<td>Boolean</td>
</tr>
<tr>
<td>SecurityBreach</td>
<td>An enumerated, integer-valued property indicating that a physical breach of the frame is in progress. Values for the SecurityBreach property are:</td>
<td>uint16</td>
</tr>
</tbody>
</table>

1 - Other 2 - Unknown 3 - No breach 4 - Breach attempted 5 - Breach successful
### CIM_PhysicalFrame

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsLocked</td>
<td>Indicates that the frame is currently locked.</td>
<td>Boolean</td>
</tr>
</tbody>
</table>

### CIM_Chassis

**CIM_ManagedSystemElement**

**CIM_PhysicalElement**

**CIM_PhysicalPackage**

**CIM_PhysicalFrame**

**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.

**Table 6. CIM_Chassis Parent Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChassisTypes</td>
<td>Values for the ChassisTypes property are:</td>
<td>uint16</td>
</tr>
<tr>
<td>1</td>
<td>Other</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Unknown</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Mini-tower</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Tower</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Space-saving</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Main system chassis</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Expansion chassis</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Subchassis</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Space-saving</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Main system chassis</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Expansion chassis</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Subchassis</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Bus expansion chassis</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Peripheral chassis</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Storage chassis</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Rack-mount chassis</td>
<td></td>
</tr>
</tbody>
</table>
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.

### Table 7. **DELL_Chassis Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AssetTag</td>
<td>Indicates the container <strong>AssetTag</strong> string. This asset tag string is assigned by the system administrator.</td>
<td>string</td>
</tr>
<tr>
<td>SystemClass</td>
<td>Refers to the system type that is installed and running the instrumentation. Values for the <strong>SystemClass</strong> property are:</td>
<td>uint16</td>
</tr>
<tr>
<td>SystemID</td>
<td>Indicates the system identifier code.</td>
<td>uint16</td>
</tr>
<tr>
<td>LogFormat</td>
<td>Defines whether the event log data is unicode formatted or binary (raw). Values for the event <strong>LogFormat</strong> property are:</td>
<td>uint16</td>
</tr>
<tr>
<td>FanStatus</td>
<td>Indicates the global status of fan sensors.</td>
<td>string</td>
</tr>
<tr>
<td>TempStatus</td>
<td>Indicates the global status of temperature sensors.</td>
<td>string</td>
</tr>
<tr>
<td>VoltStatus</td>
<td>Indicates the global status of voltage sensors.</td>
<td>string</td>
</tr>
<tr>
<td>AmpStatus</td>
<td>Indicates the global status of current sensors.</td>
<td>string</td>
</tr>
<tr>
<td>PsStatus</td>
<td>Indicates the global status of power supplies.</td>
<td>string</td>
</tr>
</tbody>
</table>
### DELL_Chassis

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>MemStatus</td>
<td>Indicates the global status of memory devices.</td>
<td>string</td>
</tr>
<tr>
<td>ProcStatus</td>
<td>Indicates the global status of processor devices.</td>
<td>string</td>
</tr>
<tr>
<td>FanRedStatus</td>
<td>Indicates the global status of the cooling unit.</td>
<td>string</td>
</tr>
<tr>
<td>PsRedStatus</td>
<td>Indicates the global status of the power unit.</td>
<td>string</td>
</tr>
<tr>
<td>IsDefaultThrSupported</td>
<td>Indicates whether resetting default thresholds are supported.</td>
<td>Boolean</td>
</tr>
<tr>
<td>ChassisSystemProperties</td>
<td>Indicates chassis characteristics, such as energy smart and so on.</td>
<td>uint16</td>
</tr>
<tr>
<td>ChassisSystemRevision</td>
<td>Indicates the chassis revision.</td>
<td>uint16</td>
</tr>
<tr>
<td>EsmLogStatus</td>
<td>Indicates the global status of ESM log.</td>
<td>string</td>
</tr>
<tr>
<td>MemoryRedStatus</td>
<td>Indicates the global status of memory redundancy.</td>
<td>string</td>
</tr>
<tr>
<td>ChassisExpressServiceCode</td>
<td>Indicates the chassis express service code.</td>
<td>string</td>
</tr>
<tr>
<td>ChassisNodeID</td>
<td>Chassis Node ID</td>
<td>string</td>
</tr>
</tbody>
</table>

---

### 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**

<table>
<thead>
<tr>
<th>Class Name:</th>
<th>CIM_PhysicalComponent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Class:</td>
<td>CIM_PhysicalElement</td>
</tr>
</tbody>
</table>

---

### CIM_Chip

The **CIM_Chip** class represents...
The **CIM_Chip** class listed in *Chip Properties* represents any type of integrated circuit hardware, including ASICs, processors, memory chips, and so on.

**Table 9. CIM_Chip Properties**

**Class Name:** CIM_Chip  
**Parent Class:** CIM_PhysicalComponent

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>FormFactor</td>
<td>0 — Unknown</td>
<td>uint16</td>
</tr>
<tr>
<td>1 — Other</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 — SIP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 — DIP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 — ZIP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 — SOJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 — Proprietary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 — SIMM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 — DIMM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9 — TSOP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 — PGA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11 — RIMM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12 — SODIMM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13 — SRIMM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14 — SMD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15 — SSMP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16 — QFP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17 — TQFP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18 — SOIC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19 — LCC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20 — PLCC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21 — BGA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22 — FPBGA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23 — LGA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24 — FB-DIMM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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.

Table 10. CIM_PhysicalMemory Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>FormFactor</td>
<td>See Chip Properties</td>
<td>uint16</td>
</tr>
<tr>
<td>MemoryType</td>
<td>Indicates the type of physical memory. Values for the MemoryType property are:</td>
<td>uint16</td>
</tr>
<tr>
<td></td>
<td>0 - Unknown</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 - Other</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 - DRAM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 - Synchronous DRAM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 - Cache DRAM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5 - EDO</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6 - EDRAM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7 - VRAM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8 - SRAM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9 - RAM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10 - ROM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>11 — Flash</td>
<td></td>
</tr>
<tr>
<td></td>
<td>12 - EEPROM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>13 - FEPROM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>14 - EPROM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>15 - CDRAM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>16 - 3DRAM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>17 - SDRAM</td>
<td></td>
</tr>
</tbody>
</table>
18 - SGRAM
19 - RDRAM
20 - DDR
21 - DDR2
22 - DDR2 FB-DIMM
24 - DDR3
25 - FBD2
26 - DDR4

TotalWidth  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 DataWidth property.

DataWidth  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 memory is solely used to provide error correction bits.

Speed  Indicates the speed of the physical memory, in nanoseconds.

Rank  The Rank values of DIMM are:

<table>
<thead>
<tr>
<th>Rank</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Unknown</td>
</tr>
<tr>
<td>1</td>
<td>Single</td>
</tr>
<tr>
<td>2</td>
<td>Dual</td>
</tr>
<tr>
<td>4</td>
<td>Quad</td>
</tr>
<tr>
<td>8</td>
<td>Octal</td>
</tr>
<tr>
<td>16</td>
<td>Hexa</td>
</tr>
</tbody>
</table>

SpeedAsString  Indicates the accurate speed of the physical memory, in string format (with units).

Capacity  Indicates the total capacity of this physical memory, in bytes.

BankLabel  A string identifying the physically labeled bank where the memory is located, for example, "Bank 0" or "Bank A."

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.

**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.

**Manufacturer**: Indicates the manufacturer of the physical memory.

**SerialNumber**: Indicates the serial number of the physical memory.

---

**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.

---

**Table 11. CIM_PhysicalConnector Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectorPinout</td>
<td>A free-form string describing the pin configuration and signal usage of a physical connector.</td>
<td>string</td>
</tr>
<tr>
<td>ConnectorType</td>
<td>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 Connector Type Values for the values of the ConnectorType property.</td>
<td>uint16</td>
</tr>
<tr>
<td>Value</td>
<td>Connector Type</td>
<td>Value</td>
</tr>
<tr>
<td>-------</td>
<td>---------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>0</td>
<td>Unknown</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>Other</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>Male</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>Female</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>Shielded</td>
<td>14</td>
</tr>
<tr>
<td>5</td>
<td>Unshielded</td>
<td>15</td>
</tr>
<tr>
<td>6</td>
<td>SCSI (A) High-Density (50 pins)</td>
<td>16</td>
</tr>
<tr>
<td>7</td>
<td>SCSI (A) Low-Density (50 pins)</td>
<td>17</td>
</tr>
<tr>
<td>8</td>
<td>SCSI (P) High-Density (68 pins)</td>
<td>18</td>
</tr>
<tr>
<td>9</td>
<td>SCSI SCA-I (80 pins)</td>
<td>19</td>
</tr>
<tr>
<td>10</td>
<td>SCSI SCA-II (80 pins)</td>
<td>20</td>
</tr>
<tr>
<td>11</td>
<td>Fibre Channel (DB-9 Copper)</td>
<td>21</td>
</tr>
<tr>
<td>12</td>
<td>Fibre Channel (Fiber Optical)</td>
<td>22</td>
</tr>
<tr>
<td>13</td>
<td>Fibre Channel SCAII (40 pins)</td>
<td>23</td>
</tr>
<tr>
<td>14</td>
<td>Fibre Channel SCAII (20 pins)</td>
<td>24</td>
</tr>
<tr>
<td>15</td>
<td>Fibre Channel BNC</td>
<td>25</td>
</tr>
<tr>
<td>16</td>
<td>ATA 3-1/2 inch (40 pins)</td>
<td>26</td>
</tr>
<tr>
<td>17</td>
<td>ATA 2-1/2 inch (44 pins)</td>
<td>27</td>
</tr>
<tr>
<td>18</td>
<td>ATA-2</td>
<td>28</td>
</tr>
<tr>
<td>19</td>
<td>ATA-3</td>
<td>29</td>
</tr>
<tr>
<td>20</td>
<td>ATA-66</td>
<td>30</td>
</tr>
<tr>
<td>21</td>
<td>DB-9</td>
<td>31</td>
</tr>
<tr>
<td>22</td>
<td>DB-15</td>
<td>32</td>
</tr>
<tr>
<td>23</td>
<td>DB-25</td>
<td>33</td>
</tr>
<tr>
<td>24</td>
<td>DB-36</td>
<td>34</td>
</tr>
<tr>
<td>25</td>
<td>RS-232C</td>
<td>35</td>
</tr>
<tr>
<td>26</td>
<td>RS-422</td>
<td>36</td>
</tr>
<tr>
<td>27</td>
<td>RS-423</td>
<td>37</td>
</tr>
<tr>
<td>28</td>
<td>RS-485</td>
<td>38</td>
</tr>
<tr>
<td>29</td>
<td>RS-449</td>
<td>39</td>
</tr>
<tr>
<td></td>
<td></td>
<td>40</td>
</tr>
<tr>
<td></td>
<td></td>
<td>41</td>
</tr>
<tr>
<td></td>
<td></td>
<td>42</td>
</tr>
<tr>
<td></td>
<td></td>
<td>43</td>
</tr>
<tr>
<td></td>
<td></td>
<td>44</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45</td>
</tr>
<tr>
<td></td>
<td></td>
<td>46</td>
</tr>
<tr>
<td></td>
<td></td>
<td>47</td>
</tr>
<tr>
<td></td>
<td></td>
<td>48</td>
</tr>
<tr>
<td></td>
<td></td>
<td>49</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50</td>
</tr>
<tr>
<td></td>
<td></td>
<td>51</td>
</tr>
<tr>
<td></td>
<td></td>
<td>52</td>
</tr>
<tr>
<td></td>
<td></td>
<td>53</td>
</tr>
<tr>
<td></td>
<td></td>
<td>54</td>
</tr>
<tr>
<td></td>
<td></td>
<td>55</td>
</tr>
<tr>
<td></td>
<td></td>
<td>56</td>
</tr>
<tr>
<td></td>
<td></td>
<td>57</td>
</tr>
<tr>
<td></td>
<td></td>
<td>58</td>
</tr>
<tr>
<td></td>
<td></td>
<td>59</td>
</tr>
<tr>
<td></td>
<td></td>
<td>60</td>
</tr>
<tr>
<td></td>
<td></td>
<td>61</td>
</tr>
<tr>
<td></td>
<td></td>
<td>62</td>
</tr>
<tr>
<td></td>
<td></td>
<td>63</td>
</tr>
<tr>
<td></td>
<td></td>
<td>64</td>
</tr>
<tr>
<td></td>
<td></td>
<td>65</td>
</tr>
<tr>
<td></td>
<td></td>
<td>66</td>
</tr>
<tr>
<td></td>
<td></td>
<td>67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>68</td>
</tr>
<tr>
<td></td>
<td></td>
<td>69</td>
</tr>
<tr>
<td></td>
<td></td>
<td>70</td>
</tr>
<tr>
<td></td>
<td></td>
<td>71</td>
</tr>
<tr>
<td></td>
<td></td>
<td>72</td>
</tr>
<tr>
<td></td>
<td></td>
<td>73</td>
</tr>
<tr>
<td></td>
<td></td>
<td>74</td>
</tr>
<tr>
<td></td>
<td></td>
<td>75</td>
</tr>
<tr>
<td></td>
<td></td>
<td>76</td>
</tr>
<tr>
<td></td>
<td></td>
<td>77</td>
</tr>
<tr>
<td></td>
<td></td>
<td>78</td>
</tr>
<tr>
<td></td>
<td></td>
<td>79</td>
</tr>
<tr>
<td></td>
<td></td>
<td>80</td>
</tr>
<tr>
<td></td>
<td></td>
<td>81</td>
</tr>
<tr>
<td></td>
<td></td>
<td>82</td>
</tr>
<tr>
<td></td>
<td></td>
<td>83</td>
</tr>
<tr>
<td></td>
<td></td>
<td>84</td>
</tr>
<tr>
<td></td>
<td></td>
<td>85</td>
</tr>
<tr>
<td></td>
<td></td>
<td>86</td>
</tr>
<tr>
<td></td>
<td></td>
<td>87</td>
</tr>
<tr>
<td></td>
<td></td>
<td>88</td>
</tr>
<tr>
<td></td>
<td></td>
<td>89</td>
</tr>
<tr>
<td></td>
<td></td>
<td>90</td>
</tr>
<tr>
<td></td>
<td></td>
<td>91</td>
</tr>
<tr>
<td></td>
<td></td>
<td>92</td>
</tr>
<tr>
<td></td>
<td></td>
<td>93</td>
</tr>
<tr>
<td></td>
<td></td>
<td>94</td>
</tr>
<tr>
<td></td>
<td></td>
<td>95</td>
</tr>
<tr>
<td></td>
<td></td>
<td>96</td>
</tr>
<tr>
<td></td>
<td></td>
<td>97</td>
</tr>
<tr>
<td></td>
<td></td>
<td>98</td>
</tr>
<tr>
<td></td>
<td></td>
<td>99</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>101</td>
</tr>
<tr>
<td></td>
<td></td>
<td>102</td>
</tr>
<tr>
<td></td>
<td></td>
<td>103</td>
</tr>
<tr>
<td></td>
<td></td>
<td>104</td>
</tr>
<tr>
<td></td>
<td></td>
<td>105</td>
</tr>
<tr>
<td></td>
<td></td>
<td>106</td>
</tr>
<tr>
<td></td>
<td></td>
<td>107</td>
</tr>
<tr>
<td></td>
<td></td>
<td>108</td>
</tr>
<tr>
<td></td>
<td></td>
<td>109</td>
</tr>
</tbody>
</table>

**Table 12. Connector Type Values**
The CIM_Slot class 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.

### Table 13. CIM_Slot Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectorType</td>
<td>See Connector Type Values</td>
<td>uint16</td>
</tr>
<tr>
<td>SupportsHotPlug</td>
<td>Indicates whether the slot supports hot-plug adapter cards.</td>
<td>Boolean</td>
</tr>
<tr>
<td>MaxDataWidth</td>
<td>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:</td>
<td>uint16</td>
</tr>
<tr>
<td></td>
<td>0 - Unknown</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 - Other</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8 - bits</td>
<td></td>
</tr>
<tr>
<td></td>
<td>16 - bits</td>
<td></td>
</tr>
<tr>
<td></td>
<td>32 - bits</td>
<td></td>
</tr>
<tr>
<td></td>
<td>64 - bits</td>
<td></td>
</tr>
<tr>
<td></td>
<td>128 - bits</td>
<td></td>
</tr>
<tr>
<td>SystemSlotType</td>
<td>Indicates the type of system slot. Values for the SystemSlotType property are as follows:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 - Other</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 - Unknown</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 - ISA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 - MCA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5 - EISA</td>
<td></td>
</tr>
</tbody>
</table>
6 - PCI
7 - PCMCIA
8 - VL-VESA
9 - Proprietary
10 - Processor card Slot
11 - Proprietary memory card slot
12 - I/O Riser card slot
13 - NuBus
14 - PCI - 66MHz capable
15 - AGP
16 - AGP 2X
17 - AGP 4X
18 - PCI-X
19 - AGP 8X
160 - PC-98/C20
161 - PC-98/C24
162 - PC-98/E
163 - PC-98/Local bus
164 - PC-98/Card
165 - PCI Express
166 - PCI Express x1
167 - PCI Express x2
168 - PCI Express x4
169 - PCI Express x8
170 - PCI Express x16
171 - PCI Express Gen 2
172 - PCI Express Gen 2 x1
173 - PCI Express Gen 2 x2
174 - PCI Express Gen 2 x4
175 - PCI Express Gen 2 x8
176 - PCI Express Gen 2 x16
CIM_LogicalElement is a CIM-defined class containing the subclasses described in the figure below.
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
### CIM_LogicalElement

![CIM_ManagedSystemElement](image)

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**

<table>
<thead>
<tr>
<th>Class Name:</th>
<th>CIM_LogicalElement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Class:</td>
<td>CIM_ManagedSystemElement</td>
</tr>
</tbody>
</table>

### CIM_System

![CIM_ManagedSystemElement](image)

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.

**Table 15. CIM_System Properties**

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

**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>
<thead>
<tr>
<th>Class Name:</th>
<th>CIM_ComputerSystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Class:</td>
<td>CIM_System</td>
</tr>
</tbody>
</table>

**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>
<thead>
<tr>
<th>Class Name:</th>
<th>DELL_System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Class:</td>
<td>CIM_ComputerSystem</td>
</tr>
</tbody>
</table>

**CIM_LogicalDevice**

<table>
<thead>
<tr>
<th>Class Name:</th>
<th>CIM_LogicalDevice</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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.

**Table 18. CIM_Logical Device Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>SystemCreationClassName</td>
<td>See Common Properties of Classes</td>
<td>string</td>
</tr>
<tr>
<td>SystemName</td>
<td>Indicates the scoping system’s name.</td>
<td>string</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>See Common Properties of Classes</td>
<td>string</td>
</tr>
<tr>
<td>DeviceID</td>
<td>Identifies an address or other identifying information to uniquely name the logical device.</td>
<td>string</td>
</tr>
</tbody>
</table>

**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.

**Table 19. CIM_FRU Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRUInformationState</td>
<td>Indicates the state and availability of FRU information.</td>
<td>uint 16</td>
</tr>
<tr>
<td>FRUDeviceName</td>
<td>Indicates the device name of the FRU.</td>
<td>string</td>
</tr>
<tr>
<td>FRUManufacturingDateName</td>
<td>Indicates the manufacturing date of the FRU in ticks.</td>
<td>datetime</td>
</tr>
</tbody>
</table>
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.

### Table 20. CIM_Sensor Properties

**Class Name:** CIM_Sensor  
**Parent Class:** CIM_LogicalDevice

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>SensorType</td>
<td>Indicates the type of the sensor, for example, voltage or temperature sensor. Values for the SensorType property are:</td>
<td>uint16</td>
</tr>
<tr>
<td></td>
<td>0 - Unknown</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 - Other</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 - Temperature sensors measure the environmental temperature.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 - Voltage sensors measure electrical voltage.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 - Current sensors measure current readings.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5 - Tachometers measure speed/revolutions of a device. For example, a fan device can have an associated tachometer that measures its speed.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6 - Batteries maintain the time and date and save the system's</td>
<td></td>
</tr>
</tbody>
</table>
BIOS configuration when the system is turned off.

**OtherSensorType**
Description Indicates the type of sensor when the SensorType property is set to *Other*.

**PossibleStates**
Enumerates the string outputs of the sensor. For example, a NumericSensor can report states based on threshold readings.

**CurrentState**
Indicates the current state of the sensor. This value is always one of the Possible States.

**PollingInterval**
Indicates the polling interval, in nanoseconds, that the sensor hardware or instrumentation uses to determine the current state of the sensor.

---

### 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.

#### Table 21. CIM_DiscreteSensor Properties

**Class Name**: CIM_DiscreteSensor  
**Parent Class**: CIM_Sensor

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>CurrentReading</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>PossibleValues</td>
<td>Enumerates the string outputs that can be reported by the sensor.</td>
<td>sint32</td>
</tr>
</tbody>
</table>

---

### CIM_NumericSensor

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

#### Table 21. CIM_NumericSensor Properties

**Class Name**: CIM_NumericSensor  
**Parent Class**: CIM_Sensor

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>PossibleValues</td>
<td>Enumerates the string outputs that can be reported by the sensor.</td>
<td>sint32</td>
</tr>
</tbody>
</table>
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**

![Diagram showing threshold ranges](image)

**Table 22. CIM_NumericSensor Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class Name:</td>
<td>CIM_NumericSensor</td>
<td></td>
</tr>
<tr>
<td>Parent Class:</td>
<td>CIM_Sensor</td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
<td>Type</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>UnitModifier</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>CurrentReading</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>IsLinear</td>
<td></td>
<td>Boolean</td>
</tr>
<tr>
<td>LowerThresholdNonCritical</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>UpperThresholdNonCritical</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>LowerThresholdCritical</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>UpperThresholdCritical</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>SupportedThresholds</td>
<td>An array representing the thresholds supported by this sensor. The supported values are as follows:</td>
<td>uint16</td>
</tr>
<tr>
<td></td>
<td>1 - LowerThresholdNonCritical</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 - UpperThresholdNonCritical</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 - LowerThresholdCritical</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 - UpperThresholdCritical</td>
<td></td>
</tr>
<tr>
<td>EnabledThresholds</td>
<td>An array representing the thresholds that are currently enabled for this sensor. Enabled threshold values are as follows:</td>
<td>uint16</td>
</tr>
<tr>
<td></td>
<td>1 - LowerThresholdNonCritical</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 - UpperThresholdNonCritical</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 - LowerThresholdCritical</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 - UpperThresholdCritical</td>
<td></td>
</tr>
<tr>
<td>SettableThresholds</td>
<td>An array representing the writable thresholds supported by this sensor. Settable threshold values are:</td>
<td>uint16</td>
</tr>
<tr>
<td></td>
<td>1 - LowerThresholdNonCritical</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 - UpperThresholdNonCritical</td>
<td></td>
</tr>
</tbody>
</table>
CIM_TemperatureSensor

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

Table 23. CIM_TemperatureSensor Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>UnitModifier</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>CurrentReading</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>IsLinear</td>
<td>See Common Properties of Classes</td>
<td>Boolean</td>
</tr>
<tr>
<td>LowerThresholdNonCritical</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>UpperThresholdNonCritical</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>LowerThresholdCritical</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>UpperThresholdCritical</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
</tbody>
</table>

CIM_CurrentSensor

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

Table 24. CIM_CurrentSensor Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>UnitModifier</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>CurrentReading</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
</tbody>
</table>
The **CIM_VoltageSensor** class described in Table below contains sensors that measure voltage and return a value in volts.

### Table 25. CIM_VoltageSensor Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsLinear</td>
<td>See Common Properties of Classes</td>
<td>Boolean</td>
</tr>
<tr>
<td>LowerThresholdNonCritical</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>UpperThresholdNonCritical</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>LowerThresholdCritical</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>UpperThresholdCritical</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
</tbody>
</table>

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.

### Table 26. CIM_Tachometer Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>UnitModifier</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>CurrentReading</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>IsLinear</td>
<td>See Common Properties of Classes</td>
<td>Boolean</td>
</tr>
<tr>
<td>LowerThresholdNonCritical</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>UpperThresholdNonCritical</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>LowerThresholdCritical</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>UpperThresholdCritical</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
</tbody>
</table>
**Parent Class:** CIM\_NumericSensor

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>SensorType</td>
<td>See Common Properties of Classes</td>
<td>uint16</td>
</tr>
<tr>
<td>UnitModifier</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>CurrentReading</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>IsLinear</td>
<td>See Common Properties of Classes</td>
<td>Boolean</td>
</tr>
<tr>
<td>LowerThresholdNonCritical</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>UpperThresholdNonCritical</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
</tbody>
</table>

---

**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.

**Table 27. CIM\_WatchDog Properties**

<table>
<thead>
<tr>
<th>Class Name: CIM_WatchDog</th>
<th>Parent Class: CIM_LogicalDevice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>MonitoredEntity</td>
<td>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 MonitoredEntity property are:</td>
</tr>
<tr>
<td></td>
<td>1 - Unknown</td>
</tr>
<tr>
<td></td>
<td>2 - Other</td>
</tr>
<tr>
<td></td>
<td>3 - Operating System</td>
</tr>
<tr>
<td>MonitoredEntity Description</td>
<td>A string describing additional textual information about the monitored entity.</td>
</tr>
<tr>
<td>TimeoutInterval</td>
<td>Indicates the time-out interval used by the watchdog, in microseconds.</td>
</tr>
<tr>
<td>TimerResolution</td>
<td>Indicates the resolution of the watchdog timer. For example, if this value is 100, then</td>
</tr>
</tbody>
</table>
CIM_WatchDog

Parent Class: CIM_LogicalDevice

Property Description Data Type
the timer can expire anytime between −100 microseconds and +100 microseconds.

CIM_CoolingDevice

Class Name: CIM_CoolingDevice
Parent Class: CIM_LogicalDevice

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

CIM_Fan

Class Name: CIM_Fan
Parent Class: CIM_CoolingDevice

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.

**CIM_UserDevice**

CIM_UserDevice

CIM_ManagedSystemElement

CIM_LogicalElement

CIM_LogicalDevice

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**.

**Table 30. CIM_UserDevice Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsLocked</td>
<td>Indicates if the device is locked, preventing user input or output.</td>
<td>Boolean</td>
</tr>
</tbody>
</table>

**CIM_PointingDevice**

CIM_PointingDevice

CIM_ManagedSystemElement

CIM_LogicalElement

CIM_LogicalDevice

CIM_UserDevice

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.

**Table 31. CIM_PointingDevice Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>PointingType</td>
<td>Indicates the type of pointing device. Values for the <strong>PointingType</strong> property are: 1 — Other</td>
<td>uint16</td>
</tr>
</tbody>
</table>
2 — Unknown
3 — Mouse
4 — Trackball
5 — Trackpoint
6 — Glidepoint
7 — Touch pad
8 — Touch screen
9 — Mouse — optical sensor

NumberOfButtons: Indicates the number of buttons. If the CIM_PointingDevice has no buttons, a value of 0 is returned.

Handedness: Integer indicating if the CIM_PointingDevice is configured for right- or left-handed operation. Values for the Handedness property are as follows:
0 — Unknown
1 — Not applicable
2 — Right-handed operation
3 — Left-handed operation

---

**CIM_Keyboard**

The CIM_Keyboard class described in Table below includes devices that allow users to enter data.

### Table 32. CIM_Keyboard Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>NumberOfFunctionKeys</td>
<td>Indicates the number of function keys on the keyboard.</td>
<td>uint16</td>
</tr>
<tr>
<td>Layout</td>
<td>A free-form string indicating the format and layout of the keyboard.</td>
<td>string</td>
</tr>
<tr>
<td>Password</td>
<td>An integer indicating if a hardware-level password is enabled at the keyboard,</td>
<td>uint16</td>
</tr>
</tbody>
</table>
preventing local input. Values for the **Password** property are:

- 1 — Other
- 2 — Unknown
- 3 — Disabled
- 4 — Enabled
- 5 — Not implemented

**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.

### Table 33. CIM_PowerSupply Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsSwitchingSupply</td>
<td>Indicates that the power supply is a switching power supply and not a linear power supply.</td>
<td>Boolean</td>
</tr>
<tr>
<td>Range1InputVoltageLow</td>
<td>Indicates the low voltage in millivolts of input voltage range 1 for this power supply. A value of 0 denotes unknown.</td>
<td>uint32</td>
</tr>
<tr>
<td>Range1InputVoltageHigh</td>
<td>Indicates the high voltage in millivolts of input voltage range 1 for this power supply. A value of 0 denotes unknown.</td>
<td>uint32</td>
</tr>
<tr>
<td>ActiveInputVoltage</td>
<td>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 <strong>ActiveInputVoltage</strong> property are:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 — Other</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 — Unknown</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 — Range 1</td>
<td></td>
</tr>
</tbody>
</table>
TotalOutputPower

- Represents the total output power of the power supply in milliwatts. A value of 0 denotes that the power output is unknown.

PMCapable

- Indicates the Power Monitoring capability.

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.

Table 34. CIM_Controller Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProtocolSupported</td>
<td>The protocol used by the controller to access controlled devices. Values for the ProtocolSupported property are:</td>
<td>uint16</td>
</tr>
<tr>
<td></td>
<td>1 — Other</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 — Unknown</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 — PCI</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 — Parallel protocol</td>
<td></td>
</tr>
</tbody>
</table>

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.
### Table 35. CIM_ParallelController Properties

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

### 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.

### Table 36. CIM_SerialController Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaxBaudRate</td>
<td>Indicates the maximum baud rate in bits per second supported by the serial controller.</td>
<td>uint32</td>
</tr>
<tr>
<td>Security</td>
<td>An enumeration indicating the operational security for the controller. Values for the Security property are:</td>
<td>uint16</td>
</tr>
<tr>
<td></td>
<td>1 — Other</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 — Unknown</td>
<td></td>
</tr>
</tbody>
</table>
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.

**Table 37. CIM_PCIController Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>CommandRegister</td>
<td>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. Values for the <strong>CommandRegister</strong> property are:</td>
<td>uint16</td>
</tr>
<tr>
<td>CommandRegister</td>
<td>0 — Unknown</td>
<td></td>
</tr>
<tr>
<td>CommandRegister</td>
<td>1 — Other</td>
<td></td>
</tr>
<tr>
<td>CommandRegister</td>
<td>2 — Supports 66 MHz</td>
<td></td>
</tr>
<tr>
<td>CommandRegister</td>
<td>3 — Supports user-definable features</td>
<td></td>
</tr>
<tr>
<td>CommandRegister</td>
<td>4 — Supports fast back-to-back transactions</td>
<td></td>
</tr>
<tr>
<td>CommandRegister</td>
<td>5 — PCI-X capable</td>
<td></td>
</tr>
<tr>
<td>CommandRegister</td>
<td>6 — PCI power management supported</td>
<td></td>
</tr>
<tr>
<td>CommandRegister</td>
<td>7 — Message signaled interrupts supported</td>
<td></td>
</tr>
</tbody>
</table>
8 — Parity error recovery capable
9 — AGP supported
10 — Vital product data supported
11 — Provides slot identification
12 — Hot swap supported

CIM_PCIDevice

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

Table 38. CIM_PCIDevice Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>BaseAddress</td>
<td>Identifies an array of up to six double-word base memory addresses.</td>
<td>uint32</td>
</tr>
<tr>
<td>SubsystemID</td>
<td>Identifies a subsystem identifier code.</td>
<td>uint16</td>
</tr>
<tr>
<td>SubsystemVendorID</td>
<td>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.</td>
<td>uint16</td>
</tr>
<tr>
<td>ExpansionROMBaseAddress</td>
<td>Identifies a double-word expansion ROM base memory address.</td>
<td>uint32</td>
</tr>
</tbody>
</table>

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.

### Table 39. CIM_PCIBridge Properties

<table>
<thead>
<tr>
<th>Class Name:</th>
<th>CIM_PCIBridge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Class:</td>
<td>CIM_PCIController</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>BaseAddress</td>
<td>Identifies an array of double-word base memory addresses.</td>
<td>uint32</td>
</tr>
<tr>
<td>BridgeType</td>
<td>Indicates the type of bridge. A bridge is PCI to &lt;value&gt;, except for the Host, which is a host-to-PCI bridge. Values for the <strong>BridgeType</strong> property are as follows: 0 — Host 1 — ISA 128 — Other</td>
<td>uint16</td>
</tr>
<tr>
<td>BaseAddress</td>
<td>Identifies an array of double-word base memory addresses.</td>
<td>uint32</td>
</tr>
</tbody>
</table>

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

### Table 40. CIM_Processor Properties

<table>
<thead>
<tr>
<th>Class Name:</th>
<th>CIM_Processor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Class:</td>
<td>CIM_LogicalDevice</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role</td>
<td>A string describing the role of the microprocessor, for example, central microprocessor or math processor.</td>
<td>string</td>
</tr>
<tr>
<td>UpgradeMethod</td>
<td>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: 1 - Other</td>
<td>uint16</td>
</tr>
</tbody>
</table>
2 - Unknown
3 - Daughter board
4 - ZIF socket
5 - Replacement/piggy back
6 - None
7 - LIF socket
8 - Slot 1
9 - Slot 2
10 - 370-pin socket
19 - Socket mPGA604
20 - Socket LGA771
21 - Socket LGA775
22 - Socket S1
23 - Socket AM2
24 - Socket F (1207)
25 - Socket LGA1366

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaxClockSpeed</td>
<td>Indicates the maximum speed (in MHz) of this microprocessor.</td>
<td>uint32</td>
</tr>
<tr>
<td>Core count</td>
<td>Indicates the number of core processors detected.</td>
<td>uint16</td>
</tr>
<tr>
<td>CoreEnabledCount</td>
<td>Indicates the number of core processors enabled.</td>
<td>uint16</td>
</tr>
<tr>
<td>CurrentClockSpeed</td>
<td>Indicates the current speed (in MHz) of this microprocessor.</td>
<td>uint32</td>
</tr>
<tr>
<td>DataWidth</td>
<td>Indicates the processor data width in bits.</td>
<td>uint16</td>
</tr>
<tr>
<td>AddressWidth</td>
<td>Indicates the processor address width in bits.</td>
<td>uint16</td>
</tr>
<tr>
<td>Stepping</td>
<td>Indicates the revision level of the processor within the microprocessor family.</td>
<td>string</td>
</tr>
<tr>
<td>UniqueID</td>
<td>Identifies a globally unique identifier for the microprocessor. This identifier may only be unique within a microprocessor family.</td>
<td>string</td>
</tr>
<tr>
<td>Brand</td>
<td>Indicates the brand name of the processor.</td>
<td>string</td>
</tr>
<tr>
<td>Model</td>
<td>Indicates the model name of the processor.</td>
<td>string</td>
</tr>
<tr>
<td>ExtendedCharacteristics</td>
<td>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: Bit 0 — Virtualization Technology (VT) supported</td>
<td>uint16</td>
</tr>
</tbody>
</table>
Bit 1 — Demand-Based Switching (DBS) supported

Bit 2 — eXecute Disable (XD) supported

Bit 3 — Hyper Threading (HT) supported

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:

Bit 0 — Virtualization Technology (VT) enabled

Bit 1 — Demand-Based Switching (DBS) enabled

Bit 2 — eXecute Disable (XD) enabled

Bit 3 — Hyper Threading (HT) enabled

CPUStatus

Indicates the current status of the microprocessor. 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:

0 - Unknown

1 - Microprocessor enabled

2 - Microprocessor disabled by user through BIOS setup

3 - Microprocessor disabled by BIOS (POST error)

4 - Microprocessor is idle

5 - Other

Family

Refers to the processor family type. Values for the Family property are as follows:

1 - Other

2 - Unknown

3 - 8086

4 - 80286

5 - 80386

6 - 80486

7 - 8087

8 - 80287

9 - 80387
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>80487</td>
</tr>
<tr>
<td>11</td>
<td>Pentium Brand</td>
</tr>
<tr>
<td>12</td>
<td>Pentium Pro</td>
</tr>
<tr>
<td>13</td>
<td>Pentium II</td>
</tr>
<tr>
<td>14</td>
<td>Pentium processor with MMX technology</td>
</tr>
<tr>
<td>15</td>
<td>Celeron</td>
</tr>
<tr>
<td>16</td>
<td>Pentium II Xeon</td>
</tr>
<tr>
<td>17</td>
<td>Pentium III</td>
</tr>
<tr>
<td>18</td>
<td>M1 family</td>
</tr>
<tr>
<td>19</td>
<td>M2 family</td>
</tr>
<tr>
<td>24</td>
<td>AMD Duron processor</td>
</tr>
<tr>
<td>25</td>
<td>K5 family</td>
</tr>
<tr>
<td>26</td>
<td>K6 family</td>
</tr>
<tr>
<td>27</td>
<td>K6-2</td>
</tr>
<tr>
<td>28</td>
<td>K6-3</td>
</tr>
<tr>
<td>29</td>
<td>AMD Athlon processor family</td>
</tr>
<tr>
<td>30</td>
<td>AMD29000 family</td>
</tr>
<tr>
<td>31</td>
<td>K6-2+</td>
</tr>
<tr>
<td>32</td>
<td>Power PC family</td>
</tr>
<tr>
<td>33</td>
<td>Power PC 601</td>
</tr>
<tr>
<td>34</td>
<td>Power PC 603</td>
</tr>
<tr>
<td>35</td>
<td>Power PC 603+</td>
</tr>
<tr>
<td>36</td>
<td>Power PC 604</td>
</tr>
<tr>
<td>37</td>
<td>Power PC 620</td>
</tr>
<tr>
<td>38</td>
<td>Power PC X704</td>
</tr>
<tr>
<td>39</td>
<td>Power PC 750</td>
</tr>
<tr>
<td>40</td>
<td>Intel Core Duo processor</td>
</tr>
<tr>
<td>41</td>
<td>Intel Core Duo mobile processor</td>
</tr>
<tr>
<td>42</td>
<td>Intel Core Solo mobile processor</td>
</tr>
<tr>
<td>43</td>
<td>Intel Atom processor</td>
</tr>
<tr>
<td>48</td>
<td>Alpha family</td>
</tr>
<tr>
<td>49</td>
<td>Alpha 21064</td>
</tr>
</tbody>
</table>
50 - Alpha 21066
51 - Alpha 21164
52 - Alpha 21164
53 - Alpha 21164a
54 - Alpha 21264
55 - Alpha 21364
60 - AMD Opteron 4100 Series processor
64 - MIPS family
65 - MIPS R4000
66 - MIPS R4200
67 - MIPS R4400
68 - MIPS R4600
69 - MIPS R10000
80 - SPARC family
81 - SuperSPARC
82 - microSPARC II
83 - microSPARC IIep
84 - UltraSPARC
85 - UltraSPARC II
86 - UltraSPARC III
87 - UltraSPARC III
88 - UltraSPARC IIIi
96 - 68040
97 - 68xxx family
98 - 68000
99 - 68010
100 - 68020
101 - 68030
112 - Hobbit family
120 - Crusoe 5000 family
121 - Crusoe 3000 family
122 - Efficeon 8000 family
128 - Weitek
130 - Itanium processor
131 - AMD Athlon 64 processor family
132 - AMD Opteron processor family
133 - AMD Sempron processor family
134 - AMD Turion 64 Mobile technology
135 - Dual-Core AMD Opteron processor family
136 - AMD Athlon 64 X2 Dual-Core processor family
137 - AMD Turion 64 X2 Mobile technology
138 - Quad-Core AMD Opteron processor family
139 - Third-Generation AMD Opteron processor family
140 - AMD Phenom FX Quad-Core processor family
141 - AMD Phenom X4 Quad-Core processor family
142 - AMD Phenom X2 Dual-Core processor family
143 - AMD Athlon X2 Dual-Core processor family
144 - PA-RISC family
145 - PA-RISC 8500
146 - PA-RISC 8000
147 - PA-RISC 7300LC
148 - PA-RISC 7200
149 - PA-RISC 7100LC
150 - PA-RISC 7100
160 - V30 family
161 - Quad-Core Intel Xeon processor 3200 Series
162 - Dual-Core Intel Xeon processor 3000 Series
163 - Quad-Core Intel Xeon processor 5300 Series
164 - Dual-Core Intel Xeon processor 5100 Series
165 - Dual-Core Intel Xeon processor 5000 Series
166 - Dual-Core Intel Xeon processor LV
167 - Dual-Core Intel Xeon processor ULV
168 - Dual-Core Intel Xeon processor 7100 Series
169 - Quad-Core Intel Xeon processor 5400 Series
170 - Quad-Core Intel Xeon processor
171 - Dual-Core Intel Xeon processor 5200 Series
172 - Dual-Core Intel Xeon processor 7200 Series
173 - Quad-Core Intel Xeon processor 7300 Series
174 - Quad-Core Intel Xeon processor 7400 Series
175 - Multi-Core Intel Xeon processor 7400 Series
176 - Pentium III Xeon
177 - Pentium III Processor with Intel SpeedStep Technology
178 - Pentium 4
179 - Intel Xeon
180 - AS400 family
181 - Intel Xeon Processor MP
182 - AMD Athlon XP family
183 - AMD Athlon MP family
184 - Intel Itanium 2
185 - Intel Pentium M processor
186 - Intel Celeron D processor
187 - Intel Pentium D processor
188 - Intel Pentium Extreme Edition processor
189 - Intel Core 2 processor
190 - Intel Core 2 processor
- Intel Core 2 Solo processor
- Intel Core 2 Extreme processor
- Intel Core 2 Quad processor
- Intel Core 2 Extreme mobile processor
- Intel Core 2 Duo mobile processor
- Intel Core 2 Solo mobile processor
- Intel Core i7 processor
- Dual-Core Intel Celeron processor
- S/390 and zSeries family
- ESA/390 G4
- ESA/390 G5
- ESA/390 G6
- z/Architecture base
- CEh 206 Intel Core i3 processor
- Multi-Core Intel Xeon processor
- Dual-Core Intel Xeon processor 3xxx Series
- Quad-Core Intel Xeon processor 3xxx Series
- D9h 217 VIA Nano processor family
- Dual-Core Intel Xeon processor 5xxx Series
- Quad-Core Intel Xeon processor 5xxx Series
- Dual-Core Intel Xeon processor 7xxx Series
- Dual-Core Intel Xeon processor 7xxx Series
- Multi-Core Intel Xeon processor 7xxx Series
- E0h 224 Multi-Core Intel Xeon processor 3400 Series
- Embedded AMD Opteron Quad-Core processor family
- AMD Phenom Triple-Core processor family
- AMD Turion Ultra Dual-Core Mobile processor family
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

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class Name:</td>
<td>CIM_StorageExtent</td>
</tr>
<tr>
<td>Parent Class:</td>
<td>CIM_LogicalDevice</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Class Name</strong></td>
<td><code>CIM_Memory</code></td>
<td></td>
</tr>
<tr>
<td><strong>Parent Class</strong></td>
<td><code>CIM_StorageExtent</code></td>
<td></td>
</tr>
</tbody>
</table>

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.

### Table 43. CIM_CacheMemory Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Class Name</strong></td>
<td><code>CIM_CacheMemory</code></td>
<td></td>
</tr>
<tr>
<td><strong>Parent Class</strong></td>
<td><code>CIM_Memory</code></td>
<td></td>
</tr>
<tr>
<td><strong>Level</strong></td>
<td>Defines if this is the primary, secondary, or tertiary cache. Values for the <code>Level</code> property are as follows:</td>
<td>uint16</td>
</tr>
<tr>
<td></td>
<td>1: Other</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2: Unknown</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3: Primary</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4: Secondary</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5: Tertiary</td>
<td></td>
</tr>
</tbody>
</table>
**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 WritePolicy property are as follows:

1. Other
2. Unknown
3. Write-back
4. Write-through
5. Varies with address
6. Determination per I/O

**CacheType**

Defines if this cache is for instruction caching, data caching, or both (unified). Values for the CacheType property are as follows:

1. Other
2. Unknown
3. Instruction
4. Data
5. Unified

**LineSize**

Indicates the size, in bytes, of a single cache bucket or line.

**ReadPolicy**

Defines the policy used by the cache for handling read requests. Values for the ReadPolicy property are as follows:

1. Other
2. Unknown
3. Read
4. Read-ahead
5. Read and read-ahead
6. Determination per I/O
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.

### Table 44. DELL_SoftwareFeature Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>OmsaURL</td>
<td>Defines the URL for Server Administrator.</td>
<td>string</td>
</tr>
<tr>
<td>Language</td>
<td>Sets the language for systems management information.</td>
<td>string</td>
</tr>
<tr>
<td>AgentVersion</td>
<td>Defines the version information of local CIM agent (same as ISVC version.)</td>
<td>string</td>
</tr>
</tbody>
</table>

### 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.

### Table 45. CIM_BIOSElement Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>Provides the product version information.</td>
<td>string</td>
</tr>
<tr>
<td>Manufacturer</td>
<td>See Common Properties of Classes</td>
<td>string</td>
</tr>
<tr>
<td>PrimaryBIOS</td>
<td>Specifies whether a given BIOS is the primary BIOS for the system. When the value = TRUE, the BIOS is the primary BIOS.</td>
<td>Boolean</td>
</tr>
</tbody>
</table>
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.

**Table 46. CIM_SoftwareFeature Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>IdentifyingNumber</td>
<td>Provides product identification such as a serial number of the software</td>
<td>string</td>
</tr>
<tr>
<td>ProductName</td>
<td>Identifies the commonly used product name.</td>
<td>string</td>
</tr>
<tr>
<td>Vendor</td>
<td>Identifies the name of the product’s supplier.</td>
<td>string</td>
</tr>
<tr>
<td></td>
<td>Corresponds to the vendor property in the product object in the DMTF solution exchange standard.</td>
<td></td>
</tr>
<tr>
<td>Version</td>
<td>Identifies the product version information.</td>
<td>string</td>
</tr>
<tr>
<td></td>
<td>Corresponds to the version property in the product object in the DMTF solution exchange standard.</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Defines the label by which the object is known to the users. This label is a user-defined name that uniquely identifies the element.</td>
<td>string</td>
</tr>
</tbody>
</table>

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.
Table 47. DELL_SoftwareFeature Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>OmsaURL</td>
<td>Defines the URL for Server Administrator.</td>
<td>string</td>
</tr>
<tr>
<td>Language</td>
<td>Sets the language for systems management information.</td>
<td>string</td>
</tr>
<tr>
<td>AgentVersion</td>
<td>Defines the version information of local CIM agent (same as ISVC version.)</td>
<td>string</td>
</tr>
</tbody>
</table>

**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

<table>
<thead>
<tr>
<th>Class Name:</th>
<th>CIM_SystemResource</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Class:</td>
<td>CIM_LogicalElement</td>
</tr>
</tbody>
</table>

**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.

Table 49. CIM_IRQ Properties

<table>
<thead>
<tr>
<th>Class Name:</th>
<th>CIM_IRQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Class:</td>
<td>CIM_SystemResource</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>CSCreationClassName</td>
<td>See Common Properties of Classes</td>
</tr>
</tbody>
</table>
CSName: See Common Properties of Classes, string

CreationClassName: See Common Properties of Classes, string

IRQNumber: Identifies the interrupt request number, uint32

Availability: Indicates the availability of the IRQ. Values for the Availability property are as follows:

1 - Other
2 - Unknown
3 - Available
4 - In use/not available
5 - In use and available

TriggerLevel: Indicates if the interrupt is triggered by the hardware signal going high or low. Values for the TriggerLevel property are as follows:

1 - Other
2 - Unknown
3 - Active low
4 - Active high

TriggerType: Indicates if edge (value=4) or level triggered (value=3) interrupts occur.

1 - Other
2 - Unknown
3 - Level
4 - 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
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.

### Table 50. CIM_MemoryMappedIO Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCreationClassName</td>
<td>See Common Properties of Classes</td>
<td>string</td>
</tr>
<tr>
<td>CSName</td>
<td>See Common Properties of Classes</td>
<td>string</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>See Common Properties of Classes</td>
<td>string</td>
</tr>
<tr>
<td>StartingAddress</td>
<td>Identifies the starting address of memory mapped I/O.</td>
<td>uint64</td>
</tr>
<tr>
<td>EndingAddress</td>
<td>Identifies the ending address of memory mapped I/O.</td>
<td>uint64</td>
</tr>
<tr>
<td>MappedResource</td>
<td>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: 1- Other 2- Mapped memory 3 - I/O mapped to memory space 4 - I/O mapped to port space</td>
<td>uint16</td>
</tr>
</tbody>
</table>

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.
### Table 51. CIM_DMA Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCreationClassName</td>
<td>See Common Properties of Classes</td>
<td>string</td>
</tr>
<tr>
<td>CSName.</td>
<td>See Common Properties of Classes</td>
<td>string</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>See Common Properties of Classes</td>
<td>string</td>
</tr>
<tr>
<td>DMAChannel</td>
<td>A part of the object’s key value, the DMA channel number.</td>
<td>uint32</td>
</tr>
<tr>
<td>Availability</td>
<td>Availability of the DMA. <strong>Availability</strong> values are defined as follows:</td>
<td>uint16</td>
</tr>
<tr>
<td></td>
<td>- 1 - Other</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- 2 - Unknown</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- 3 - Available</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- 4 - In Use/Not Available</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- 5 - In Use and Available/Shareable</td>
<td></td>
</tr>
</tbody>
</table>

### 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.

### Table 52. CIM_RedundancyGroup Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreationClassName</td>
<td>See Common Properties of Classes</td>
<td>string</td>
</tr>
<tr>
<td>Name</td>
<td>Serves as the key for the redundancy group’s instance in an enterprise environment.</td>
<td>string</td>
</tr>
<tr>
<td>RedundancyStatus</td>
<td>Provides information on the state of the redundancy group. Values for the RedundancyStatus property are as follows:</td>
<td>uint16</td>
</tr>
<tr>
<td></td>
<td>0 - Unknown</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 - Other</td>
<td></td>
</tr>
</tbody>
</table>
2. Fully redundant. Fully redundant - all of the configured redundancy is still available.

3. Degraded redundancy. Degraded redundancy - that some failures have been experienced but some reduced amount of redundancy is still available.

4. 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.

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.

Table 53. CIM_ExtraCapacityGroup Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>MinNumberNeeded</td>
<td>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.</td>
<td>uint32</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Class Name:</th>
<th>DELL_PSRedundancyGroup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Class:</td>
<td>CIM_ExtraCapacityGroup</td>
</tr>
</tbody>
</table>

DELL_FanRedundancyGroup

The DELL_FanRedundancyGroup described in Table below defines what constitutes fan redundancy in a system.

Table 55. DELL_FanRedundancyGroup Properties

<table>
<thead>
<tr>
<th>Class Name:</th>
<th>DELL_FanRedundancyGroup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Class:</td>
<td>CIM_ExtraCapacityGroup</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Class Name:</th>
<th>CIM_EnabledLogicalElementGroup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Class:</td>
<td>CIM_LogicalElementGroup</td>
</tr>
</tbody>
</table>

CIM_ServiceAccessPoint

CIM_ManagedSystemElement

CIM_LogicalElement

CIM_EnabledLogicalElement

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

<table>
<thead>
<tr>
<th>Class Name:</th>
<th>CIM_ServiceAccessPoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Class:</td>
<td>CIM_EnabledLogicalElement</td>
</tr>
</tbody>
</table>

**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.

### Table 58. CIM_RemoteServiceAccessPoint Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccessInfo</td>
<td>Describes accessing or addressing of information for a remote connection. This can be a host name, network address, and other similar information.</td>
<td>string</td>
</tr>
<tr>
<td>InfoFormat</td>
<td>Indicates an enumerated integer describing the format and interpretation of the AccessInfo property. This property can have the following values:&lt;br&gt;1 - Other&lt;br&gt;2 - Host Name&lt;br&gt;3 - IPv4 Address&lt;br&gt;4 - IPv6 Address&lt;br&gt;5 - IPX Address&lt;br&gt;6 - DECnet Address&lt;br&gt;7 - SNA Address</td>
<td>uint16</td>
</tr>
</tbody>
</table>
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.

### Table 59. DELL_RemoteServiceAccessPort Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>PortName</td>
<td>Displays the name of the service access port.</td>
<td>string</td>
</tr>
<tr>
<td>VersionString</td>
<td>Indicates the version of the access point service.</td>
<td>string</td>
</tr>
<tr>
<td>RemoteAccessType</td>
<td>Indicated the type of remote access service.</td>
<td>uint16</td>
</tr>
</tbody>
</table>

This property can have the following values:

- 0: BMC
<table>
<thead>
<tr>
<th>BladeFormFactor</th>
<th>Type of Blade Form Factor. This property can have the following values:</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>singleWidthHalfHeight</td>
</tr>
<tr>
<td>1</td>
<td>dualWidthHalfHeight</td>
</tr>
<tr>
<td>2</td>
<td>singleWidthFullHeight</td>
</tr>
<tr>
<td>3</td>
<td>dualWidthFullHeight</td>
</tr>
<tr>
<td>4</td>
<td>singleWidthQuarterHeight</td>
</tr>
<tr>
<td>5</td>
<td>1UHalfWidth</td>
</tr>
<tr>
<td>6</td>
<td>1UQuarterWidth</td>
</tr>
<tr>
<td>7</td>
<td>1UFullWidth</td>
</tr>
<tr>
<td>255</td>
<td>notApplicable</td>
</tr>
</tbody>
</table>

- **IMC**
- **CMC**
- **iDRAC6**
- **iDRAC6 for modular systems**
- **BMC**
- **iDRAC7**
- **iDRAC7 for modular systems**
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.

The DELL_EsmLog class described in DELL_EsmLog Properties records failure threshold violations collected by Server Administrator’s embedded server management (ESM) capabilities.

Table 60. DELL_EsmLog Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>recordNumber</td>
<td>Provides an index to the ESM table.</td>
<td>uint32</td>
</tr>
<tr>
<td>logRecord</td>
<td>Provides the ESM message content.</td>
<td>string</td>
</tr>
<tr>
<td>eventTime</td>
<td>Indicates the time that the message is generated.</td>
<td>datetime</td>
</tr>
<tr>
<td>status</td>
<td>Indicates the severity of the event that caused the log to be generated.</td>
<td>string</td>
</tr>
</tbody>
</table>

Topics:
- DELL_PostLog
- DELL_CMApplication
- DELL_CMDevice
- DELL_CMDeviceApplication
- DELL_CMInventory
- DELL_CMOS
- DELL_CMProductInfo
- DELL_BIOSExtensions
- DELL_BIOSSettings
- DELL_SDCardDevice
- DELL_NetworkPort
- DELL_PowerConsumptionAmpsSensor
- DELL_PowerConsumptionWattsSensor
DELL_PowerConsumptionData

DCIM_OEM_DataAccessModule

DCIM_RegisteredProfile

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

Class Name: DELL_PostLog

Parent Class: None

DELL_CMApplication

NOTE: Dell-updateable components, such as BIOS and firmware, are considered applications.

The DELL_CMApplication class described in DELL_CMApplication contains information related to the Dell change management applications.

Table 62. DELL_CMApplication

Class Name: DELL_CMApplication

Parent Class: None

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>componentType</td>
<td>Defines the application type.</td>
<td>string</td>
</tr>
<tr>
<td>subComponentID</td>
<td>Defines an application string.</td>
<td>string</td>
</tr>
<tr>
<td>version</td>
<td>Indicates the current version of the application.</td>
<td>string</td>
</tr>
<tr>
<td>name</td>
<td>Indicates the name of the application.</td>
<td>string</td>
</tr>
<tr>
<td>deviceKey</td>
<td>Indicates the device key of the application.</td>
<td>string</td>
</tr>
</tbody>
</table>
DELL_CMDevice

The DELL_CMDevice class described in DELL_CMDevice Properties contains information related to the Dell change management device.

Table 63. DELL_CMDevice Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>componentID</td>
<td>Defines a component string.</td>
<td>string</td>
</tr>
<tr>
<td>name</td>
<td>Indicates the name of the device.</td>
<td>string</td>
</tr>
<tr>
<td>vendorID</td>
<td>Defines an ID for vendor supplying the device.</td>
<td>string</td>
</tr>
<tr>
<td>subVendorID</td>
<td>Defines an ID for an additional vendor supplying the device.</td>
<td>string</td>
</tr>
<tr>
<td>deviceID</td>
<td>Indicates the ID of the device.</td>
<td>string</td>
</tr>
<tr>
<td>subDeviceID</td>
<td>Indicates the ID for additional device.</td>
<td>string</td>
</tr>
<tr>
<td>bus</td>
<td>Indicates the PCI bus number.</td>
<td>string</td>
</tr>
<tr>
<td>device</td>
<td>Indicates the PCI device number.</td>
<td>string</td>
</tr>
<tr>
<td>function</td>
<td>Indicates the PCI function number.</td>
<td>string</td>
</tr>
</tbody>
</table>

DELL_CMDeviceApplication

The DELL_CMDeviceApplication class described in CMDeviceApplication Properties contains information related to the Dell change management association between the device and application.

Table 64. DELL_CMDeviceApplication Properties

| Class Name: DELL_CMDeviceApplication |
**DELL_CMInventory**

The DELL_CMInventory class described in DELL_CMInventory Properties contains information related to the Dell Change Management inventory.

**Table 65. DELL_CMInventory Properties**

- **Class Name:** DELL_CMInventory
- **Parent Class:** None

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>local</td>
<td>Indicates the locale of the system.</td>
<td>string</td>
</tr>
<tr>
<td>schemaVersion</td>
<td>Indicates the inventory schema implemented by the system.</td>
<td>string</td>
</tr>
<tr>
<td>systemID</td>
<td>Defines the system ID.</td>
<td>string</td>
</tr>
</tbody>
</table>

**DELL_CMOS**

The DELL_CMOS class described in DELL_CMOS Properties contains information related to the Dell change management operating system.

**Table 66. DELL_CMOS Properties**

- **Class Name:** DELL_CMOS
- **Parent Class:** None

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
architecture Indicates the architecture of the operating system. string

vendor Indicates the vendor of the operating system. string

majorVersion Indicates the major version of the operating system. string

minorVersion Indicates the minor version of the operating system. string

spMajorVersion Indicates the current service pack number for the operating system’s major version. string

spMinorVersion Indicates the current service pack number for the operating system’s minor version. string

DELL_CMProductInfo

The DELL_CMProductInfo class described in DELL_CMProductInfo Properties contains information related to the Dell change management product.

Table 67. DELL_CMProductInfo Properties

Class Name: DELL_CMProductInfo

Parent Class: None

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Indicates the name of the product.</td>
<td>string</td>
</tr>
<tr>
<td>description</td>
<td>Provides a short description of the product.</td>
<td>string</td>
</tr>
<tr>
<td>vendor</td>
<td>Indicates the name of the product manufacturer.</td>
<td>string</td>
</tr>
<tr>
<td>version</td>
<td>Indicates the current version number of the product.</td>
<td>string</td>
</tr>
<tr>
<td>timestamp</td>
<td>Indicate the timestamp value when the inventory information collected from the system.</td>
<td>string</td>
</tr>
</tbody>
</table>
DELL_BIOSExtensions

The DELL_BIOSExtensions identified in DELL_BIOSExtensions Properties contains information related to the specific extension of the data attributes on your system.

Table 68. DELL_BIOSExtensions Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>systemBIOSCharacteristics</td>
<td>Indicates the characteristics of BIOS on your system.</td>
<td>uint64</td>
</tr>
<tr>
<td>systemBIOSCharacteristicsExt1</td>
<td>Indicates the specific extension of the data attributes on your system.</td>
<td>uint8</td>
</tr>
<tr>
<td>systemBIOSCharacteristicsExt2</td>
<td>Indicates the specific extension of the data attributes on your system.</td>
<td>uint8</td>
</tr>
</tbody>
</table>

DELL_BIOSSettings

The DELL_BIOSSettings identified in DELL_BIOSSettings Properties contains information related to setting parameters in the Dell System Management BIOS.

Table 69. DELL_BIOSSettings Properties

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

The DELL_SDCard Devices identified in DELL_SDCardDevice Properties contains information related to the SD card devices.

Table 70. DELL SDCardDevice Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdType</td>
<td>An enumerated storage device type. The values for this property are:</td>
<td>uint8</td>
</tr>
<tr>
<td></td>
<td>1 - Other</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 - Unknown</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 - Hypervisor SD</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 - Virtual Flash SD</td>
<td></td>
</tr>
<tr>
<td>sdCertified</td>
<td>Indicates the licensing information of SD media. The values for this property are:</td>
<td>uint8</td>
</tr>
<tr>
<td></td>
<td>0 - Unknown</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 - Unlicensed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 - Licensed</td>
<td></td>
</tr>
<tr>
<td>sdCardSizeMB</td>
<td>Indicates the size of the storage device in MB.</td>
<td>uint32</td>
</tr>
<tr>
<td>sdCardFreeSizeMB</td>
<td>Indicates the available size of SD Media in MB.</td>
<td>uint32</td>
</tr>
<tr>
<td>sdCardState</td>
<td>Indicates the value of the SD Card. The values for this property are:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 - Present</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 and 2 - Reserved</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 - Offline Detected</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 - Failed Detectez</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5 - Active</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6 - Bootable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7 - Write Protected</td>
<td></td>
</tr>
</tbody>
</table>
The `DELL_Network Port` class described in `DELL_NetworkPort Properties` represents the Dell-specific features of the network adapters.

### Table 71. DELL NetworkPort Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>NIC Capabilities</td>
<td>NIC Capabilities bitmask indicates the capabilities of the NIC.</td>
<td>uint 32</td>
</tr>
<tr>
<td></td>
<td>The bitmask for the NIC Capability property are:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bit 0, Value 0 - Reporting NIC capabilities through this attribute is not supported.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bit 0, Value 1 - Reporting NIC capabilities through this attribute is supported.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bit 1, Value 0 - NIC is not TOE capable.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bit 1, Value 1 - NIC is TOE capable.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bit 2, Value 0 - NIC is not iSOE capable.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bit 2, Value 1 - NIC is iSOE capable.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bit 3, Value 0 - NIC is not FCoE capable.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bit 3, Value 1 - NIC is FCoE capable.</td>
<td></td>
</tr>
<tr>
<td>NIC TOE Capability</td>
<td>Defines the TOE capability of the NIC.</td>
<td>uint 32</td>
</tr>
<tr>
<td></td>
<td>Values for the NIC TOE Capability property are:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 - NIC/driver does not support querying for capability.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 - NIC/driver supports querying for capability but query returned an error.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 - NIC/driver supports querying for capability and querying indicates that it is capable.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 - NIC/driver supports querying for capability and querying indicates that it is not capable.</td>
<td></td>
</tr>
</tbody>
</table>
### NIC RDMA Capability

*Defined the RDMA capability of the NIC.*

**Values for the NIC RDMA Capability property are:**

- **0:** NIC/driver does not support querying for capability.
- **1:** NIC/driver supports querying for capability but query returned an error.
- **2:** NIC/driver supports querying for capability and querying indicates that it is capable.
- **4:** NIC/driver supports querying for capability and querying indicates that it is not capable.
- **8:** NIC/driver supports querying for capability but an error prevented querying the NIC/driver.
- **16:** NIC/driver supports querying for capability but NIC/driver did not respond to query.

**NOTE:** Boolean value is defined if TOE is enabled (Boolean is TOEEnable).

### NIC iSCSI Capability

*Defined the iSCSI capability of the NIC.*

**Values for the NIC iSCSI Capability property are:**

- **0:** NIC/driver does not support querying for capability.
- **1:** NIC/driver supports querying for capability but query returned an error.
- **2:** NIC/driver supports querying for capability and querying indicates that it is capable.
- **4:** NIC/driver supports querying for capability and querying indicates that it is not capable.
- **8:** NIC/driver supports querying for capability but an error prevented querying the NIC/driver.
- **16:** NIC/driver supports querying for capability but NIC/driver did not respond to query.

**NOTE:** Boolean value is defined if RDMA is enabled (Boolean is RDMAEnable).
16 - NIC/driver supports querying for capability but NIC/driver did not respond to query.

**NOTE:** Boolean value is defined if iSCSI is enabled (Boolean is iSCSIEnable).

### NIC Status
Indicates the status of the NIC or driver.

The values for the NIC Status property are:

- **0** - Unknown
- **1** - Connected
- **2** - Disconnected
- **3** - Driver is bad
- **4** - Driver is disabled
- **10** - Hardware is initializing
- **12** - Hardware is closing
- **13** - Hardware is not ready

### NParEPEnable
Indicates the mode for NParEP.

The values for the NParEPEnable are:

- **0** - Disabled
- **1** - Enabled
- **2** - Unknown

### BusNumber
Indicates the PCI bus number.

### DeviceNumber
Indicates the PCI device number.

### FunctionNumber
Indicates the PCI function number.

### DriverVersion
Indicates the NIC driver version.

### IPAddress
Indicates the NIC IP address.

### SubnetMask
Indicates the NIC subnet mask.

### DHCPServer
Indicates the DHCP server.

### DefaultGateway
Indicates the default gateway.

### CurrentMacAddress
Indicates the NIC’s current MAC address.

### OSAdapterDescription
Describes the operating system adapter.

### OSProductName
Describes the product name of the operating system.

### ServiceName
Indicates the service name.

---

**DELL_PowerConsumptionAmpsSensor**

The DELL_PowerConsumptionAmpsSensor identified in DELL_PowerConsumptionAmpsSensor contains information related to monitoring the power consumption.
### Table 72. DELL_PowerConsumptionAmpsSensor

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>UnitModifier</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>CurrentReading</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>IsLinear</td>
<td>See Common Properties of Classes</td>
<td>Boolean</td>
</tr>
<tr>
<td>LowerThresholdNonCritical</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>UpperThresholdNonCritical</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>LowerThresholdCritical</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>UpperThresholdCritical</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
</tbody>
</table>

**DELL_PowerConsumptionWattsSensor**

The **DELL_PowerConsumptionWattsSensor** identified in **DELL_PowerConsumptionWattsSensor** contains information related to monitoring the power consumption.

### Table 73. DELL PowerConsumptionWattsSensor

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>UnitModifier</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>CurrentReading</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>IsLinear</td>
<td>See Common Properties of Classes</td>
<td>Boolean</td>
</tr>
<tr>
<td>LowerThresholdNonCritical</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>UpperThresholdNonCritical</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>LowerThresholdCritical</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
<tr>
<td>UpperThresholdCritical</td>
<td>See Common Properties of Classes</td>
<td>sint32</td>
</tr>
</tbody>
</table>

**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.
Table 74. DELL PowerConsumptionData

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>cumulative PowerReading</td>
<td>Indicates the total power consumed from a start time.</td>
<td>uint 32</td>
</tr>
<tr>
<td>peakAmpReading</td>
<td>Indicates the time from which the peak amperage reading is being monitored.</td>
<td>uint 16</td>
</tr>
<tr>
<td>peakWattReading</td>
<td>Indicates the time from which the peak watt reading is being monitored.</td>
<td>uint 16</td>
</tr>
<tr>
<td>ResetCounters</td>
<td>Is the function used to reset the peak readings.</td>
<td>uint 32</td>
</tr>
<tr>
<td>powerCapSetting</td>
<td>This refers to the user configured power setting.</td>
<td>uint 16</td>
</tr>
<tr>
<td>instHeadroom</td>
<td>This refers to the instantaneous headroom.</td>
<td>uint 32</td>
</tr>
<tr>
<td>peakHeadRoom</td>
<td>Is the function used to set the power budget.</td>
<td>uint 32</td>
</tr>
</tbody>
</table>

DCIM_OEM_DataAccessModule

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

Table 75. DCIM OEM DataAccessModule

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>InstanceID</td>
<td>Identifies the instance.</td>
<td>string</td>
</tr>
<tr>
<td>GlobalStatus</td>
<td>Represents the global health status of the system. This property can have the following values:</td>
<td>sint32</td>
</tr>
<tr>
<td></td>
<td>0 - Other</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 - Unknown</td>
<td></td>
</tr>
</tbody>
</table>
# Class Name

**DCIM_OEM_DataAccessModule**

## Parent Class

**CIM_ManagedElement**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>OK</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Warning / Non-Critical</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Critical</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Non-Recoverable</td>
<td></td>
</tr>
<tr>
<td>..</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:** GlobalStatus property is available only for Linux systems.

### SendCmd

The SendCmd method is used to invoke proprietary hardware management operation.

### iDRACIPv4

Provides Remote Access controller (iDRAC ) IPv4 address.

### iDRACIPv6

Provides Remote Access controller (iDRAC ) IPv6 address.

## DCIM_RegisteredProfile

The **DCIM_RegisteredProfile** class is derived from the **CIM_RegisteredProfile** class. This class advertises the capabilities of **DCIM_OEM_DataAccessModule**.
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_PackagePSPredundancy
- DELL_PSRedundancy
- DELL_AssociatedSupplyPCWatts
- 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.

### Table 76. DELL_FanSensor Properties

**Class Name:** DELL_FanSensor  
**Parent Class:** CIM_Dependency

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antecedent</td>
<td>CIM_Tachometer refers to the tachometer (fan sensor) that measures the RPM of the fan.</td>
</tr>
<tr>
<td>Dependent</td>
<td>CIM_Fan refers to the fan whose revolutions are measured by the tachometer.</td>
</tr>
</tbody>
</table>

### 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.

### Table 77. CIM_PackageTempSensor Properties

**Class Name:** CIM_PackageTempSensor  
**Parent Class:** CIM_Dependency

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antecedent</td>
<td>CIM_TempSensor refers to the temperature sensor for the package.</td>
</tr>
<tr>
<td>Dependent</td>
<td>CIM_PhysicalPackage refers to the physical package whose environment is being monitored.</td>
</tr>
</tbody>
</table>

### 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.

**Table 78. CIM_PackageVoltage Properties**

**Class Name:** CIM_PackageVoltSensor  
**Parent Class:** CIM_Dependency

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antecedent</td>
<td>CIM_PackageVoltSensor refers to the voltage sensor for the package.</td>
</tr>
<tr>
<td>Dependent</td>
<td>CIM_PhysicalPackage refers to the physical package whose voltages are being monitored.</td>
</tr>
</tbody>
</table>

**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.

**Table 79. CIM_PackageCurrentSensor Properties**

**Class Name:** CIM_PackageCurrentSensor  
**Parent Class:** CIM_Dependency

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antecedent</td>
<td>CIM_CurrentSensor refers to the amperage sensor for the package.</td>
</tr>
<tr>
<td>Dependent</td>
<td>CIM_PhysicalPackage refers to the physical package whose amperage is being monitored.</td>
</tr>
</tbody>
</table>

**CIM_PackageFanSensor**


The CIM_PackageFanSensor class described in CIM_PackageFanSensor Properties contains fan sensors that monitor the whole package.

### Table 80. CIM_PackageFanSensor Properties

- **Class Name:** CIM_PackageFanSensor
- **Parent Class:** CIM_Dependency

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antecedent</td>
<td>CIM_Fan refers to the cooling device for the package.</td>
</tr>
<tr>
<td>Dependent</td>
<td>CIM_PhysicalPackage refers to the physical package whose environment is being monitored.</td>
</tr>
</tbody>
</table>

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.

### Table 81. CIM_PackagePowerSupplySensor Properties

- **Class Name:** CIM_PackagePowerSupplySensor
- **Parent Class:** CIM_Dependency

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antecedent</td>
<td>CIM_PowerSupplySensor refers to the power supply sensor that monitors wattage for the entire package.</td>
</tr>
<tr>
<td>Dependent</td>
<td>CIM_PhysicalPackage refers to the package whose wattage is being monitored.</td>
</tr>
</tbody>
</table>

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.
Table 82. DELL_PackagePSRedundancy Properties

Class Name: DELL_PackagePSRedundancy

Parent Class: CIM_Dependency

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antecedent</td>
<td>DELL_PSRedundancyGroup refers to power supplies that deliver wattage for the entire package.</td>
</tr>
<tr>
<td>Dependent</td>
<td>CIM_PhysicalPackage refers to the package to which the wattage is being supplied.</td>
</tr>
</tbody>
</table>

DELL_PSRedundancy

The DELL_PSRedundancy class described in DELL_PSRedundancy Properties defines what constitutes a power supply redundancy for Dell systems.

Table 83. DELL_PSRedundancy Properties

Class Name: DELL_PSRedundancy

Parent Class: CIM_Dependency

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antecedent</td>
<td>CIM_PowerSupplySensor refers to the power supply sensor that monitors wattage for the entire package.</td>
</tr>
<tr>
<td>Dependent</td>
<td>CIM_PhysicalPackage refers to the package whose wattage is being monitored.</td>
</tr>
</tbody>
</table>

DELL_AssociatedSupplyPCAmps

The DELL_AssociatedSupplyPCAmps class described in DELL_AssociatedSupplyPCAmps is a PowerConsumptionAmpsSensor associated with a CIM_PowerSupply which is defined by this class.
Table 84. DELL_AssociatedSupplyPCAmps

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antecedent</td>
<td>Indicates the PowerSupply instance.</td>
<td>uint 16</td>
</tr>
<tr>
<td>Dependent</td>
<td>Indicates the PowerConsumptionAmpsSensor associated with the CIM_PowerSupply.</td>
<td>uint 16</td>
</tr>
</tbody>
</table>

**DELL_AssociatedSystemPCWatts**

The DELL_AssociatedSystemPCWatts class described in DELL_AssociatedSystemPCWatts is a PowerConsumptionWattsSensor associated with a Dell_System which is defined by this class.

Table 85. DELL_AssociatedSystemPCWatts

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antecedent</td>
<td>Indicates the Dell_System instance.</td>
<td>uint 16</td>
</tr>
<tr>
<td>Dependent</td>
<td>Indicates the PowerConsumptionWattsSensor associated with the system.</td>
<td>uint 16</td>
</tr>
</tbody>
</table>

**AssociatedSystemPCData**

The AssociatedSystemPCData identified in AssociatedSystemPCData is a PowerConsumptionData associated with a Dell_System which is defined by this class.

Table 86. AssociatedSystemPCData

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antecedent</td>
<td>Indicates the Dell_System instance.</td>
<td>uint 16</td>
</tr>
<tr>
<td>Dependent</td>
<td>Indicates the PowerConsumptionData associated with the Power Supply.</td>
<td>uint 16</td>
</tr>
</tbody>
</table>
DELL_PowerProfileData

The DELL_PowerProfileData identified in DELL_PowerProfileData contains information related to power profiling and power knob data.

Table 87. DELL_PowerProfileData

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>chassisIndex</td>
<td>Indicates the chassisIndex for this power profile.</td>
<td>uint 8</td>
</tr>
<tr>
<td>supportedProfile</td>
<td>Indicates the supported profiles.</td>
<td>uint 16</td>
</tr>
<tr>
<td>profileSetting</td>
<td>Indicates the Profile setting.</td>
<td>uint 16</td>
</tr>
<tr>
<td>customCPUCaps</td>
<td>Indicates the Custom Profile CPU management capability.</td>
<td>uint 16</td>
</tr>
<tr>
<td>customCPUSettings</td>
<td>Indicates the Custom Profile CPU management setting.</td>
<td>uint 16</td>
</tr>
<tr>
<td>customMemCaps</td>
<td>Indicates the Custom Profile memory management capability.</td>
<td>uint 16</td>
</tr>
<tr>
<td>customMemSettings</td>
<td>Indicates the Custom Profile memory management capability.</td>
<td>uint 16</td>
</tr>
<tr>
<td>customFanSettings</td>
<td>Indicates the Custom Profile fan management setting.</td>
<td>uint 16</td>
</tr>
</tbody>
</table>