

Dell OpenManage  
Server Administrator  
Version 6.4

# CIM Reference Guide



# Notes and Cautions



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



**CAUTION:** A CAUTION indicates potential damage to hardware or loss of data instructions are not followed.

---

**Information in this publication is subject to change without notice.**

**© 2010 Dell Inc. All rights reserved.**

Reproduction of these materials in any manner whatsoever without the written permission of Dell Inc. is strictly forbidden.

Trademarks used in this text: Dell™, the DELL logo, and OpenManage™ are trademarks of Dell Inc. Microsoft® and Windows NT® are either trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries. Intel®, Pentium®, Xeon®, Itanium®, i860®, i960®, and Celeron® are registered trademarks and MMX™, i386™, i486™, SpeedStep™, and Core™ are trademarks of Intel Corporation in the United States and/or other countries. AMD™, AMD Athlon™, AMD Duron™, AMD-K5™, AMD-K6™, Opteron™, Sempron™, Phenom™ and Turion™ are trademarks and AMD-K6-2® and AMD-K6-III® are registered trademarks of Advanced Micro Devices, Inc. in the United States and/or other countries. Crusoe™ and Efficeon™ are trademarks of Transmeta Corporation in the United States and/or other countries.

Other trademarks and trade names may be used in this document to refer to either the entities claiming the marks and names or their products. Dell Inc. disclaims any proprietary interest in trademarks and trade names other than its own.

**December 2010**

# Contents

1	Introduction . . . . .	9
	<b>Server Administrator</b> . . . . .	9
	<b>What's New in this Release</b> . . . . .	10
	<b>Documenting CIM Classes and Their Properties</b> . . . . .	10
	Base Classes . . . . .	11
	Parent Classes . . . . .	12
	Classes That Describe Relationships . . . . .	12
	Dell-Defined Classes . . . . .	12
	<b>Common Properties of Classes</b> . . . . .	13
	<b>Other Documents You May Need</b> . . . . .	15
	<b>Typographical Conventions</b> . . . . .	16
2	CIM_PhysicalElement . . . . .	19
	<b>CIM_PhysicalElement</b> . . . . .	19
	<b>CIM_PhysicalPackage</b> . . . . .	21
	<b>CIM_PhysicalFrame</b> . . . . .	22
	<b>CIM_Chassis</b> . . . . .	23
	<b>DELL_Chassis</b> . . . . .	24
	<b>CIM_PhysicalComponent</b> . . . . .	26

	<b>CIM_Chip</b> . . . . .	<b>26</b>
	<b>CIM_PhysicalMemory</b> . . . . .	<b>28</b>
	<b>CIM_PhysicalConnector</b> . . . . .	<b>30</b>
	<b>CIM_Slot</b> . . . . .	<b>33</b>
<b>3</b>	<b>CIM_LogicalElement</b> . . . . .	<b>37</b>
	<b>CIM_LogicalElement</b> . . . . .	<b>38</b>
	<b>CIM_System</b> . . . . .	<b>39</b>
	<b>CIM_ComputerSystem</b> . . . . .	<b>40</b>
	<b>DELL_System</b> . . . . .	<b>40</b>
	<b>CIM_LogicalDevice</b> . . . . .	<b>41</b>
	<b>CIM_FRU</b> . . . . .	<b>42</b>
	<b>CIM_Sensor</b> . . . . .	<b>43</b>
	<b>CIM_DiscreteSensor</b> . . . . .	<b>44</b>
	<b>CIM_NumericSensor</b> . . . . .	<b>45</b>
	<b>CIM_TemperatureSensor</b> . . . . .	<b>48</b>
	<b>CIM_CurrentSensor</b> . . . . .	<b>49</b>
	<b>CIM_VoltageSensor</b> . . . . .	<b>50</b>
	<b>CIM_Tachometer</b> . . . . .	<b>51</b>
	<b>CIM_WatchDog</b> . . . . .	<b>51</b>
	<b>CIM_CoolingDevice</b> . . . . .	<b>53</b>
	<b>CIM_Fan</b> . . . . .	<b>53</b>

<b>CIM_UserDevice</b> . . . . .	<b>54</b>
<b>CIM_PointingDevice</b> . . . . .	<b>55</b>
<b>CIM_Keyboard</b> . . . . .	<b>56</b>
<b>CIM_PowerSupply</b> . . . . .	<b>57</b>
<b>CIM_Controller</b> . . . . .	<b>60</b>
<b>CIM_ParallelController</b> . . . . .	<b>61</b>
<b>CIM_SerialController</b> . . . . .	<b>62</b>
<b>CIM_PCIController</b> . . . . .	<b>63</b>
<b>CIM_PCIDevice</b> . . . . .	<b>64</b>
<b>CIM_PCIBridge</b> . . . . .	<b>65</b>
<b>CIM_Processor</b> . . . . .	<b>66</b>
<b>CIM_StorageExtent</b> . . . . .	<b>75</b>
<b>CIM_Memory</b> . . . . .	<b>76</b>
<b>CIM_CacheMemory</b> . . . . .	<b>76</b>
<b>CIM_SoftwareElement</b> . . . . .	<b>78</b>
<b>DELL_SoftwareFeature</b> . . . . .	<b>82</b>
<b>CIM_SystemResource</b> . . . . .	<b>83</b>
<b>CIM_IRQ</b> . . . . .	<b>83</b>
<b>CIM_MemoryMappedIO</b> . . . . .	<b>85</b>
<b>CIM_DMA</b> . . . . .	<b>86</b>
<b>CIM_RedundancyGroup</b> . . . . .	<b>87</b>

	<b>CIM_ExtraCapacityGroup</b> . . . . .	<b>88</b>
	<b>DELL_PSRedundancyGroup</b> . . . . .	<b>89</b>
	<b>DELL_FanRedundancyGroup</b> . . . . .	<b>90</b>
	<b>CIM_EnabledLogicalElement Group</b> . . . . .	<b>90</b>
	<b>CIM_ServiceAccessPoint</b> . . . . .	<b>91</b>
	<b>CIM_RemoteServiceAccessPoint</b> . . . . .	<b>91</b>
	<b>DELL_RemoteServiceAccessPort</b> . . . . .	<b>93</b>
<b>4</b>	<b>Dell-Defined Classes</b> . . . . .	<b>95</b>
	<b>DELL_PostLog</b> . . . . .	<b>96</b>
	<b>DELL_CMAApplication</b> . . . . .	<b>96</b>
	<b>DELL_CMDevice</b> . . . . .	<b>97</b>
	<b>DELL_CMDeviceApplication</b> . . . . .	<b>98</b>
	<b>DELL_CMInventory</b> . . . . .	<b>99</b>
	<b>DELL_CMOS</b> . . . . .	<b>99</b>
	<b>DELL_CMPProductInfo</b> . . . . .	<b>100</b>
	<b>DELL_BIOSExtensions</b> . . . . .	<b>101</b>
	<b>DELL_BIOSSettings</b> . . . . .	<b>102</b>
	<b>DELL_SDCardDevice</b> . . . . .	<b>103</b>
	<b>DELL_NetworkPort</b> . . . . .	<b>104</b>
	<b>DELL_PowerConsumptionAmpsSensor</b> . . . . .	<b>109</b>
	<b>DELL_PowerConsumptionWattsSensor</b> . . . . .	<b>109</b>

	<b>DELL_PowerConsumptionData</b> . . . . .	<b>110</b>
	<b>DCIM_OEM_DataAccessModule</b> . . . . .	<b>111</b>
	<b>DCIM_RegisteredProfile</b> . . . . .	<b>112</b>
<b>5</b>	<b>CIM_Dependency</b> . . . . .	<b>113</b>
	<b>DELL_FanSensor</b> . . . . .	<b>114</b>
	<b>CIM_PackageTempSensor</b> . . . . .	<b>114</b>
	<b>CIM_PackageVoltSensor</b> . . . . .	<b>115</b>
	<b>CIM_PackageCurrentSensor</b> . . . . .	<b>116</b>
	<b>CIM_PackageFanSensor</b> . . . . .	<b>116</b>
	<b>CIM_PackagePowerSupplySensor</b> . . . . .	<b>117</b>
	<b>DELL_PackagePSRedundancy</b> . . . . .	<b>118</b>
	<b>DELL_PSRRedundancy</b> . . . . .	<b>118</b>
	<b>DELL_AssociatedSupplyPCAmps</b> . . . . .	<b>119</b>
	<b>DELL_AssociatedSystemPCWatts</b> . . . . .	<b>119</b>
	<b>AssociatedSystemPCData</b> . . . . .	<b>120</b>
	<b>DELL_PowerProfileData</b> . . . . .	<b>120</b>
	<b>Index</b> . . . . .	<b>123</b>





# Introduction

This reference guide documents the Dell 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](http://dmtf.org) and the Microsoft website at [microsoft.com](http://microsoft.com).

## 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.

# What's New in this Release

These are the additions to this guide:

- Added new `Dell_NetworkPort` class with properties.
- Added new enumeration values (iDRAC IP values) to `DCIM_OEM_DataAccessModule`.
- Added new enumeration values to `CIM_SoftwareElement`.
- Added new enumeration values to `Dell_Chassis`.

## Documenting CIM Classes and Their Properties

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

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

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

- `CIM_PhysicalElement`: `CreationClassName`, `Tag`
- `CIM_System`: `CreationClassName`, `Name`
- `CIM_LogicalDevice`: `SystemCreationClassName`, `SystemName`, `CreationClassName`, `DeviceID`
- `CIM_Dependency`: `Antecedent`, `Dependent`
- `CIM_SoftwareElement`: `Name`, `Version`, `SoftwareElementState`, `SoftwareElementID`, `TargetOperatingSystem`
- `CIM_SoftwareFeature`: `IdentifyingNumber`, `ProductName`, `Vendor`, `Version`, `Name`

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

## Base Classes

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

- CIM\_ManagedSystemElement
- CIM\_Dependency
- DELL\_EsmLog
- DELL\_PostLog
- 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 Table 1-1.

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

## Parent Classes

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


## Classes That Describe Relationships

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

Antecedent	<code>CIM_PackageCurrentSensor</code>
Dependent	<code>CIM_PhysicalPackage</code>

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

## Dell-Defined Classes

Server Administrator has extended some CIM classes and has created new classes to assist in managing systems and their components. In the diagrams that appear in the documentation for each class, those classes created and populated by Dell are designated by the gold (lighter gray) triangle  icon.

# Common Properties of Classes

Many classes have properties such as `Caption`, `Description`, and `CreationClassName`. Table 1-1 defines properties that have the same meaning in every class that has this property and are defined more than once in this guide.

**Table 1-1. Common Properties of Classes**

<b>Property</b>	<b>Description</b>	<b>Data Type</b>
<code>Caption</code>	Describes the object using a short textual description (one-line string).	string
<code>CreationClassName</code>	Indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.	string
<code>CSCreationClassName</code>	Indicates the computer system's creation class name.	string
<code>CSName</code>	Indicates the computer system's name.	string
<code>CurrentReading</code>	Indicates the actual current value indicated by the sensor in amperes.	sint32
<code>Description</code>	Provides a textual description of the object.	string
<code>LowerThresholdNonCritical</code>	If current reading is between lower threshold noncritical and upper threshold noncritical, the current state is normal. See Figure 3-2.	sint32
<code>LowerThresholdCritical</code>	If the current reading is between upper threshold critical and upper threshold fatal, the current state is critical. See Figure 3-2.	sint32
<code>IsLinear</code>	Indicates that the sensor is linear over its dynamic range.	Boolean

**Table 1-1. Common Properties of Classes (continued)**

<b>Property</b>	<b>Description</b>	<b>Data Type</b>
Manufacturer	Provides the name of the organization responsible for producing the CIM_PhysicalElement or CIM_SoftwareElement. This may be the entity from whom the element is purchased, but not necessarily. Purchase information is contained in the <b>Vendor</b> property of CIM_Product.	string
Name	Defines the label by which the object is known. When subclassed, the <b>Name</b> property can be overridden to be a <b>Key</b> property.	string
Status	<p>Provides a string indicating the status of the component. Status values include:</p> <p><b>Operational Status Values:</b></p> <p><b>OK</b> indicates that the object is functioning normally.</p> <p><b>Degraded</b> means that the item is functioning, but not optimally.</p> <p><b>Stressed</b> indicates that the element is functioning, but needs attention. Examples of <b>Stressed</b> states are overloaded, overheated, and so on.</p> <p><b>Nonoperational Status Values:</b></p> <p><b>Non-recover</b> means that a nonrecoverable error has occurred.</p> <p><b>Error</b> means that an element has encountered an operational condition that is severe as compared to its normal mode of operation.</p>	string
SystemCreationClassName	Indicates the system's creation class name.	string

**Table 1-1. Common Properties of Classes (continued)**

Property	Description	Data Type
UnitModifier	Provides the unit multiplier for the values returned by this sensor. All the values returned by this sensor are represented in units of 10 raised to the power of the unit modifier. If the unit modifier is -6, then the units of the values returned are microvolts. The units apply to all numeric properties of the sensor, unless explicitly overridden by the units' qualifier.	sint32
UpperThresholdCritical	If the current reading is between upper threshold critical and upper threshold fatal, the current status is critical. See Figure 3-2.	sint32
UpperThresholdNonCritical	If the current reading is between lower threshold noncritical and lower threshold critical, the current status is noncritical. See Figure 3-2.	sint32
Version	Version should be in the form <major>.<minor>.<revision> or <major>.<minor><letter><revision>; for example, 1.2.3 or 1.2a3.	string

## Other Documents You May Need

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

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

- *Dell OpenManage Server Administrator Command Line Interface User's Guide* explains how to perform tasks using the text-based command line interface.
- *Dell 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 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. Table 1-2 shows a partial class description for the DELL\_DMA class. (For a full class description, see Table 3-38)

**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.



**Table 1-2. CIM\_DMA Properties**

---

<b>Class Name:</b>	CIM_DMA	
<b>Parent Class:</b>	CIM_SystemResource	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
DMAChannel	A part of the object's key value, the DMA channel number.	uint32
Availability	Availability of the DMA. <b>Availability</b> values are defined as follows: 1 - Other 2 - Unknown 3 - Available 4 - In Use/Not Available 5 - In Use and Available/Shareable	uint16

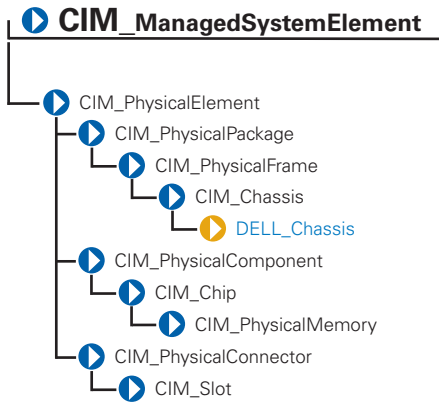
---



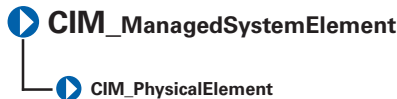
# CIM\_PhysicalElement

CIM\_PhysicalElement is a CIM-defined class. The CIM\_PhysicalElement class contains the subclasses shown in Figure 2-1.

**Figure 2-1. CIM\_PhysicalElement Class Structure**



## CIM\_PhysicalElement



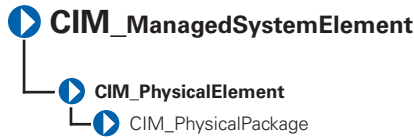
Subclasses of the CIM\_PhysicalElement class listed in Table 2-1 define any component of a system that has a distinct physical identity. Physical elements are tangible managed system elements (usually actual hardware items) that have a physical manifestation of some sort. By contrast, processes, files, 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 2-1. CIM\_PhysicalElement Properties**

<b>Class Name:</b>	CIM_PhysicalElement	
<b>Parent Class:</b>	CIM_ManagedSystemElement	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
CreationClassName	See Table 1-1.	
Manufacturer	See Table 1-1.	
Model	The name by which the physical element is generally known.	string
SerialNumber	A manufacturer-allocated number used to identify the physical element.	string
Tag	Uniquely identifies the physical element and serves as the element's key. The <b>Tag</b> property can contain information such as asset tag or serial number data. The key for physical element is placed very high in the object hierarchy in order to identify the hardware/entity independently, regardless of physical placement in or on cabinets, adapters, and so on. For example, a hot-swappable or removable component can be taken from its containing (scoping) package and temporarily unused. The object still continues to exist and may even be inserted into a different scoping container. Therefore, the key for physical element is an arbitrary string that is defined independently of any placement or location-oriented hierarchy.	string

# CIM\_PhysicalPackage

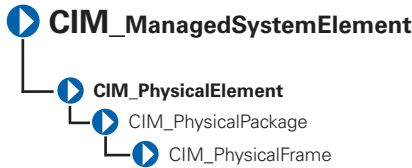


The `CIM_PhysicalPackage` class listed in Table 2-2 represents physical elements that contain or host other components. Examples are a rack enclosure or an adapter card with multiple functions.

**Table 2-2. CIM\_PhysicalPackage Properties**

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

# CIM\_PhysicalFrame

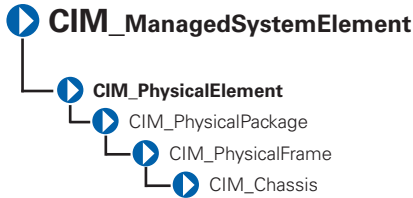


The `CIM_PhysicalFrame` class described in Table 2-3 contains other frame enclosures such as racks and chassis. Properties like **VisibleAlarm** or **AudibleAlarm**, and data related to security breaches are also members of this class.

**Table 2-3. CIM\_Physical Frame Properties**

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

# CIM\_Chassis

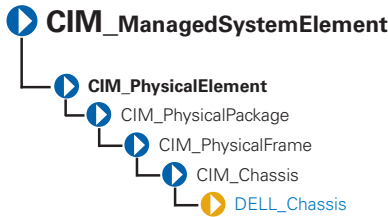


The `CIM_Chassis` class described in Table 2-4 represents the physical elements that enclose physical elements such as power supplies, fans, and processors.

**Table 2-4. CIM\_Chassis Parent Properties**

<b>Class Name:</b>	<code>CIM_Chassis</code>	
<b>Parent Class:</b>	<code>CIM_PhysicalFrame</code>	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
<code>ChassisTypes</code>	Values for the <code>ChassisTypes</code> property are: 1 - Other 2 - Unknown 3 - Mini-tower 4 - Tower 5 - Space-saving 6 - Main system chassis 7 - Expansion chassis 8 - Subchassis 9 - Space-saving 10 - Main system chassis 11 - Expansion chassis 12 - Subchassis 13 - Bus expansion chassis 14 - Peripheral chassis 15 - Storage chassis 16 - Rack-mount chassis	<code>uint16</code>

# DELL\_Chassis



The `DELL_Chassis` class explained in Table 2-5 defines the identifying and status properties of the chassis. `DELL_Chassis` inherits from CIM-defined classes, but is populated by Dell properties.

**Table 2-5. DELL\_Chassis Properties**

<b>Class Name:</b>	<code>DELL_Chassis</code>	
<b>Parent Class:</b>	<code>CIM_Chassis</code>	
Property	Description	Data Type
<code>AssetTag</code>	Indicates the container AssetTag string. This asset tag string is written by the system administrator.	string
<code>SystemClass</code>	Refers to the system type that is installed and running the instrumentation. Values for the <code>SystemClass</code> property are: 1 - Other 2 - Unknown 3 - Workstation 4 - Server 5 - Desktop 6 - Portable 7 - Net PC	uint16
<code>SystemID</code>	Indicates the system identifier code.	uint16



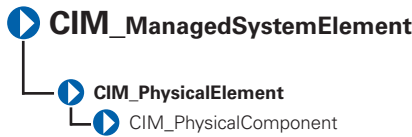
**Table 2-5. DELL\_Chassis Properties (continued)**

<b>Class Name:</b>	DELL_Chassis	
<b>Parent Class:</b>	CIM_Chassis	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
LogFormat	Defines whether the event log data is unicode formatted or binary (raw). Values for the event <b>LogFormat</b> property are: 1 - Formatted (event log only) 2 - Unformatted 3 - Events_and_POST_Formatted (both the event log and the power-on self-test (POST) log are unicode for matted)	uint16
FanStatus	Indicates the global status of fan sensors.	string
TempStatus	Indicates the global status of temperature sensors.	string
VoltStatus	Indicates the global status of voltage sensors.	string
AmpStatus	Indicates the global status of current sensors.	string
PsStatus	Indicates the global status of power supplies.	string
MemStatus	Indicates the global status of memory devices.	string
ProcStatus	Indicates the global status of processor devices.	string
FanRedStatus	Indicates the global status of the cooling unit.	string
PsRedStatus	Indicates the global status of the power unit.	string
IsDefaultThrSupported	Indicates whether resetting default thresholds are supported.	Boolean
ChassisSystemProperties	Indicates chassis characteristics, such as energy smart and so on.	uint16
ChassisSystemRevision	Indicates the chassis revision.	uint16
EsmLogStatus	Indicates the global status of ESM log.	string
MemoryRedStatus	Indicates the global status of memory redundancy.	string

**Table 2-5. DELL\_Chassis Properties (continued)**

<b>Class Name:</b>	DELL_Chassis	
<b>Parent Class:</b>	CIM_Chassis	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
ChassisExpressServiceCode	Indicates the chassis express service code	string

## CIM\_PhysicalComponent

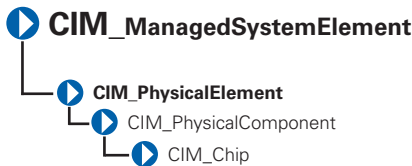


The CIM\_PhysicalComponent class listed in Table 2-6 represents any low-level or basic component within a package. A component object either cannot or does not need to be broken down into its constituent parts. For example, an application specific integrated circuit (ASIC) cannot be broken down into smaller discrete parts.

**Table 2-6. CIM\_PhysicalComponent Properties**

<b>Class Name:</b>	CIM_PhysicalComponent
<b>Parent Class:</b>	CIM_PhysicalElement

## CIM\_Chip

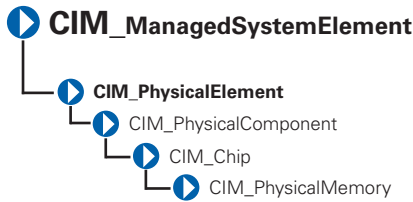


The CIM\_Chip class listed in Table 2-7 represents any type of integrated circuit hardware, including ASICs, processors, memory chips, and so on.

**Table 2-7. CIM\_Chip Properties**

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

# CIM\_PhysicalMemory



The `CIM_PhysicalMemory` class described in Table 2-8 is a subclass of `CIM_Chip`, representing low-level memory devices, such as SIMMS, DIMMs, and so on.

**Table 2-8. CIM\_PhysicalMemory Properties**

<b>Class Name:</b>	<code>CIM_PhysicalMemory</code>	
<b>Parent Class:</b>	<code>CIM_Chip</code>	
Property	Description	Data Type
<code>FormFactor</code>	See Table 2-7.	<code>uint16</code>
<code>MemoryType</code>	Indicates the type of physical memory. Values for the <code>MemoryType</code> property are: 0 - Unknown 1 - Other 2 - DRAM 3 - Synchronous DRAM 4 - Cache DRAM 5 - EDO 6 - EDRAM 7 - VRAM 8 - SRAM 9 - RAM 10 - ROM	<code>uint16</code>

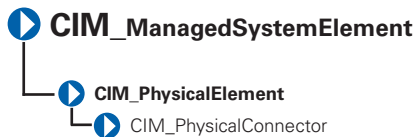
**Table 2-8. CIM\_PhysicalMemory Properties (continued)**

<b>Class Name:</b>	CIM_PhysicalMemory	
<b>Parent Class:</b>	CIM_Chip	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
MemoryType (continued)	11 - Flash 12 - EEPROM 13 - FEPRM 14 - EPROM 15 - CDRAM 16 - 3DRAM 17 - SDRAM 18 - SGRAM 19 - RDRAM 20 - DDR 21 - DDR2 22 - DDR2 FB-DIMM 24 - DDR3 25 - FBD2	
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 <b>DataWidth</b> property.	uint16
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.	uint16
Speed	Indicates the speed of the physical memory, in nanoseconds.	uint32
SpeedAsString	Indicates the accurate speed of the physical memory, in string format (with units).	string
Capacity	Indicates the total capacity of this physical memory, in bytes.	uint64

**Table 2-8. CIM\_PhysicalMemory Properties (continued)**

<b>Class Name:</b>	CIM_PhysicalMemory	
<b>Parent Class:</b>	CIM_Chip	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
BankLabel	A string identifying the physically labeled bank where the memory is located, for example, "Bank 0" or "Bank A."	string
PositionInRow	Specifies the position of the physical memory in a "row." For example, if it takes two 8-bit memory devices to form a 16-bit row, then a value of 2 means that this memory is the second device. 0 is an invalid value for this property.	uint32
InterleavePosition	Indicates the position of this physical memory in an interleave. 0 indicates noninterleaved. 1 indicates the first position, 2 the second position, and so on. For example, in a 2:1 interleave, a value of 1 indicates that the memory is in the "even" position.	uint32

## CIM\_PhysicalConnector



The `CIM_PhysicalConnector` class explained in Table 2-9 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 Table 2-10 for a list of valid connector type values.

**Table 2-9. CIM\_PhysicalConnector Properties**

<b>Class Name:</b>	CIM_PhysicalConnector	
<b>Parent Class:</b>	CIM_PhysicalElement	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
ConnectorPinout	A free-form string describing the pin configuration and signal usage of a physical connector.	string
ConnectorType	An array of integers defining the type of physical connector. An array is specified to allow the description of “combinations” of connector information. For example, one array entry could specify RS-232, another DB-25, and a third entry could define the connector as male. See Table 2-10 for the values of the <b>ConnectorType</b> property.	uint16

**Table 2-10. Connector Type Values**

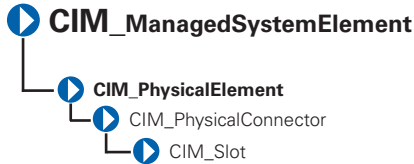
0 - Unknown	30 - <i>unused</i>	60 - Micro-DIN	90 - On Board IDE Connector
1 - Other	31 - <i>unused</i>	61 - PS/2	91 - On Board Floppy Connector
2 - Male	32 - IEEE-48	62 - Infrared	92 - 9 Pin Dual Inline
3 - Female	33 - AUI	63 - <i>unused</i>	93 - 25 Pin Dual Inline
4 - Shielded	34 - UTP Category 3	64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

**Table 2-10. Connector Type Values (continued)**

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



# CIM\_Slot



The `CIM_Slot` class described in Table 2-11 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 2-11. CIM\_Slot Properties**

<b>Class Name:</b>	class <code>CIM_Slot</code>	
<b>Parent Class:</b>	<code>CIM_PhysicalConnector</code>	
Property	Description	Data Type
ConnectorType	See Table 2-10.	uint16
SupportsHotPlug	Indicates whether the slot supports hot-plug adapter cards.	Boolean
MaxDataWidth	Indicates the maximum bus width in bits of adapter cards that can be inserted into this slot. Values for the <code>MaxDataWidth</code> property are as follows: 0 - Unknown 1 - Other 8 - Bits 16 - Bits 32 - Bits 64 - Bits 128 - Bits	uint16

**Table 2-11. CIM\_Slot Properties (continued)**

<b>Class Name:</b>	class CIM_Slot	
<b>Parent Class:</b>	CIM_PhysicalConnector	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
SystemSlotType	Indicates the type of system slot. Values for the SystemSlotType property are as follows: 1 - Other 2 - Unknown 3 - ISA 4 - MCA 5 - EISA 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	

**Table 2-11. CIM\_Slot Properties (continued)**

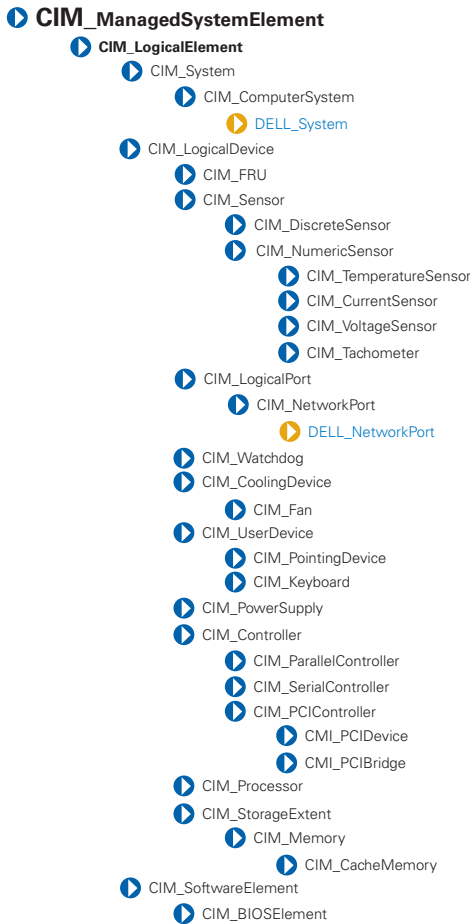
<b>Class Name:</b>	class CIM_Slot	
<b>Parent Class:</b>	CIM_PhysicalConnector	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
SystemSlotType (continued)	168 - PCI Express x4	
	169 - PCI Express x8	
	170 - PCI Express x16	
	171 - PCI Express Gen 2	
	172 - PCI Express Gen 2 x1	
	173 - PCI Express Gen 2 x2	
	174 - PCI Express Gen 2 x4	
	175 - PCI Express Gen 2 x8	
176 - PCI Express Gen 2 x16		

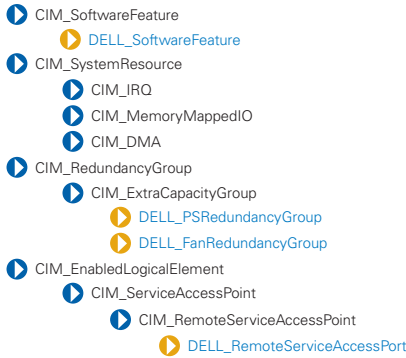


# CIM\_LogicalElement

CIM\_LogicalElement is a CIM-defined class containing the subclasses shown in Figure 3-1.

**Figure 3-1. CIM\_LogicalElement Class Structure**





## CIM\_LogicalElement

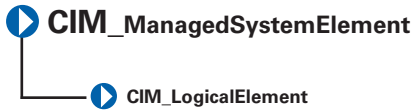


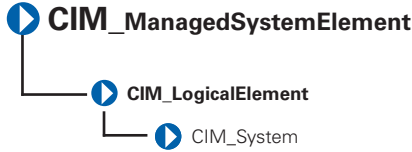
Table 3-1 lists the following characteristics for members of the `CIM_LogicalElement` class:

- 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 3-1. CIM\_LogicalElement Properties**

<b>Class Name:</b>	<code>CIM_LogicalElement</code>
<b>Parent Class:</b>	<code>CIM_ManagedSystemElement</code>

# CIM\_System

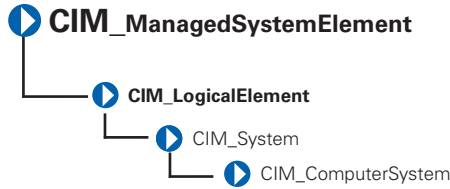


The `CIM_System` class shown in Table 3-2 defines a collection of managed system elements that operates as a functional whole. An instance of the `CIM_System` class contains a well-defined list of components that work together to perform a specific function.

**Table 3-2. CIM\_System Properties**

<b>Class Name:</b>	<code>CIM_System</code>	
<b>Parent Class:</b>	<code>CIM_LogicalElement</code>	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
<code>CreationClassName</code>	See Table 1-1.	string
<code>Name</code>	Indicates the name of a specific system, such as a particular storage system or server.	string
<code>PrimaryOwnerContact</code>	Provides information on how the primary system owner can be reached, for example, a phone number or e-mail address.	string
<code>PrimaryOwnerName</code>	Indicates the name of the primary system owner.	string
<code>Roles</code>	An array of strings that specifies the roles this system plays in the IT environment. For example, for an instance of a network system, the <b>Roles</b> property might contain the string "storage system."	string

# CIM\_ComputerSystem



The CIM\_ComputerSystem class listed in Table 3-3 contains some or all of the following CIM\_ManagedSystemElements: file system, operating system, processor, and memory (volatile and/or nonvolatile storage). For properties, see Table 3-2.

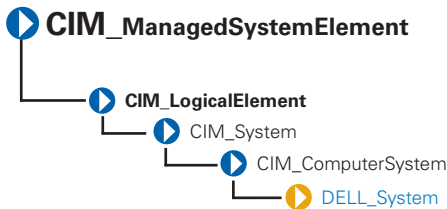
**Table 3-3. CIM\_ComputerSystem Properties**

---

<b>Class Name:</b>	CIM_ComputerSystem
<b>Parent Class:</b>	CIM_System

---

# DELL\_System



The DELL\_System class listed in Table 3-4 is the set of all Dell instrumented systems, including server, and storage systems. For properties, see Table 3-2.

**Table 3-4. DELL\_System Properties**

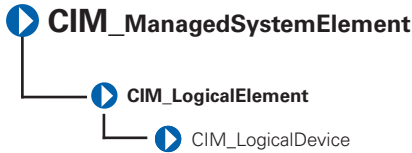
---

<b>Class Name:</b>	DELL_System
<b>Parent Class:</b>	CIM_ComputerSystem

---



# CIM\_LogicalDevice

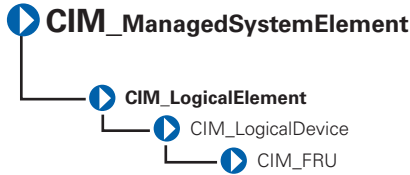


The `CIM_LogicalDevice` class described in Table 3-5 models a hardware entity that may be realized in physical hardware. `CIM_LogicalDevice` includes any characteristics of a logical device that manages its operation or configuration. An example of a logical device is a temperature sensor’s reading of actual temperature.

**Table 3-5. CIM\_Logical Device Properties**

<b>Class Name:</b>	CIM_LogicalDevice	
<b>Parent Class:</b>	CIM_LogicalElement	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
SystemCreationClassName	See Table 1-1.	string
SystemName	Indicates the scoping system’s name.	string
CreationClassName	See Table 1-1.	string
DeviceID	Identifies an address or other identifying information to uniquely name the logical device.	string

# CIM\_FRU

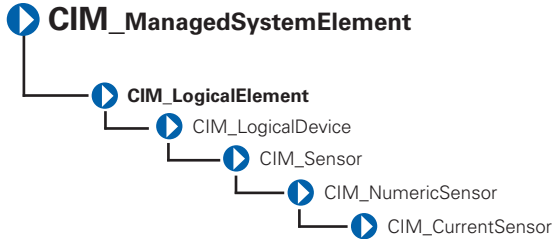


The `CIM_FRU` class described in Table 3-6 contains manufacturing information related to the Field Replaceable Units (FRU) of a system such as a system planar or I/O riser card.

**Table 3-6. CIM\_FRU Properties**

<b>Class Name:</b>	CIM_FRU	
<b>Parent Class:</b>	CIM_LogicalDevice	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
FRUInformationState	Indicates the state and availability of FRU information.	uint 16
FRUDeviceName	Indicates the device name of the FRU.	string
FRUManufacturingDateName	Indicates the manufacturing date of the FRU in ticks.	datetime
FRUManufacturerName	Indicates the name of the manufacturer.	string
FRUPartNumberName	Indicates the FRU part number.	string
FRUSerialNumberName	Indicates the FRU serial number.	string
FRURevisionName	Indicates the FRU Revision number.	string

# CIM\_Sensor



The `CIM_Sensor` class explained in Table 3-7 contains hardware devices capable of measuring the characteristics of some physical property, for example, the temperature or voltage characteristics of a computer system.

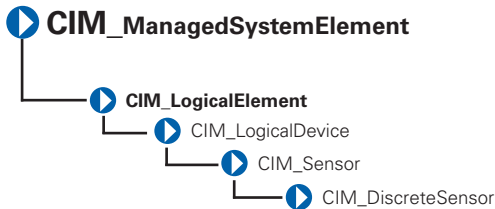
**Table 3-7. CIM\_Sensor Properties**

<b>Class Name:</b>	CIM_Sensor	
<b>Parent Class:</b>	CIM_LogicalDevice	
Property	Description	Data Type
SensorType	<p>The type of the sensor, for example, voltage or temperature sensor.</p> <p>Values for the <b>SensorType</b> property are:</p> <ul style="list-style-type: none"> <li>0 - Unknown</li> <li>1 - Other</li> <li>2 - Temperature sensors measure the environmental temperature.</li> <li>3 - Voltage sensors measure electrical voltage.</li> <li>4 - Current sensors measure current readings.</li> <li>5 - Tachometers measure speed/revolutions of a device. For example, a fan device can have an associated tachometer that measures its speed.</li> <li>6 - Batteries maintain the time and date and save the system's BIOS configuration when the system is switched off.</li> </ul>	uint16
OtherSensorType Description	The type of sensor when the SensorType property is set to <b>Other</b> .	string

**Table 3-7. CIM\_Sensor Properties (continued)**

<b>Class Name:</b>	CIM_Sensor	
<b>Parent Class:</b>	CIM_LogicalDevice	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
PossibleStates	Enumerates the string outputs of the sensor. For example, a NumericSensor can report states based on threshold readings.	string
CurrentState	Indicates the current state of the sensor. This value is always one of the Possible States.	string
PollingInterval	Indicates the polling interval, in nanoseconds, that the sensor hardware or instrumentation uses to determine the current state of the sensor.	uint64

## CIM\_DiscreteSensor

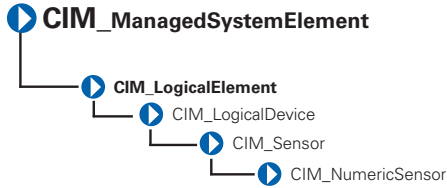


The CIM\_DiscreteSensor class described in Table 3-8 has a set of legal string values that it can report. The CIM\_DiscreteSensor always has a "current reading" that corresponds to one of the enumerated values.

**Table 3-8. CIM\_DiscreteSensor Properties**

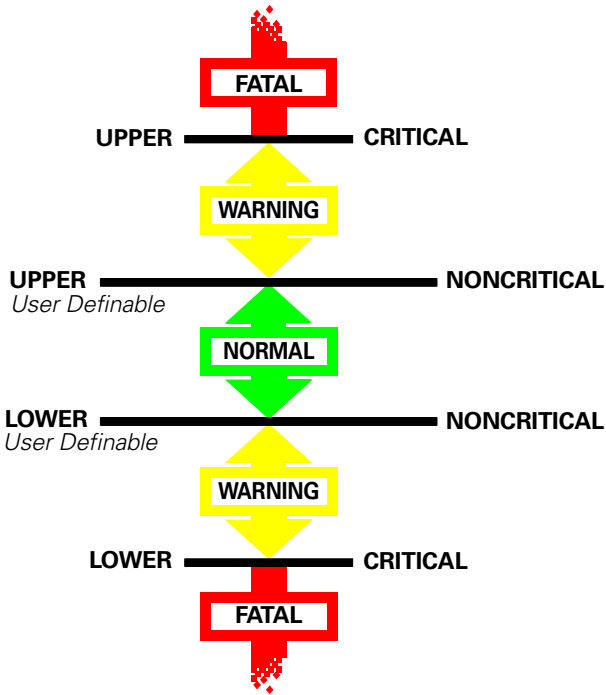
<b>Class Name:</b>	CIM_DiscreteSensor	
<b>Parent Class:</b>	CIM_Sensor	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
CurrentReading	See Table 1-1.	sint32
PossibleValues	Enumerates the string outputs that can be reported by the sensor.	sint32

# CIM\_NumericSensor



The `CIM_NumericSensor` class described in Table 3-9 returns numeric settings and may also support threshold settings. Figure 3-2 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 3-2. Ranges for Threshold Values**



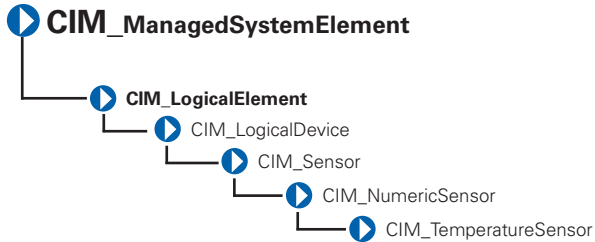
**Table 3-9. CIM\_NumericSensor Properties**

<b>Class Name:</b>	CIM_NumericSensor	
<b>Parent Class:</b>	CIM_Sensor	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
UnitModifier	See Table 1-1.	sint32
CurrentReading	See Table 1-1.	sint32
IsLinear	See Table 1-1.	Boolean
LowerThresholdNonCritical	See Table 1-1.	sint32
UpperThresholdNonCritical	See Table 1-1.	sint32
LowerThresholdCritical	See Table 1-1.	sint32

**Table 3-9. CIM\_NumericSensor Properties (continued)**

<b>Class Name:</b>	CIM_NumericSensor	
<b>Parent Class:</b>	CIM_Sensor	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
UpperThresholdCritical	See Table 1-1.	sint32
SupportedThresholds	An array representing the thresholds supported by this sensor. The supported values are as follows: <b>1 - LowerThresholdNonCritical</b> <b>2 - UpperThresholdNonCritical</b> <b>3 - LowerThresholdCritical</b> <b>4 - UpperThresholdCritical</b>	uint16
EnabledThresholds	An array representing the thresholds that are currently enabled for this sensor. Enabled threshold values are as follows: <b>1 - LowerThresholdNonCritical</b> <b>2 - UpperThresholdNonCritical</b> <b>3 - LowerThresholdCritical</b> <b>4 - UpperThresholdCritical</b>	uint16
SettableThresholds	An array representing the writable thresholds supported by sensor. Settable threshold values are: <b>1 - LowerThresholdNonCritical</b> <b>2 - UpperThresholdNonCritical</b>	uint16

# CIM\_TemperatureSensor



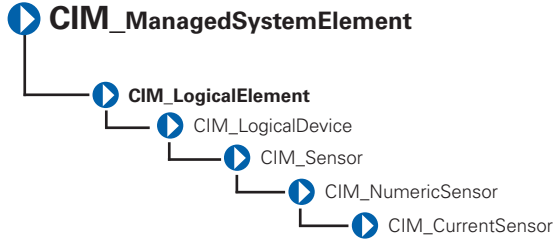
The `CIM_TemperatureSensor` class listed in Table 3-10 contains sensors that sample ambient temperature and return a value in degrees Celsius.

**Table 3-10. CIM\_TemperatureSensor Properties**

<b>Class Name:</b>	<code>CIM_TemperatureSensor</code>	
<b>Parent Class:</b>	<code>CIM_NumericSensor</code>	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
<code>UnitModifier</code>	See Table 1-1.	<code>sint32</code>
<code>CurrentReading</code>	See Table 1-1.	<code>sint32</code>
<code>IsLinear</code>	See Table 1-1.	<code>Boolean</code>
<code>LowerThresholdNonCritical</code>	See Table 1-1.	<code>sint32</code>
<code>UpperThresholdNonCritical</code>	See Table 1-1.	<code>sint32</code>
<code>LowerThresholdCritical</code>	See Table 1-1.	<code>sint32</code>
<code>UpperThresholdCritical</code>	See Table 1-1.	<code>sint32</code>



# CIM\_CurrentSensor

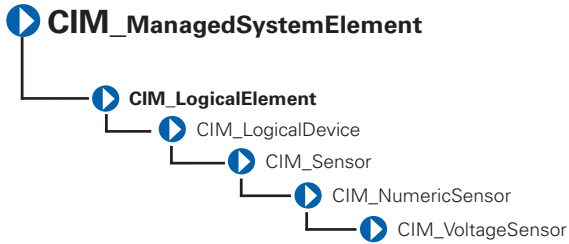


The `CIM_CurrentSensor` class listed in Table 3-11 contains sensors that measure amperage and returns a value in amperes and watts.

**Table 3-11. CIM\_CurrentSensor Properties**

<b>Class Name:</b>	<code>CIM_CurrentSensor</code>	
<b>Parent Class:</b>	<code>CIM_NumericSensor</code>	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
<code>UnitModifier</code>	See Table 1-1.	<code>sint32</code>
<code>CurrentReading</code>	See Table 1-1.	<code>sint32</code>
<code>IsLinear</code>	See Table 1-1.	<code>Boolean</code>
<code>LowerThresholdNonCritical</code>	See Table 1-1.	<code>sint32</code>
<code>UpperThresholdNonCritical</code>	See Table 1-1.	<code>sint32</code>
<code>LowerThresholdCritical</code>	See Table 1-1.	<code>sint32</code>
<code>UpperThresholdCritical</code>	See Table 1-1.	<code>sint32</code>

# CIM\_VoltageSensor

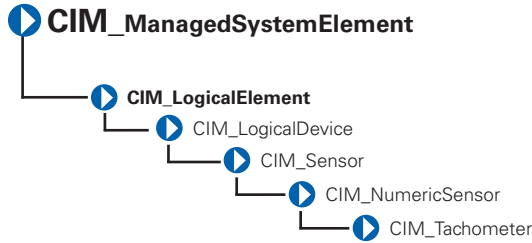


The `CIM_VoltageSensor` class shown in Table 3-12 contains sensors that measure voltage and return a value in volts.

**Table 3-12. CIM\_VoltageSensor Properties**

<b>Class Name:</b>	CIM_VoltageSensor	
<b>Parent Class:</b>	CIM_NumericSensor	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
UnitModifier	See Table 1-1.	sint32
CurrentReading	See Table 1-1.	sint32
IsLinear	See Table 1-1.	Boolean
LowerThresholdNonCritical	See Table 1-1.	sint32
UpperThresholdNonCritical	See Table 1-1.	sint32
LowerThresholdCritical	See Table 1-1.	sint32
UpperThresholdCritical	See Table 1-1.	sint32

# CIM\_Tachometer

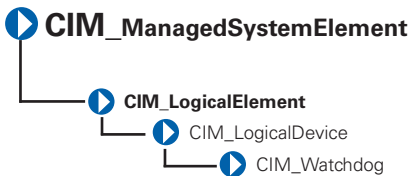


The `CIM_Tachometer` class listed in Table 3-13 contains devices that measure revolutions per minute (RPM) of a fan and return the value in RPMs.

**Table 3-13. CIM\_Tachometer Properties**

<b>Class Name:</b>	<code>CIM_Tachometer</code>	
<b>Parent Class:</b>	<code>CIM_NumericSensor</code>	
Property	Description	Data Type
<code>SensorType</code>	See Table 1-1.	<code>uint16</code>
<code>UnitModifier</code>	See Table 1-1.	<code>sint32</code>
<code>CurrentReading</code>	See Table 1-1.	<code>sint32</code>
<code>IsLinear</code>	See Table 1-1.	<code>Boolean</code>
<code>LowerThresholdNonCritical</code>	See Table 1-1.	<code>sint32</code>
<code>UpperThresholdNonCritical</code>	See Table 1-1.	<code>sint32</code>

# CIM\_WatchDog

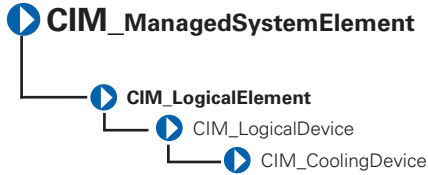


The `CIM_WatchDog` class described in Table 3-14 represents a timer that is implemented in system hardware. The watchdog feature allows the hardware to monitor the state of the operating system, BIOS, or a software component installed on the system. If the monitored component fails to rearm the timer 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 3-14. CIM\_WatchDog Properties**

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

# CIM\_CoolingDevice

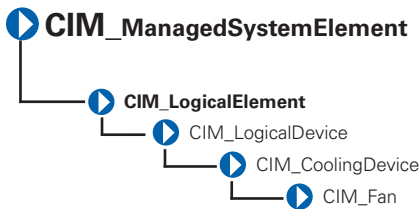


The CIM\_CoolingDevice class described in Table 3-15 contains a set of devices that work to keep the ambient internal temperature of the system at a safe value.

**Table 3-15. CIM\_CoolingDevice Properties**

<b>Class Name:</b>	CIM_CoolingDevice	
<b>Parent Class:</b>	CIM_LogicalDevice	
Property	Description	Data Type
ActiveCooling	Specifies whether the device provides active (as opposed to passive) cooling.	Boolean

# CIM\_Fan

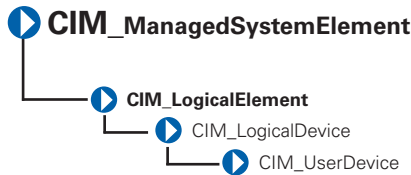


The CIM\_Fan class explained in Table 3-16 contains a set of devices that work to keep the ambient internal temperature of the system at a safe value by circulating air.

**Table 3-16. CIM\_Fan Properties**

<b>Class Name:</b>	CIM_Fan	
<b>Parent Class:</b>	CIM_CoolingDevice	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
VariableSpeed	Specifies whether the fan supports variable speeds.	Boolean
DesiredSpeed	Indicates the currently requested fan speed, defined in RPM. When the value = TRUE, the fan supports variable speeds. When a variable speed fan is supported ( <b>VariableSpeed Boolean = TRUE</b> ), the actual speed is determined using a sensor (CIM_Tachometer) that is associated with the fan.	uint64

## CIM\_UserDevice

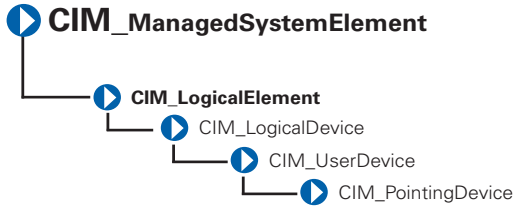


The CIM\_UserDevice class shown in Table 3-17 contains logical devices that allow a computer system’s users to input, view, or hear data. Classes derived from CIM\_UserDevice include CIM\_Keyboard and CIM\_PointingDevice.

**Table 3-17. CIM\_UserDevice Properties**

<b>Class Name:</b>	CIM_UserDevice	
<b>Parent Class:</b>	CIM_LogicalDevice	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
IsLocked	Indicates whether the device is locked, preventing user input or output.	Boolean

# CIM\_PointingDevice



The `CIM_PointingDevice` class described in Table 3-18 includes those devices used to point to regions of a display. Examples are a mouse or a trackball.

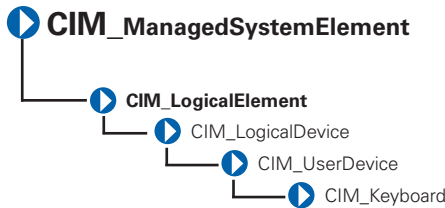
**Table 3-18. CIM\_PointingDevice Properties**

<b>Class Name:</b>	<code>CIM_PointingDevice</code>	
<b>Parent Class:</b>	<code>CIM_UserDevice</code>	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
PointingType	Indicates the type of pointing device. Values for the <b>PointingType</b> property are: 1 - Other 2 - Unknown 3 - Mouse 4 - Trackball 5 - Trackpoint 6 - Glidepoint 7 - Touch pad 8 - Touch screen 9 - Mouse—optical sensor	uint16
NumberOfButtons	Indicates the number of buttons. If the <code>CIM_PointingDevice</code> has no buttons, a value of 0 is returned.	uint8

**Table 3-18. CIM\_PointingDevice Properties (continued)**

<b>Class Name:</b>	CIM_PointingDevice	
<b>Parent Class:</b>	CIM_UserDevice	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
Handedness	Integer indicating whether the CIM_PointingDevice is configured for right- or left-handed operation. Values for the <b>Handedness</b> property are as follows:  0 - Unknown 1 - Not applicable 2 - Right-handed operation 3 - Left-handed operation	uint16

## CIM\_Keyboard



The CIM\_Keyboard class explained in Table 3-19 includes devices that allow users to enter data.

**Table 3-19. CIM\_Keyboard Properties**

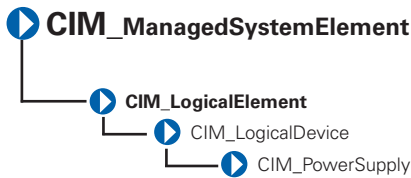
<b>Class Name:</b>	CIM_Keyboard	
<b>Parent Class:</b>	CIM_UserDevice	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
NumberOfFunctionKeys	Indicates the number of function keys on the keyboard.	uint16
Layout	A free-form string indicating the format and layout of the keyboard.	string



**Table 3-19. CIM\_Keyboard Properties (continued)**

<b>Class Name:</b>	CIM_Keyboard	
<b>Parent Class:</b>	CIM_UserDevice	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
Password	An integer indicating whether a hardware-level password is enabled at the keyboard, preventing local input. Values for the <b>Password</b> property are:  1 - Other 2 - Unknown 3 - Disabled 4 - Enabled 5 - Not implemented	uint16

## CIM\_PowerSupply



The `CIM_PowerSupply` class described in Table 3-20 contains devices that provide current and voltage for the operation of the system and its components.

**Table 3-20. CIM\_PowerSupply Properties**

<b>Class Name:</b>	CIM_PowerSupply	
<b>Parent Class:</b>	CIM_LogicalDevice	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
IsSwitchingSupply	Indicates that the power supply is a switching power supply and not a linear power supply.	Boolean

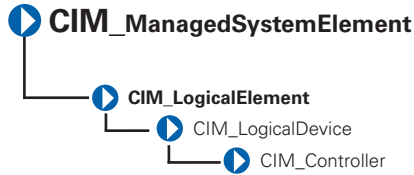
**Table 3-20. CIM\_PowerSupply Properties (continued)**

<b>Class Name:</b>	CIM_PowerSupply	
<b>Parent Class:</b>	CIM_LogicalDevice	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
Range1InputVoltageLow	Indicates the low voltage in millivolts of input voltage range 1 for this power supply. A value of 0 denotes unknown.	uint32
Range1InputVoltageHigh	Indicates the high voltage in millivolts of input voltage range 1 for this power supply. A value of 0 denotes unknown.	uint32
ActiveInputVoltage	Indicates which input voltage range is currently in use. Range 1, 2, or both can be specified using the values 3, 4, or 5, respectively. If the supply is not drawing power, a value of 6 (neither) can be specified. This information is necessary in the case of an uninterruptible power supply (UPS), a subclass of power supply. Values for the <b>ActiveInputVoltage</b> property are:  1 - Other 2 - Unknown 3 - Range 1 4 - Range 2 5 - Both range 1 and range 2 6 - Neither range 1 nor range 2	uint16

**Table 3-20. CIM\_PowerSupply Properties (continued)**

<b>Class Name:</b>	CIM_PowerSupply	
<b>Parent Class:</b>	CIM_LogicalDevice	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
ActiveInputVoltage	Indicates which input voltage range is currently in use. Range 1, 2, or both can be specified using the values 3, 4, or 5, respectively. If the supply is not drawing power, a value of 6 (neither) can be specified. This information is necessary in the case of an uninterruptible power supply (UPS), a subclass of power supply. Values for the <b>ActiveInputVoltage</b> property are: 1 - Other 2 - Unknown 3 - Range 1 4 - Range 2 5 - Both range 1 and range 2 6 - Neither range 1 nor range 2	uint16
TotalOutputPower	Represents the total output power of the power supply in milliwatts. A value of 0 denotes that the power output is unknown.	uint32
PMCapable	Indicates the Power Monitoring capability.	Boolean

# CIM\_Controller

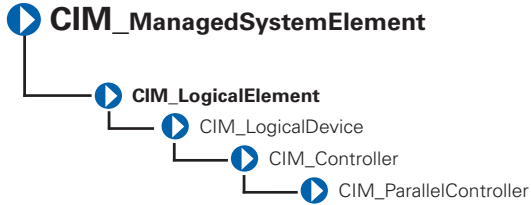


The `CIM_Controller` class shown in Table 3-21 groups miscellaneous control-related devices. Examples of controllers are small computer system interface (SCSI) controllers, Universal Serial Bus (USB) controllers, and serial controllers.

**Table 3-21. CIM\_Controller Properties**

<b>Class Name:</b>	<code>CIM_Controller</code>	
<b>Parent Class:</b>	<code>CIM_LogicalDevice</code>	
Property	Description	Data Type
<code>ProtocolSupported</code>	The protocol used by the controller to access controlled devices. Values for the <b>ProtocolSupported</b> property are: 1 - Other 2 - Unknown 3 - PCI 4 - Parallel protocol	<code>uint16</code>

# CIM\_ParallelController

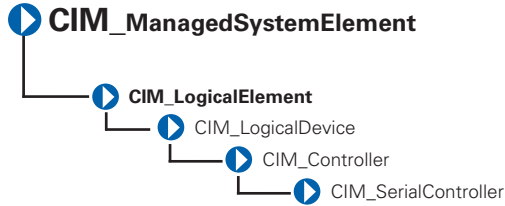


The `CIM_ParallelController` class identified in Table 3-22 contains a set of objects that control parallel devices. Parallel controllers transfer 8 or 16 bits of data at a time to the devices they control, for example, a parallel port controlling a printer.

**Table 3-22. CIM\_ParallelController Properties**

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

# CIM\_SerialController

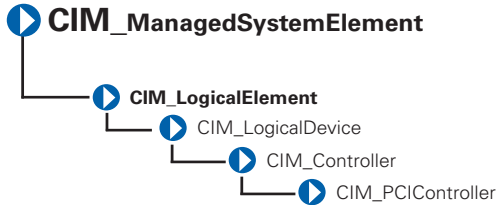


The `CIM_SerialController` class explained in Table 3-23 contains controllers that transfer data one bit at a time to the devices they control, for example, a serial port controlling a modem.

**Table 3-23. CIM\_SerialController Properties**

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

# CIM\_PCIController



The `CIM_PCIController` class listed in Table 3-24 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.

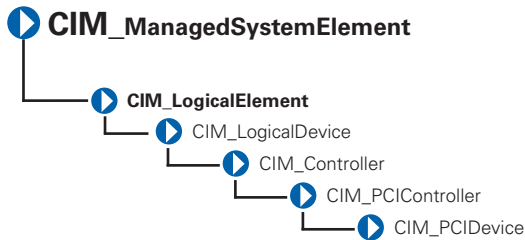
**Table 3-24. CIM\_PCIController Properties**

<b>Class Name:</b>	<code>CIM_PCIController</code>	
<b>Parent Class:</b>	<code>CIM_Controller</code>	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
<code>CommandRegister</code>	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.	<code>uint16</code>

**Table 3-24. CIM\_PCIController Properties (continued)**

<b>Class Name:</b>	CIM_PCIController	
<b>Parent Class:</b>	CIM_Controller	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
CommandRegister (continued)	Values for the <b>CommandRegister</b> property are: 0 - Unknown 1 - Other 2 - Supports 66 MHz 3 - Supports user-definable features 4 - Supports fast back-to-back transactions 5 - PCI-X capable 6 - PCI power management supported 7 - Message signaled interrupts supported 8 - Parity error recovery capable 9 - AGP supported 10 - Vital product data supported 11 - Provides slot identification 12 - Hot swap supported	

## CIM\_PCIDevice



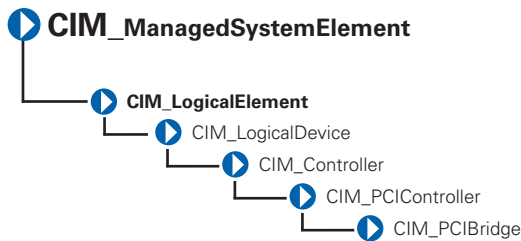
The `CIM_PCIDevice` class shown in Table 3-25 describes the capabilities and management of a PCI device controller on an adapter card.



**Table 3-25. CIM\_PCIDevice Properties**

<b>Class Name:</b>	CIM_PCIDevice	
<b>Parent Class:</b>	CIM_PCIController	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
BaseAddress	Identifies an array of up to six double-word base memory addresses.	uint32
SubsystemID	Identifies a subsystem identifier code.	uint16
SubsystemVendorID	Identifies a subsystem vendor ID. ID information is reported from a PCI device via protocol-specific requests. This information is also present in the CIM_PhysicalElement class (the manufacturer property) for hardware, and the CIM_Product class (the vendor property) for information related to product acquisition.	uint16
ExpansionROMBaseAddress	Identifies a double-word expansion ROM base memory address.	uint32

## CIM\_PCIBridge

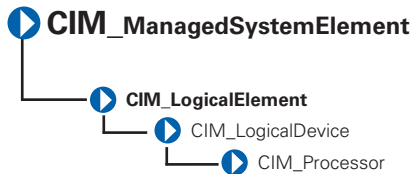


The CIM\_PCIBridge class shown in Table 3-26 describes the capabilities and management of a PCI controller providing bridge-to-bridge capability. An example is a PCI to Industry-Standard Architecture (ISA) bus bridge.

**Table 3-26. CIM\_PCIBridge Properties**

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

## CIM\_Processor



The `CIM_Processor` class described in Table 3-27 contains devices that interpret and execute commands, for example, the Intel Xeon microprocessor.

**Table 3-27. CIM\_Processor Properties**

<b>Class Name:</b>	CIM_Processor	
<b>Parent Class:</b>	CIM_LogicalDevice	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
Role	A string describing the role of the microprocessor, for example, central microprocessor or math processor.	string
UpgradeMethod	Provides microprocessor socket information including data on how this microprocessor can be upgraded (if upgrades are supported). This property is an integer enumeration. Values for the <b>UpgradeMethod</b> property are as follows: 1 - Other 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	uint16
MaxClockSpeed	Indicates the maximum speed (in MHz) of this microprocessor.	uint32
Core count	Indicates the number of core processors detected.	uint16
CoreEnabledCount	Indicates the number of core processors enabled.	uint16

**Table 3-27. CIM\_Processor Properties (continued)**

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

**Table 3-27. CIM\_Processor Properties (continued)**

<b>Class Name:</b>	CIM_Processor	
<b>Parent Class:</b>	CIM_LogicalDevice	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
	the BIOS or disabled due to a POST error. Values for the <b>CPUStatus</b> property are as follows: 0 - Unknown 1 - Microprocessor enabled 2 - Microprocessor disabled by user via 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 <b>Family</b> property are as follows: 1 - Other 2 - Unknown 3 - 8086 4 - 80286 5 - 80386 6 - 80486 7 - 8087 8 - 80287 9 - 80387 10 - 80487 11 - Pentium Brand 12 - Pentium Pro 13 - Pentium II 14 - Pentium processor with MMX technology 15 - Celeron 16 - Pentium II Xeon 17 - Pentium III	uint16

**Table 3-27. CIM\_Processor Properties (continued)**

<b>Class Name:</b>	CIM_Processor	
<b>Parent Class:</b>	CIM_LogicalDevice	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
Family (continued)	18 - M1 family	
	19 - M2 family	
	24 - AMD Duron Processor	
	25 - K5 family	
	26 - K6 family	
	27 - K6 -2	
	28 - K6-3	
	29 - AMD Athlon Processor Family	
	30 - AMD29000 Family	
	31 - K6-2+	
	32 - Power PC Family	
	33 - Power PC 601	
	34 - Power PC 603	
	35 - Power PC 603 +	
	36 - Power PC 604	
	37 - Power PC 620	
	38 - Power PC X704	
	39 - Power PC 750	
	40 - Intel Core Duo processor	
	41 - Intel Core Duo mobile processor	
42 - Intel Core Solo mobile processor		
43 - Intel Atom processor		
48 - Alpha Family		
49 - Alpha 21064		
50 - Alpha 21066		
51 - Alpha 21164		
52 - Alpha 21164PC		
53 - Alpha 21164a		

**Table 3-27. CIM\_Processor Properties (continued)**

<b>Class Name:</b>	CIM_Processor	
<b>Parent Class:</b>	CIM_LogicalDevice	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
Family (continued)	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		

**Table 3-27. CIM\_Processor Properties (continued)**

<b>Class Name:</b>	CIM_Processor	
<b>Parent Class:</b>	CIM_LogicalDevice	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
Family (continued)	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		



**Table 3-27. CIM\_Processor Properties (continued)**

<b>Class Name:</b>	CIM_Processor	
<b>Parent Class:</b>	CIM_LogicalDevice	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
Family (continued)	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	
	178 - Technology	
	179 - Pentium 4	
	180 - Intel Xeon	
	181 - AS400 Family	
	182 - Intel Xeon Processor MP	
	183 - AMD Athlon XP family	
	184 - AMD Athlon MP family	
	185 - Intel Itanium 2	
	186 - Intel Pentium M processor	
	187 - Intel Celeron D Processor	
	188 - Intel Pentium D Processor	
	189 - Intel Pentium Extreme Edition processor	
	190 - Intel Core 2 processor	
	192 - Intel Core 2 Solo processor	
	193 - Intel Core 2 Extreme processor	
	194 - Intel Core 2 Quad processor	

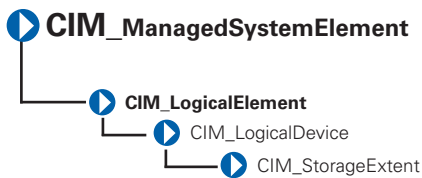
**Table 3-27. CIM\_Processor Properties (continued)**

<b>Class Name:</b>	CIM_Processor	
<b>Parent Class:</b>	CIM_LogicalDevice	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
Family (continued)	195 - Intel Core 2 Extreme mobile processor 196 - Intel Core 2 Duo mobile processor 197 - Intel Core 2 Solo mobile processor 198 - Intel Core i7 Processor 199 - Dual-Core Intel Celeron Processor 200 - S/390 and zSeries family 201 - ESA/390 G4 202 - ESA/390 G5 203 - ESA/390 G6 204 - z/Architecture base 206 - CEh 206 Intel Core i3 processor 214 - Multi-Core Intel Xeon processor 215 - Dual-Core Intel Xeon processor 3xxx Series 216 - Quad-Core Intel Xeon processor 3xxx Series 217 - D9h 217 VIA Nano Processor Family 218 - Dual-Core Intel Xeon processor 5xxx Series 219 - Quad-Core Intel Xeon processor 5xxx Series 221 - Dual-Core Intel Xeon processor 7xxx Series 222 - Quad-Core Intel Xeon processor 7xxx Series 223 - Multi-Core Intel Xeon processor 7xxx Series 224 - E0h 224 Multi-Core Intel Xeon processor 3400 Series 230 - Embedded AMD Opteron Quad-Core Processor Family 231 - AMD Phenom Triple-Core Processor Family 232 - AMD Turion Ultra Dual-Core Mobile Processor Family 233 - AMD Turion Dual-Core Mobile Processor Family 234 - AMD Athlon Dual-Core Processor Family 235 - AMD Sempron SI Processor Family	

**Table 3-27. CIM\_Processor Properties (continued)**

<b>Class Name:</b>	CIM_Processor	
<b>Parent Class:</b>	CIM_LogicalDevice	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
Family (continued)	238 - AMD Opteron Six-Core Processor Family	
	250 - i860	
	251 - i960	
	260 - SH-3	
	261 - SH-4	
	280 - ARM	
	281 - StrongARM	
	300 - 6x86	
	301 - MediaGX	
	302 - MII	
	320 - WinChip	
	350 - DSP	
	500 - Video processor	

## CIM\_StorageExtent

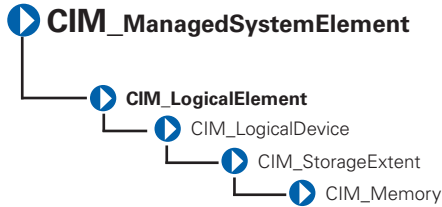


CIM\_StorageExtent identified in Table 3-28 contains devices that manage data storage, for example, hard drives or microprocessor memory.

**Table 3-28. CIM\_StorageExtent Properties**

<b>Class Name:</b>	CIM_StorageExtent
<b>Parent Class:</b>	CIM_LogicalDevice

# CIM\_Memory



The CIM\_Memory class identified in Table 3-29 describes the capabilities and management of storage extent devices, for example, cache memory or system memory.

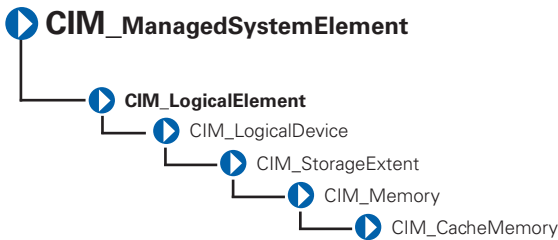
**Table 3-29. CIM\_Memory Properties**

---

<b>Class Name:</b>	CIM_Memory
<b>Parent Class:</b>	CIM_StorageExtent

---

# CIM\_CacheMemory



The CIM\_CacheMemory class explained in Table 3-30 describes the capabilities and management of cache memory. Cache memory allows a microprocessor to access data and instructions faster than normal system memory.

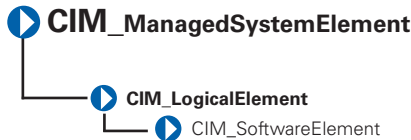
**Table 3-30. CIM\_CacheMemory Properties**

<b>Class Name:</b> CIM_CacheMemory		
<b>Parent Class:</b> CIM_Memory		
Property	Description	Data Type
Level	Defines whether this is the primary, secondary, or tertiary cache. Values for the Level property are as follows: 1 - Other 2 - Unknown 3 - Primary 4 - Secondary 5 - Tertiary 6 - Not applicable	uint16
WritePolicy	Either defines whether this cache is a write-back or write-through cache or whether this information varies with address or is defined individually for each input/output (I/O). Values for the <b>WritePolicy</b> property are as follows: 1 - Other 2 - Unknown 3 - Write-back 4 - Write-through 5 - Varies with address 6 - Determination per I/O	uint16
CacheType	Defines whether this cache is for instruction caching, data caching, or both (unified). Values for the <b>CacheType</b> property are as follows: 1 - Other 2 - Unknown 3 - Instruction 4 - Data 5 - Unified	uint16

**Table 3-30. CIM\_CacheMemory Properties (continued)**

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

## CIM\_SoftwareElement



The `CIM_SoftwareElement` class described in Table 3-31 is used to define `CIM_SoftwareFeature`. The `CIM_SoftwareElement` class consists of individually manageable or deployable parts for a particular platform. A software element’s platform is uniquely identified by its underlying hardware architecture and operating system (for example, a system running Microsoft Windows NT on an Intel microprocessor). A software element’s implementation on a particular platform depends on the platform’s operating system.

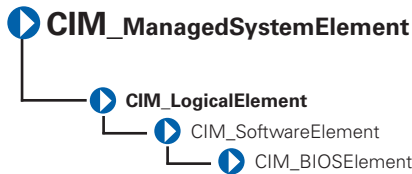
**Table 3-31. CIM\_SoftwareElement Properties**

<b>Class Name:</b>	CIM_SoftwareElement	
<b>Parent Class:</b>	CIM_LogicalElement	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
Name	Indicates the name that identifies this software element.	string
Version	Provides the version in the form <major>.<minor>.<revision> or <major>.<minor><letter><revision>; for example, 1.2.3 or 1.2a3.	string
Manufacturer	See Table 1-1.	string
BuildNumber	Indicates the internal identifier for this build of the software element.	string
IdentificationCode	Provides the manufacturer's identifier for this software element. Often this is a stock keeping unit (SKU) or a part number.	string
SoftwareElementType	Indicates the type of software element. Values for SoftwareElementType are: 1 - Other 2 - Unknown 3 - BIOS 4 - ESM 5 - PSPB 6 - System Backplane 7 - Hendrix (PV20x) Kernel 8 - Hendrix (PV20x) Application 9 - Front Panel 10 - BMC 11 - Hot Plug PCI 12 - SDR 13 - Peripheral Bay Backplane	uint16

**Table 3-31. CIM\_SoftwareElement Properties (continued)**

<b>Class Name:</b>	CIM_SoftwareElement	
<b>Parent Class:</b>	CIM_LogicalElement	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
SoftwareElementType	14 - Slimfast Secondary Backplane	uint16
(continued)	15 - Generic Secondary Backplane (ESM 3&4)	
	16 - RAC4	
	17 - iDRAC	
	18 - iDRAC6	
	19 Lifecycle Controller	
	20 Unified Server Configurator	

## CIM\_BIOSElement



The CIM\_BIOSElement class listed in Table 3-32 describes the BIOS for the system. The BIOS controls the following:

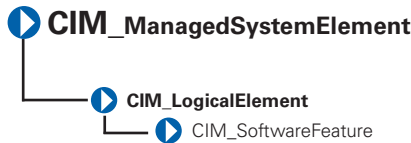
- Communications between the microprocessor and peripheral devices, such as the keyboard and the video adapter.
- Miscellaneous functions, such as system messages.
- Miscellaneous functions, such as system messages.



**Table 3-32. CIM\_BIOSElement Properties**

<b>Class Name:</b>	CIM_BIOSElement	
<b>Parent Class:</b>	CIM_SoftwareElement	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
Version	Provides the product version information.	string
Manufacturer	See Table 1-1	string
PrimaryBIOS	Specifies whether a given BIOS is the primary BIOS for the system. When the value = TRUE, the BIOS is the primary BIOS.	Boolean

## CIM\_SoftwareFeature

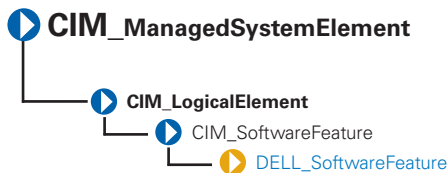


The `CIM_SoftwareFeature` class shown in Table 3-33 defines a particular function or capability of a product or application system. This class is intended to be meaningful to a consumer, or user of a product, rather than to explain how the product is built or packaged. When a software feature can exist on multiple platforms or operating systems (for example, a client component of a three-tiered client/server application might run on Windows NT), 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 3-33. CIM\_SoftwareFeature Properties**

<b>Class Name:</b>	CIM_SoftwareFeature	
<b>Parent Class:</b>	CIM_LogicalElement	
Property	Description	Data Type
IdentifyingNumber	Provides product identification such as a serial number on software.	string
ProductName	Identifies the commonly used product name.	string
Vendor	Identifies the name of the product's supplier. Corresponds to the vendor property in the product object in the DMTF solution exchange standard.	string
Version	Identifies the product version information. Corresponds to the version property in the product object in the DMTF solution exchange standard.	string
Name	Defines the label by which the object is known to the users. This label is a user-defined name that uniquely identifies the element.	string

## DELL\_SoftwareFeature

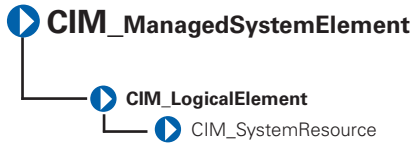


The `DELL_SoftwareFeature` described in Table 3-34 defines the universal resource locator (URL) of the systems management software and the language in which systems management information displays. Defining these properties enables users to manage a system using an Internet browser. You can access Server 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 3-34. DELL\_SoftwareFeature Properties**

<b>Class Name:</b>	DELL_SoftwareFeature	
<b>Parent Class:</b>	CIM_SoftwareFeature	
Property	Description	Data Type
OmsaURL	Defines the URL for Server Administrator.	string
Language	Sets the language for systems management information.	string
AgentVersion	Defines the version information of local CIM agent (same as ISVC version.)	string

## CIM\_SystemResource

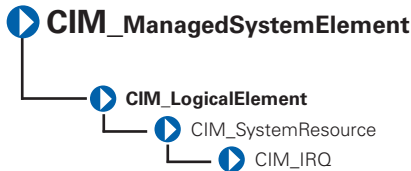


The `CIM_SystemResource` class listed in Table 3-35 provides access to system resources from an operating system. SystemResources consist of interrupt requests (IRQs) and direct memory access (DMA) capabilities.

**Table 3-35. CIM\_SystemResource Properties**

<b>Class Name:</b>	CIM_SystemResource
<b>Parent Class:</b>	CIM_LogicalElement

## CIM\_IRQ



The `CIM_IRQ` class described in Table 3-36 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 your computer (COM1) is assigned to `IRQ4` by default.

**Table 3-36. CIM\_IRQ Properties**

<b>Class Name:</b>	<code>CIM_IRQ</code>	
<b>Parent Class:</b>	<code>CIM_SystemResource</code>	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
<code>CSCreationClassName</code>	See Table 1-1.	string
<code>CSName</code>	See Table 1-1.	string
<code>CreationClassName</code>	See Table 1-1.	string
<code>IRQNumber</code>	Identifies the interrupt request number.	uint32
<code>Availability</code>	Indicates the availability of the <code>IRQ</code> . Values for the <code>Availability</code> property are as follows: 1 - Other 2 - Unknown 3 - Available 4 - In use/not available 5 - In use and available	uint16
<code>TriggerLevel</code>	Indicates whether the interrupt is triggered by the hardware signal going high or low. Values for the <code>TriggerLevel</code> property are as follows: 1 - Other 2 - Unknown 3 - Active low 4 - Active high	uint16
<code>TriggerType</code>	Indicates whether edge (value=4) or level triggered (value=3) interrupts occur. 1 - Other 2 - Unknown	uint16

**Table 3-36. CIM\_IRQ Properties (continued)**

<b>Class Name:</b>	CIM_IRQ	
<b>Parent Class:</b>	CIM_SystemResource	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
TriggerType	3 - Level 4 - Edge	uint16
Shareable	Indicates whether the IRQ can be shared. A value of TRUE indicates that the IRQ can be shared.	Boolean
Hardware	Indicates whether the interrupt is hardware- or software-based. (A value of TRUE indicates that the interrupt is hardware based.) On a personal computer, a hardware IRQ is a physical wire to a programmable interrupt controller (PIC) chip set through which the microprocessor can be notified of time critical events. Some IRQ lines are reserved for standard devices such as the keyboard, diskette drive, and the system clock. A software interrupt is a programmatic mechanism to allow an application to get the attention of the processor.	Boolean

## CIM\_MemoryMappedIO

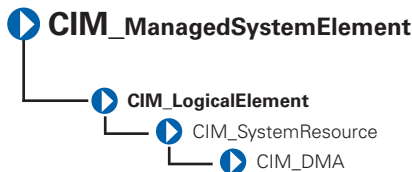


The CIM\_MemoryMappedIO class explained in Table 3-37 addresses both memory and port I/O resources for personal computer architecture memory mapped I/O.

**Table 3-37. CIM\_MemoryMappedIO Properties**

<b>Class Name:</b>	CIM_MemoryMappedIO	
<b>Parent Class:</b>	CIM_SystemResource	
Property	Description	Data Type
CSCreationClassName	See Table 1-1.	string
CSName	See Table 1-1.	string
CreationClassName	See Table 1-1.	string
StartingAddress	Identifies the starting address of memory mapped I/O.	uint64
EndingAddress	Identifies the ending address of memory mapped I/O.	uint64
MappedResource	Indicates the type of memory mapped I/O. MappedResource defines whether memory or I/O is mapped, and for I/O, whether 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	uint16

## CIM\_DMA

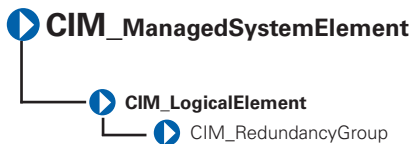


The CIM\_DMA class explained in Table 3-38 contains DMA information. A DMA channel allows certain types of data transfer between RAM and a device to bypass the microprocessor.

**Table 3-38. CIM\_DMA Properties**

<b>Class Name:</b>	CIM_DMA	
<b>Parent Class:</b>	CIM_SystemResource	
Property	Description	Data Type
CSCreationClassName	See Table 1-1.	string
CSName	See Table 1-1.	string
CreationClassName	See Table 1-1.	string
DMACHannel	Identifies a part of the object's key value, the DMA channel number.	uint32
Availability	Indicates the availability of the DMA. Values for the <b>Availability</b> property are as follows: 1 - Other 2 - Unknown 3 - Available 4 - In use/not available 5 - In use and available/shareable	uint16

## CIM\_RedundancyGroup

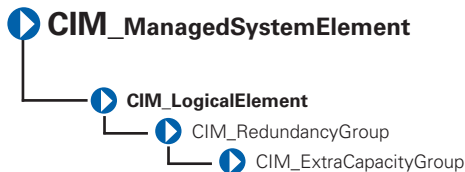


The `CIM_RedundancyGroup` class explained in Table 3-39 is a set of components that provide more instances of a critical component than are required for the system's operation. The extra components are used in case of critical component failure. For example, multiple power supplies allow a working power supply to take over when another power supply has failed.

**Table 3-39. CIM\_RedundancyGroup Properties**

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

## CIM\_ExtraCapacityGroup



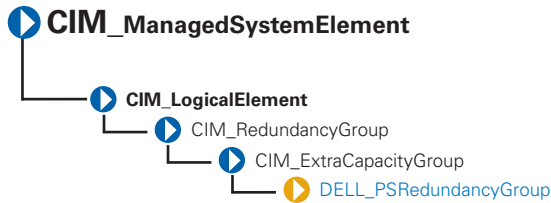
The `CIM_ExtraCapacityGroup` class explained in Table 3-40 applies to systems that have more capability and components than are required for normal operation, for example, systems that have extra fans or power supplies.



**Table 3-40. CIM\_ExtraCapacityGroup Properties**

<b>Class Name:</b>	CIM_ExtraCapacityGroup	
<b>Parent Class:</b>	CIM_RedundancyGroup	
Property	Description	Data Type
MinNumberNeeded	Specifies the smallest number of elements that must be operational in order to have redundancy. For example, in an N+1 redundancy relationship, the MinNumberNeeded property should be set to N.	uint32

## DELL\_PSRedundancyGroup

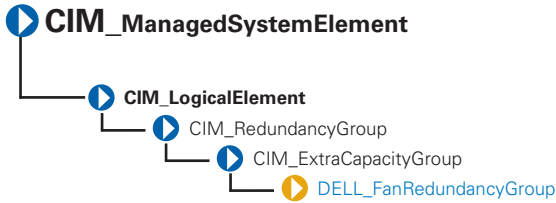


The **DELL\_PSRedundancyGroup** described in Table 3-41 is a Dell-specific extension of the **CIM\_PowerSupply** class. The **DELL\_PSRedundancyGroup** class defines what constitutes power supply redundancy in a system.

**Table 3-41. DELL\_PSRedundancyGroup Properties**

<b>Class Name:</b>	DELL_PSRedundancyGroup
<b>Parent Class:</b>	CIM_ExtraCapacityGroup

# DELL\_FanRedundancyGroup



The `DELL_FanRedundancyGroup` described in Table 3-42 defines what constitutes fan redundancy in a system.

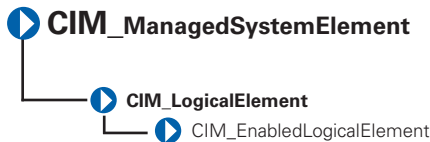
**Table 3-42. DELL\_FanRedundancyGroup Properties**

---

<b>Class Name:</b>	<code>DELL_FanRedundancyGroup</code>
<b>Parent Class:</b>	<code>CIM_ExtraCapacityGroup</code>

---

# CIM\_EnabledLogicalElementGroup



The `CIM_EnabledLogicalElementGroup` class described in Table 3-43 extends the `CIM_LogicalElementGroup` class to abstract the concept of an element that is enabled or disabled, such as a `LogicalDevice` or `ServiceAccessPoint`.

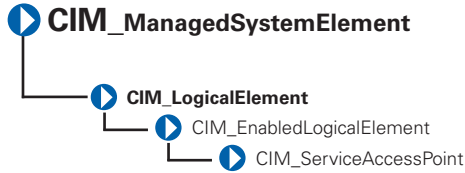
**Table 3-43. CIM\_EnabledLogicalElementGroup Properties**

---

<b>Class Name:</b>	<code>CIM_EnabledLogicalElementGroup</code>
<b>Parent Class:</b>	<code>CIM_LogicalElementGroup</code>

---

# CIM\_ServiceAccessPoint



The `CIM_ServiceAccessPointGroup` class described in Table 3-44 represents the ability to utilize or invoke a service. Access points indicate that a service is available to other entities for use.

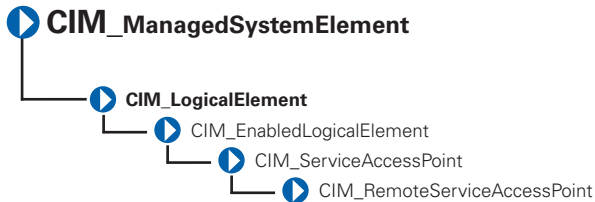
**Table 3-44. CIM\_ServiceAccessPointGroup Properties**

---

<b>Class Name:</b>	<code>CIM_ServiceAccessPointGroup</code>
<b>Parent Class:</b>	<code>CIM_EnabledLogicalElement</code>

---

# CIM\_RemoteServiceAccessPoint

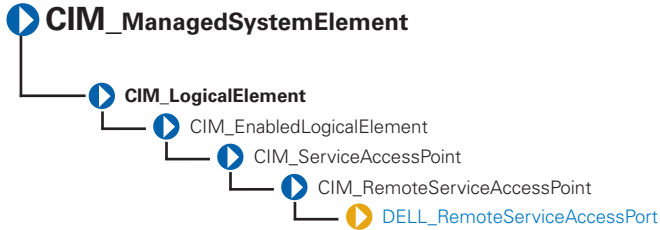


The `CIM_RemoteServiceAccessPointGroup` class identified in Table 3-45 describes the accessing and addressing of information for a remote connection that is known to a *local* network element. This information is contained in the *local* network element since this is the context in which it is *remote*. The relevance of the remote service access point and information on its use are described by subclassing or associating to the `CIM_RemoteServiceAccessPointGroup` class.

**Table 3-45. CIM\_RemoteServiceAccessPointGroup Properties**

<b>Class Name:</b>	CIM_RemoteServiceAccessPointGroup	
<b>Parent Class:</b>	CIM_ServiceAccessPointGroup	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
AccessInfo	Describes accessing or addressing of information for a remote connection. This can be a host name, network address, and other similar information.	string
InfoFormat	Indicates an enumerated integer describing the format and interpretation of the AccessInfo property. This property can have the following values: 1 - Other 2 - Host Name 3 - Ipv4 Address 4 - Ipv6 Address 5 - IPX Address 6 - DECnet Address 7 - SNA Address 8 - Autonomous System Number 9 - MPLS Label 10..99 - DMTF Reserved 100 - Dial String 101 - Ethernet Address 102 - Token Ring Address 103 - ATM Address 104 - Frame Relay Address 105..199 - DMTF Reserved 200 - URL 32768..65535 - Vendor Specific	uint16

# DELL\_RemoteServiceAccessPort



The `DELL_RemoteServiceAccessPortGroup` class described in Table 3-46 is an extended class of the `CIM_RemoteServiceAccessPointGroup` class. The `DELL_RemoteServiceAccessPortGroup` class provides information about Dell implementation-specific attributes.

**Table 3-46. DELL\_RemoteServiceAccessPortGroup Properties**

<b>Class Name:</b>	<code>DELL_RemoteServiceAccessPortGroup</code>	
<b>Parent Class:</b>	<code>CIM_RemoteServiceAccessPointGroup</code>	
Property	Description	Data Type
PortName	Displays the name of the service access port.	string
VersionString	Indicates the version of the access point service.	string
RemoteAccessType	Indicated the type of remote access service. This property can have the following values: 0 - BMC 8 - IMC 9 - CMC 10 - iDRAC6 11 - iDRAC6 for modular systems 13 - BMC	uint16



## Dell-Defined Classes

The Dell-defined classes are defined and populated by Dell rather than by CIM. None of these classes have a parent class and are on the same level as `CIM_ManagedSystemElement`. For information on how the logs are formatted, see Table 2-5.

**Figure 4-1. Dell\_EsmLog**



The `DELL_EsmLog` class described in Table 4-1 records failure threshold violations collected by Server Administrator's embedded server management (ESM) capabilities.

**Table 4-1. DELL\_EsmLog Properties**

<b>Class Name:</b>	DELL_EsmLog	
<b>Parent Class:</b>	None	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
recordNumber	Provides an index to the ESM table.	uint32
logRecord	Provides the ESM message content.	string
eventTime	Indicates the time that the message is generated.	datetime
status	Indicates the severity of the event that caused the log to be generated.	string

# DELL\_PostLog



The `DELL_PostLog` identified in Table 4-2 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 4-2. DELL\_PostLog Properties**

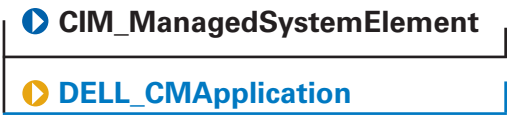
---

<b>Class Name:</b>	<code>DELL_PostLog</code>
<b>Parent Class:</b>	None

---

# DELL\_CMApplication

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





The `DELL_CMApplication` class identified in Table 4-3 contains information related to the Dell change management applications.

**Table 4-3. DELL\_CMApplication**

<b>Class Name:</b>	<code>DELL_CMApplication</code>	
<b>Parent Class:</b>	None	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
<code>componentType</code>	Defines the application type.	string
<code>subComponentID</code>	Defines an application string.	string
<code>version</code>	Indicates the current version of the application.	string
<code>name</code>	Indicates the name of the application.	string
<code>deviceKey</code>	Indicates the device key of the application.	string

## DELL\_CMDevice

 **CIM\_ManagedSystemElement**

 **DELL\_CMDevice**

The `DELL_CMDevice` identified in Table 4-4 contains information related to the Dell change management device.

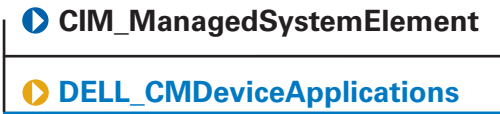
**Table 4-4. DELL\_CMDevice Properties**

<b>Class Name:</b>	<code>DELL_CMDevice</code>	
<b>Parent Class:</b>	None	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
<code>componentID</code>	Defines a component string.	string
<code>name</code>	Indicates the name of the device.	string
<code>vendorID</code>	Defines an ID for vendor supplying the device.	string

**Table 4-4. DELL\_CMDevice Properties (continued)**

<b>Class Name:</b>	DELL_CMDevice	
<b>Parent Class:</b>	None	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
subVendorID	Defines an ID for an additional vendor supplying the device.	string
deviceID	Indicates the ID of the device.	string
subDeviceID	Indicates the ID for additional device.	string
bus	Indicates the PCI bus number.	string
device	Indicates the PCI device number.	string
function	Indicates the PCI Function number.	string

## DELL\_CMDeviceApplication

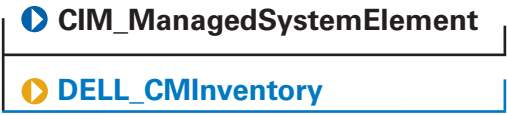


The DELL\_CMDeviceApplication class identified in Table 4-5 contains information related to the Dell change management association between the device and application.

**Table 4-5. DELL\_CMDeviceApplication Properties**

<b>Class Name:</b>	DELL_CMDeviceApplication	
<b>Parent Class:</b>	None	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
antecedent	Refers to the device.	string
dependent	Refers to the application.	string

# DELL\_CMInventory

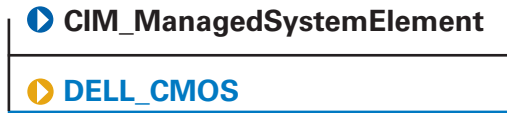


The DELL\_CMInventory identified in Table 4-6 contains information related to the Dell Change Management inventory.

**Table 4-6. DELL\_CMInventory Properties**

<b>Class Name:</b>	DELL_CMInventory	
<b>Parent Class:</b>	None	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
local	Indicates the locale of the system.	string
schemaVersion	Indicates the Inventory schema implemented by the system.	string
systemID	Defines the System ID.	string

# DELL\_CMOS

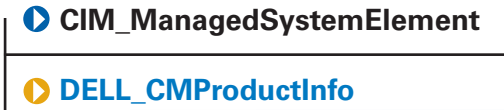


The DELL\_CMOS class identified in Table 4-7 contains information related to the Dell change management operating system.

**Table 4-7. DELL\_CMOS Properties**

<b>Class Name:</b>	DELL_CMOS	
<b>Parent Class:</b>	None	
Property	Description	Data Type
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` identified in Table 4-8 contains information related to the Dell change management product.

**Table 4-8. DELL\_CMProductInfo Properties**

<b>Class Name:</b>	DELL_CMProductInfo	
<b>Parent Class:</b>	None	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
name	Indicates the name of the product.	string
description	Provides a short description of the product.	string
vendor	Indicates the name of the product manufacturer.	string
version	Indicates the current version number of the product.	string

## DELL\_BIOSExtensions

The `DELL_BIOSExtensions` identified in Table 4-9 contains information related to the specific extension of the data attributes on your system.

**Table 4-9. DELL\_BIOSExtensions Properties**

<b>Class Name:</b>	DELL_BIOSExtensions	
<b>Parent Class:</b>	CIM_ManagedSystemElement	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
systemBIOSCharacteristics	Indicates the characteristics of BIOS on your system.	uint64
systemBIOSCharacteristicsExt1	Indicates the specific extension of the data attributes on your system.	uint8
systemBIOSCharacteristicsExt2	Indicates the specific extension of the data attributes on your system.	uint8

# DELL\_BIOSSettings

The DELL\_BIOSSettings identified in Table 4-10 contains information related to setting parameters in the Dell System Management BIOS.

**Table 4-10. DELL\_BIOSSettings Properties**

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

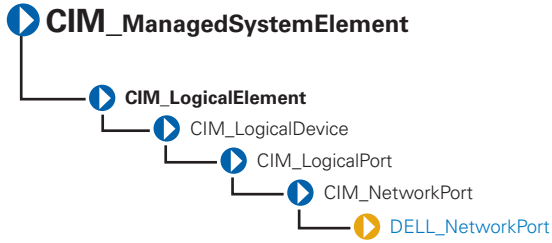
# DELL\_SDCardDevice

The DELL\_SDCard Devices identified in Table 4-11 contains information related to the SD card devices.

**Table 4-11. DELL\_SDCardDevice Properties**

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

# DELL\_NetworkPort



The `Dell_NetworkPort` class described in Table 4-12 represents the Dell specific features of the network adapters.



**Table 4-12. Dell\_NetworkPort Properties**

<b>Class Name:</b>	DELL_NetworkPort	
<b>Parent Class:</b>	CIM_Network Port	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
NIC TOE Capability	<p>Defines the TOE capability of the NIC. Values for the NIC TOE Capability property are:</p> <p>0 - NIC/driver does not support querying for capability.</p> <p>1 - NIC/driver supports querying for capability but query returned an error.</p> <p>2 - NIC/driver supports querying for capability and query says it is capable.</p> <p>4 - NIC/driver supports querying for capability and query says it is not capable.</p> <p>8 - NIC/driver supports querying for capability but an error prevented querying NIC/driver.</p> <p>16 - NIC/driver supports querying for capability but NIC/driver did not respond to query.</p> <p><b>NOTE:</b> Boolean value is defined if TOE is enabled. (Boolean is TOEEnable)</p>	uint 32

**Table 4-12. Dell\_NetworkPort Properties (continued)**

<b>Class Name:</b>	DELL_NetworkPort	
<b>Parent Class:</b>	CIM_NetworkPort	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
NIC RDMA Capability	<p>Defines the RDMA capability of the NIC. Values for the NIC RDMA Capability property are:</p> <p>0 - NIC/driver does not support querying for capability.</p> <p>1 - NIC/driver supports querying for capability but query returned an error.</p> <p>2 - NIC/driver supports querying for capability and query says it is capable.</p> <p>4 - NIC/driver supports querying for capability and query says it is not capable.</p> <p>8 - NIC/driver supports querying for capability but an error prevented querying NIC/driver.</p> <p>16 - NIC/driver supports querying for capability but NIC/driver did not respond to query.</p> <p><b>NOTE:</b> Boolean value is defined if RDMA is enabled. (Boolean is RDMAEnable).</p>	uint 32

**Table 4-12. Dell\_NetworkPort Properties (continued)**

<b>Class Name:</b>	DELL_NetworkPort	
<b>Parent Class:</b>	CIM_NetworkPort	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
NIC iSCSI Capability	<p>Defines the iSCSI capability of the NIC. Values for the NIC iSCSI Capability property are:</p> <p>0 - NIC/driver does not support querying for capability.</p> <p>1 - NIC/driver supports querying for capability but query returned an error.</p> <p>2 - NIC/driver supports querying for capability and query says it is capable.</p> <p>4 - NIC/driver supports querying for capability and query says it is not capable.</p> <p>8 - NIC/driver supports querying for capability but an error prevented querying NIC/driver.</p> <p>16 - NIC/driver supports querying for capability but NIC/driver did not respond to query.</p> <p><b>NOTE:</b> Boolean value is defined if iSCSI is enabled. (Boolean is iSCSIEnabled).</p>	uint 32
NicStatus	<p>Indicates the status of the NIC or driver</p> <p>The values for this property are:</p> <p>0 - Unknown</p> <p>1 - Connected</p> <p>2 - Disconnected</p> <p>3 - Driver Bad</p> <p>4 - Driver Disabled</p> <p>10 - Hardware Initializing</p>	uint 32

**Table 4-12. Dell\_NetworkPort Properties (continued)**

<b>Class Name:</b>	DELL_NetworkPort	
<b>Parent Class:</b>	CIM_NetworkPort	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
	12 - Hardware Closing 13 - Hardware Not Ready	
BusNumber	Indicates the PCI bus number.	uint 8
DeviceNumber	Indicates the PCI device number.	uint 8
FunctionNumber	Indicates the PCI function number.	uint 8
DriverVersion	Indicates the NIC driver version.	string
IPAddress	Indicates the NIC IP Address.	string
SubnetMask	Indicates the NIC subnet mask.	string
DHCPServer	Indicates the DHCP server.	string
DefaultGateway	Indicates the Default Gateway.	string
CurrentMacAddress	Indicates the NIC current MAC address.	string
OSAdapterDescription	Describes the Operating System adapter.	string
OSAdapterVendor	Describes the Operating System vendor.	string
OSProductName	Describes the product name of the Operating System.	string
ServiceName	Indicates the service name.	string

## DELL\_PowerConsumptionAmpsSensor

The DELL\_PowerConsumptionAmpsSensor identified in Table 4-13 contains information related to monitoring the power consumption.

**Table 4-13. DELL\_PowerConsumptionAmpsSensor**

<b>Class Name:</b>	DELL_PowerConsumptionAmpsSensor	
<b>Parent Class:</b>	CIM_Numeric Sensor	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
UnitModifier	See Table 1-1.	sint32
CurrentReading	See Table 1-1.	sint32
IsLinear	See Table 1-1.	Boolean
LowerThresholdNonCritical	See Table 1-1.	sint32
UpperThresholdNonCritical	See Table 1-1.	sint32
LowerThresholdCritical	See Table 1-1.	sint32
UpperThresholdCritical	See Table 1-1.	sint32

## DELL\_PowerConsumptionWattsSensor

The DELL\_PowerConsumptionWattsSensor identified in Table 4-14 contains information related to monitoring the power consumption.

**Table 4-14. DELL\_PowerConsumptionWattsSensor**

<b>Class Name:</b>	DELL_PowerConsumptionWattsSensor	
<b>Parent Class:</b>	CIM_Numeric Sensor	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
UnitModifier	See Table 1-1.	sint32
CurrentReading	See Table 1-1.	sint32
IsLinear	See Table 1-1.	Boolean
LowerThresholdNonCritical	See Table 1-1.	sint32
UpperThresholdNonCritical	See Table 1-1.	sint32
LowerThresholdCritical	See Table 1-1.	sint32

**Table 4-14. DELL\_PowerConsumptionWattsSensor (continued)**

<b>Class Name:</b>	DELL_PowerConsumptionWattsSensor	
<b>Parent Class:</b>	CIM_Numeric Sensor	
UpperThresholdCritical	See Table 1-1.	sint32

## DELL\_PowerConsumptionData

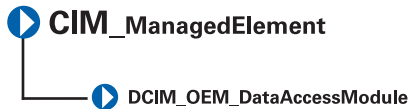
The DELL\_PowerConsumptionData identified in Table 4-15 contains information about the total power consumed from a start time and peak values registered during a time period.

**Table 4-15. DELL\_PowerConsumptionData**

<b>Class Name:</b>	DELL_PowerConsumptionData	
<b>Parent Class:</b>	CIM_Logical Device	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
cumulative PowerReading	Indicates the total power consumed from a start time.	uint 32
peakAmpReading	Indicates the time from which the peak amperage reading is being monitored.	uint 16
peakWattReading	Indicates the time from which the peak watt reading is being monitored.	uint 16
ResetCounters	Is the function used to reset the peak readings.	uint 32
powerCapSetting	This refers to the user configured power setting.	uint 16
instHeadroom	This refers to the instantaneous headroom.	uint 32
peakHeadRoom	Is the function used to set the power budget.	uint 32

# DCIM\_OEM\_DataAccessModule

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



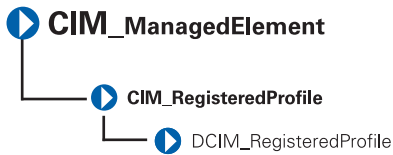
**Table 4-16. DCIM\_OEM\_DataAccessModule Properties**

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

**Table 4-16. DCIM\_OEM\_DataAccessModule Properties (continued)**

<b>Class Name:</b> DCIM_OEM_DataAccessModule		
<b>Parent Class:</b> CIM_ManagedElement		
Property	Description	Data Type
iDRACIPv6	Provides Remote Access controller (iDRAC ) IPv6 address.	string

## DCIM\_RegisteredProfile



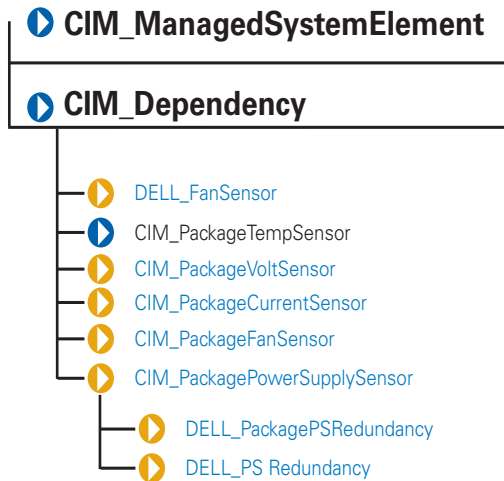
The `DCIM_RegisteredProfile` class is derived from the `CIM_RegisteredProfile` class. This class advertises the capabilities of `DCIM_OEM_DataAccessModule`.



# CIM\_Dependency

The `CIM_Dependency` class is an association used to establish dependency relationships between two managed system elements. `CIM_Dependency` shown in Figure 5-1 does not have a parent class because it is a relationship or association between two elements.

**Figure 5-1. CIM\_Dependency Class Structure**



Each class derived from `CIM_Dependency` 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.

# DELL\_FanSensor

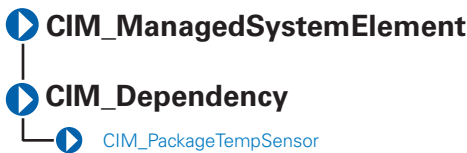


The `DELL_FanSensor` class described in Table 5-1 defines a Dell-specific association between a fan and a sensor. The `CIM_PackageFanSensor` class contains fans that assist in cooling the entire package as opposed to a fan dedicated to cooling only some of the components in the package.

**Table 5-1. DELL\_FanSensor Properties**

<b>Class Name:</b>	<code>DELL_FanSensor</code>
<b>Parent Class:</b>	<code>CIM_Dependency</code>
<b>Element</b>	<b>Description</b>
Antecedent	<code>CIM_Tachometer</code> refers to the tachometer (fan sensor) that measures the RPM of the fan.
Dependent	<code>CIM_Fan</code> refers to the fan whose revolutions are measured by the tachometer.

# CIM\_PackageTempSensor

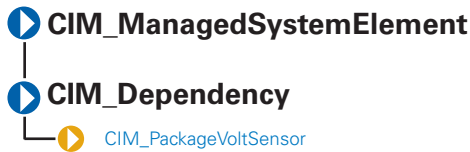


The `CIM_PackageTempSensor` class listed in Table 5-2 contains temperature sensors that are often installed in a package such as a chassis or a rack to assist in the monitoring of the package in general. This relationship is described by the `CIM_PackageTempSensor` association.

**Table 5-2. CIM\_PackageTempSensor Properties**

<b>Class Name:</b>	CIM_PackageTempSensor
<b>Parent Class:</b>	CIM_Dependency
<b>Element</b>	<b>Description</b>
Antecedent	CIM_TempSensor refers to the temperature sensor for the package.
Dependent	CIM_PhysicalPackage refers to the physical package whose environment is being monitored.

## CIM\_PackageVoltSensor

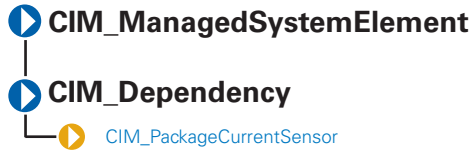


The `CIM_PackageVoltSensor` identified in Table 5-3 contains voltage sensors that are often installed in a package such as a chassis or a rack to assist in the monitoring of the package in general. This relationship is described by the `CIM_PackageVoltSensor` association.

**Table 5-3. CIM\_PackageVoltage Properties**

<b>Class Name:</b>	CIM_PackageVoltSensor
<b>Parent Class:</b>	CIM_Dependency
<b>Element</b>	<b>Description</b>
Antecedent	CIM_PackageVoltSensor refers to the voltage sensor for the package.
Dependent	CIM_PhysicalPackage refers to the physical package whose voltages are being monitored.

# CIM\_PackageCurrentSensor

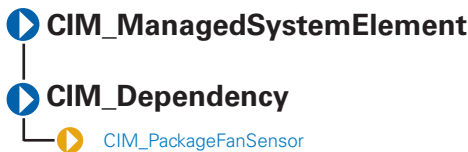


The `CIM_PackageCurrentSensor` shown in Table 5-4 contains amperage sensors that are often installed in a package such as a chassis or a rack to assist in the monitoring of the package in general. This relationship is described by the `CIM_PackageCurrentSensor` association.

**Table 5-4. CIM\_PackageCurrentSensor Properties**

<b>Class Name:</b>	<code>CIM_PackageCurrentSensor</code>
<b>Parent Class:</b>	<code>CIM_Dependency</code>
Element	Description
Antecedent	<code>CIM_CurrentSensor</code> refers to the amperage sensor for the package.
Dependent	<code>CIM_PhysicalPackage</code> refers to the physical package whose amperage is being monitored.

# CIM\_PackageFanSensor



The `CIM_PackageFanSensor` class described in Table 5-5 contains fan sensors that monitor the whole package.

**Table 5-5. CIM\_PackageFanSensor Properties**

<b>Class Name:</b>	CIM_PackageFanSensor
<b>Parent Class:</b>	CIM_Dependency
<b>Element</b>	<b>Description</b>
Antecedent	CIM_Fan refers to the cooling device for the package.
Dependent	CIM_PhysicalPackage refers to the physical package whose environment is being monitored.

## CIM\_PackagePowerSupplySensor

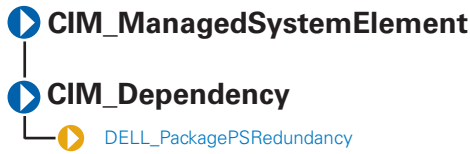


The CIM\_PackagePowerSupplySensor class described in Table 5-6 contains power supplies that provide power to the whole package.

**Table 5-6. CIM\_PackagePowerSupplySensor Properties**

<b>Class Name:</b>	CIM_PackagePowerSupplySensor
<b>Parent Class:</b>	CIM_Dependency
<b>Element</b>	<b>Description</b>
Antecedent	CIM_PowerSupplySensor refers to the power supply sensor that monitors wattage for the entire package.
Dependent	CIM_PhysicalPackage refers to the package whose wattage is being monitored.

# DELL\_PackagePSRedundancy



The `DELL_PackagePSRedundancy` class listed in Table 5-7 defines what constitutes power supply redundancy for an entire package.

**Table 5-7. DELL\_PackagePSRedundancy Properties**

<b>Class Name:</b>	<code>DELL_PackagePSRedundancy</code>
<b>Parent Class:</b>	<code>CIM_Dependency</code>
Element	Description
Antecedent	<code>DELL_PSRedundancyGroup</code> refers to power supplies that deliver wattage for the entire package.
Dependent	<code>CIM_PhysicalPackage</code> refers to the package to which the wattage is being supplied.

# DELL\_PSRedundancy



The `DELL_PSRedundancy` class shown in Table 5-8 defines what constitutes power supply redundancy for Dell systems.

**Table 5-8. DELL\_PSRedundancy Properties**

<b>Class Name:</b>	DELL_PSRedundancy
<b>Parent Class:</b>	CIM_Dependency
<b>Element</b>	<b>Description</b>
Antecedent	CIM_PowerSupplySensor refers to the power supply sensor that monitors wattage for the entire package.
Dependent	CIM_PhysicalPackage refers to the package whose wattage is being monitored.

## DELL\_AssociatedSupplyPCAmps

The DELL\_AssociatedSupplyPCAmps identified in Table 5-9 is a PowerConsumptionAmpsSensor associated with a CIM\_PowerSupply which is defined by this class.

**Table 5-9. DELL\_AssociatedSupplyPCAmps**

<b>Class Name:</b>	DELL_AssociatedSupplyPCAmps	
<b>Parent Class:</b>	CIM_Dependency	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
Antecedent	Indicates the PowerSupply instance.	uint 16
Dependent	Indicates the PowerConsumptionAmpsSensor associated with the CIM_PowerSupply.	uint 16

## DELL\_AssociatedSystemPCWatts

The DELL\_AssociatedSystemPCWatts identified in Table 5-10 a PowerConsumptionAmpsSensor associated with a Dell\_System which is defined by this class.

**Table 5-10. DELL\_AssociatedSystemPCWatts**

<b>Class Name:</b>	DELL_AssociatedSystemPCWatts	
<b>Parent Class:</b>	CIM_Dependency	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
Antecedent	Indicates the Dell_System instance.	uint 16
Dependent	Indicates the PowerConsumptionWattsSensor associated with the system.	uint 16

## AssociatedSystemPCData

The AssociatedSystemPCData identified in Table 5-11 is a PowerConsumptionData associated with a Dell\_System which is defined by this class.

**Table 5-11. AssocaitedSystemPCData**

<b>Class Name:</b>	DELL_AssociatedSupplyPCAmps	
<b>Parent Class:</b>	CIM_Dependency	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
Antecedent	Indicates the Dell_System instance.	uint 16
Dependent	Indicates the PowerConsumptionData associated with the Power Supply.	uint 16



# DELL\_PowerProfileData

The DELL\_PowerProfileData identified in Table 5-12 contains information related to power profiling and power knob data.

**Table 5-12. DELL\_PowerProfileData**

<b>Class Name:</b>	DELL_PowerProfileData	
<b>Parent Class:</b>	CIM_LogicalDevice	
<b>Property</b>	<b>Description</b>	<b>Data Type</b>
chassisIndex	Indicates the chassisIndex for this power profile.	uint 8
supportedProfile	Indicates the supported profiles.	uint 16
profileSetting	Indicates the Profile setting.	uint 16
customCPUCaps	Indicates the Custom Profile CPU management capability.	uint 16
customCPUSettings	Indicates the Custom Profile CPU management setting.	uint 16
customMemCaps	Indicates the Custom Profile memory management capability.	uint 16
customMemSettings	Indicates the Custom Profile memory management capability.	uint 16
customFanSettings	Indicates the Custom Profile fan management setting.	uint 16



# Index

## C

CIM classes and properties, 10  
  base classes, 11  
  classes that describe  
    relationships, 12  
  common properties of classes, 13  
  conventions, 12  
  parent classes, 12

CIM\_Chip, 26

CIM\_Dependency

  CIM\_PackageCurrentSensor, 116

  CIM\_PackageFanSensor, 116

  CIM\_PackagePowerSupplySensor,  
    117

  CIM\_PackageTempSensor, 114

  CIM\_PackageVoltSensor, 115

  DELL\_FanSensor, 114

  DELL\_PackagePSRedundancy, 1  
    18

  DELL\_PSRedundancy, 118

CIM\_LogicalElement

  CIM\_BIOSElement, 78

  CIM\_CacheMemory, 76

  CIM\_ComputerSystem, 40

  CIM\_Controller, 60

  CIM\_CurrentSensor, 49

  CIM\_DiscreteSensor, 44

  CIM\_DMA, 86

  CIM\_ExtraCapacityGroup, 88

  CIM\_Fan, 53

  CIM\_FRU, 42

  CIM\_IRQ, 83

  CIM\_Keyboard, 56

  CIM\_LogicalDevice, 41

  CIM\_LogicalPort, 43

  CIM\_MemoryMappedIO, 85

  CIM\_NumericSensor, 45

  CIM\_ParallelController, 61

  CIM\_PCIBridge, 65

  CIM\_PCIController, 63

  CIM\_PointingDevice, 55

  CIM\_PowerSupply, 57

  CIM\_Processor, 66

  CIM\_RedundancyGroup, 87

  CIM\_Sensor, 42

  CIM\_SerialController, 62

  CIM\_SoftwareElement, 78

  CIM\_SoftwareFeature, 81

  CIM\_System, 39

  CIM\_SystemResource, 83

  CIM\_Tachometer, 51, 104

  CIM\_TemperatureSensor, 48

  CIM\_UserDevice, 54

  CIM\_VoltageSensor, 50

  CIM\_Watchdog, 104

  DELL\_FanExtraCapacityGroup,  
    90

  DELL\_PSRedundancyGroup, 89

  DELL\_System, 40

CIM\_PhysicalComponent, 26

CIM\_PhysicalElement

- CIM\_PhysicalConnector, 30
- CIM\_PhysicalMemory, 28
- CIM\_PhysicalElement, 19
  - CIM\_Chassis, 23
  - CIM\_Chip, 26
  - CIM\_PhysicalComponent, 26
  - CIM\_PhysicalFrame, 22
  - CIM\_PhysicalPackage, 21
  - CIM\_Slot, 33
  - DELL\_Chassis, 24
  - structure of, 19
- CIM\_PhysicalElementClass
  - structure of, 19
- CIM\_SoftwareElement, 78
  - class name, 16
  - common properties of classes, 13
  - current reading, 13

## **D**

- data type, 16
- DCIM\_OEM\_DataAccessModule, 93
- DCIM\_RegisteredProfile, 112
- Dell OpenManage Server Agent, 9
- DELL\_CMApplication, 96
- DELL\_CMDevice, 97
- DELL\_CMDeviceApplication, 98
- DELL\_CMInventory, 99

- DELL\_CMOS, 99
- DELL\_CMProductInfo, 100
- DELL\_PostLog, 96
- DELL\_PSRedundancy, 118
- description, 16

## **P**

- parent class, 16
- property, 16

## **S**

- Server Administrator 1.0, 9

## **V**

- version, 15