Dell™ PowerVault™ ML6000
SMI-S Reference Guide

Contents

About This Guide and Your Product

This reference guide provides a high-level overview of Storage Management Initiative-Specification (SMI-S) and the Common Information Model (CIM) for use with the PowerVault™ ML6000 libraries. It describes how to use CIM and Managed Object Format (MOF) files to obtain physical and logical entity information from your library, and how to use CIM indications to monitor the status of the library.

This reference guide is written for management application developers and system administrators who have a working knowledge of SMI-S. Readers should be able to understand and use intrinsic methods supported by CIM, and should also understand Unified Modeling Language (UML).

For information about integrating SMI-S into a management application, contact your software vendor. For more information about SMI-S, refer to the Storage Network Industry Association (SNIA) website at www.snia.org.

Explanation of Symbols and Notes

The following symbols appear throughout this document to highlight important information.

WARNING: A WARNING INDICATES A POTENTIAL FOR PROPERTY DAMAGE, PERSONAL INJURY, OR DEATH.

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

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

Other Documents you Might Need

The following documents are also available for this product. These documents can be found on the product CD.

- *Dell PowerVault ML6000 Tape Library User's Guide*

Getting More Information or Help

More information about this product is available at http://support.dell.com.

Understanding SMI-S, CIM, and MOF

The ML6000 library supports the industry standard SMI-S application programming interface (API) described in the Storage Management Initiative Specification (SMI-S) version 1.2.

SMI-S is a standard management interface that facilitates the management of multi-vendor devices in a storage area networks (SANs) environment. SMI-S uses an object-oriented model based on the Common Information Model (CIM) and Web-Based Enterprise Management (WBEM) standards to define a set of objects and services that can manage the various elements of a SAN. By using a standardized architecture, SMI-S helps management application developers create common and extensible applications that work across multiple SAN vendor products.

The SMI-S server provides a hosting environment for plug-in instrumentation of host-based resources and management proxies for resources with remote management protocols. The general purpose of the SMI-S server is as follows:

- Implements Service Location Protocol (SLP) service agent functionality

- Implements CIM-server functionality as specified by the CIM-XML standard

- Supports the simple authentication scheme

- Supports Distributed Management Task Force (DMTF) CIM schema version 2.10

Managing SANs With SMI-S

SANs are becoming more and more common in multi-vendor network environments. SANs integrate host applications, fabric elements (such as switches and directors), and data storage devices from different vendors to create an interoperable storage network. Managing these elements from different vendors can be very challenging to network administrators, because each element has its own management interface, which may be proprietary. Network administrators must work with these disparate management APIs to build a cohesive management application that controls and monitors the SAN.

SMI-S addresses this management problem by creating a suite of flexible, open management API standards based on the vendor- and technology-independent CIM. Using the SMI-S APIs collected in profiles of common management classes, network administrators can create an interoperable management application, or CIM client, to control and monitor the disparate SAN elements that support SMI-S and CIM. With CIM servers either embedded within the SAN elements or supported by a proxy CIM server, these elements are accessible to the network administrator's CIM client application.

SMI-S uses SLP version 2 (RFC 2608) to discover CIM servers. After the CIM servers are identified, the CIM client uses a standard Server profile to determine which element-specific profiles are supported by the CIM servers. All SMI-S based CIM servers must have a Server profile.

Your Library, SMI-S, and CIM

A CIM server is embedded within the ML6000 library. The CIM server is SMI-S compliant and contains a Server profile and several subprofiles. For details about these profiles, refer to CIM Profiles and Subprofiles on page 7.

Understanding CIM

CIM is an object-oriented information model that describes management information in a network or enterprise environment. Because it is object-oriented, CIM provides abstraction, inheritance, and dependency or association relationships between objects within the model. CIM is based on XML and is platform-independent and technology neutral. Therefore, management application developers do not need to understand how CIM was implemented on a vendor product. Only the API is required to interact with a vendor product.

CIM uses a client/server model. The *CIM server* can either be embedded into the vendor product (as it is with your library) or it can be implemented by a proxy server that provides the CIM server functionality for the legacy vendor product. The *CIM client* is the management application that communicates with one or more CIM servers to manage the SAN. The CIM client discovers CIM servers through Service Location Protocol (SLP) version 2, as defined in RFC 2608. SLPv2 uses UDP port 427 for communication and is a discovery protocol that is separate from the CIM client/server communication path.

WBEM Initiative

The WBEM initiative is a set of management and Internet standards developed to unify the management of enterprise computing environments. The WBEM initiative includes the following:

- CIM, which provides a common format, language, and methodology for collecting and describing management data.

- The CIM-XML Encoding Specification, a standards-based method for exchanging CIM information. CIM-XML uses an xmlCIM encoded payload and HTTP as the transport mechanism. CIM-XML consists of the following specifications:

    - The xmlCIM encoding, a standard way to represent CIM information in XML format.

    - CIM Operations over HTTP, a transportation method that describes how to pass xmlCIM encoded messages over HTTP.

For more information about the WBEM initiative, go to the DMTF website at http://www.dmtf.org.

Unified Modeling Language

SMI-S relies on object-oriented classes as defined in CIM. These classes are frequently defined using Unified Modeling Language (UML). To understand SMI-S and the Dell extensions present in this document, you must have a basic understanding of CIM classes and UML.

A class is a collection of properties and methods that define a type of object. For example, a generic network device is a type of object. To describe this object, you could define the `NetworkDevice` class. You could then define this `NetworkDevice` class with the properties or attributes of a network device, such as `IpAddress` and `DeviceType`. You can also control your network device through the `NetworkDevice` class by adding methods or routines that can trigger specific actions on your network device. Example methods are `enablePort()` and `rebootDevice()`.

After you have defined the `NetworkDevice` class, you can define a class for just switches. Since a switch is a special type of `NetworkDevice`, you can use the object-oriented concept of inheritance to define your `Switch` class. You can define the `Switch` class as a child of the `NetworkDevice` class, meaning that the `Switch` class automatically has the properties and methods of its parent class. From there, you can add properties and methods unique to a switch.

CIM defines a special type of class called an *association class*. An association class represents relationships between two or more classes. For example, you can define an association class to show the relationship between a `NetworkDevice` class and an `OperatingSystem` class. If there is a many-to-one or many-to-many relationship, the association class is considered an aggregation.

UML draws a visual representation of the classes that describe a product or technology. UML contains many visual elements, and only a subset of elements have been described here. For a full explanation of UML, go to http://www.uml.org.

Understanding MOF

CIM is described in the DMTF's Managed Object Format (MOF), a language based on the Object Management Group's Interface Definition Language (IDL). The MOF syntax describes object-oriented class and instance definitions in textual form, with the goals of human readability and parsing by a compiler.

The main components of a MOF specification are:

- Textual descriptions of element qualifiers (meta-data about classes, properties, methods, etc.)
- Comments and compiler directives
- The specific class and instance definitions that convey the semantics of the CIM schema

These MOF files are an extension to the standard CIM schema version 2.9.

For information about the standard CIM schema version 2.9 MOF files, go to the DMTF website at the following URL: http://www.dmtf.org. To view copies of the Dell Managed Object Format (MOF) file for the ML6000 library, see .

3

CIM Profiles and Subprofiles

SMI-S defines a number of profiles that specify the managed objects that control and monitor elements of a Storage Area Network (SAN). The CIM server embedded within your library supports the following standard profiles:

- Server Profile, Version 1.2

- Storage Library Profile, Version 1.2

When using CIM servers to manage a network of SAN elements, you must first discover the location of all available CIM servers and discover which services they support. The Service Location Protocol (SLP) version 2 provides this discovery mechanism. CIM clients use SLPv2 to discover CIM servers by gathering generic information about which services the CIM servers provide and the specific URL where these services are located. After the CIM client discovers the CIM servers within the SAN, the CIM client must determine the level of support that each CIM server provides. For details about discovering the CIM server embedded in your library, refer to Discovering the CIM Server on page 12.

The next step to using CIM servers to manage a network of SAN elements is to implement CIM indications. For details about CIM indications, refer to Using CIM Indications on page 13.

Server Profile, Version 1.2

Your library contains a CIM server, and the CIM server includes a standard profile called the Server profile. The Server profile defines the capabilities of a CIM server. This includes providing the namespace for the CIM server and all profiles and subprofiles that the CIM server supports. For each supported profile, the Server profile instantiates the `RegisteredProfile` class. Each instance of this class gives the CIM client the profile name and unique ID that is supported by the CIM server. Similarly, the CIM server lists all supported optional subprofiles, using the `RegisteredSubProfile` class and the `SubprofileRequiresProfile` association class to associate the subprofile with the profile.

Storage Library Profile, Version 1.2

The schema for a storage library provides the classes and associations necessary to represent various forms of removable media libraries. This profile is based upon the CIM 2.9 object model and defines the subset of classes that supply the information necessary for robotic storage libraries.

This profile further describes how the classes are to be used to satisfy various use cases, and offers suggestions to agent implementors and client application developers. Detailed descriptions of classes can be found in the CIM 2.9 preliminary schema.

For a comprehensive view of all the supported profiles and subprofiles, see Figure 1 on page 8.

Figure 1 Storage Library Profile, Version 1.2
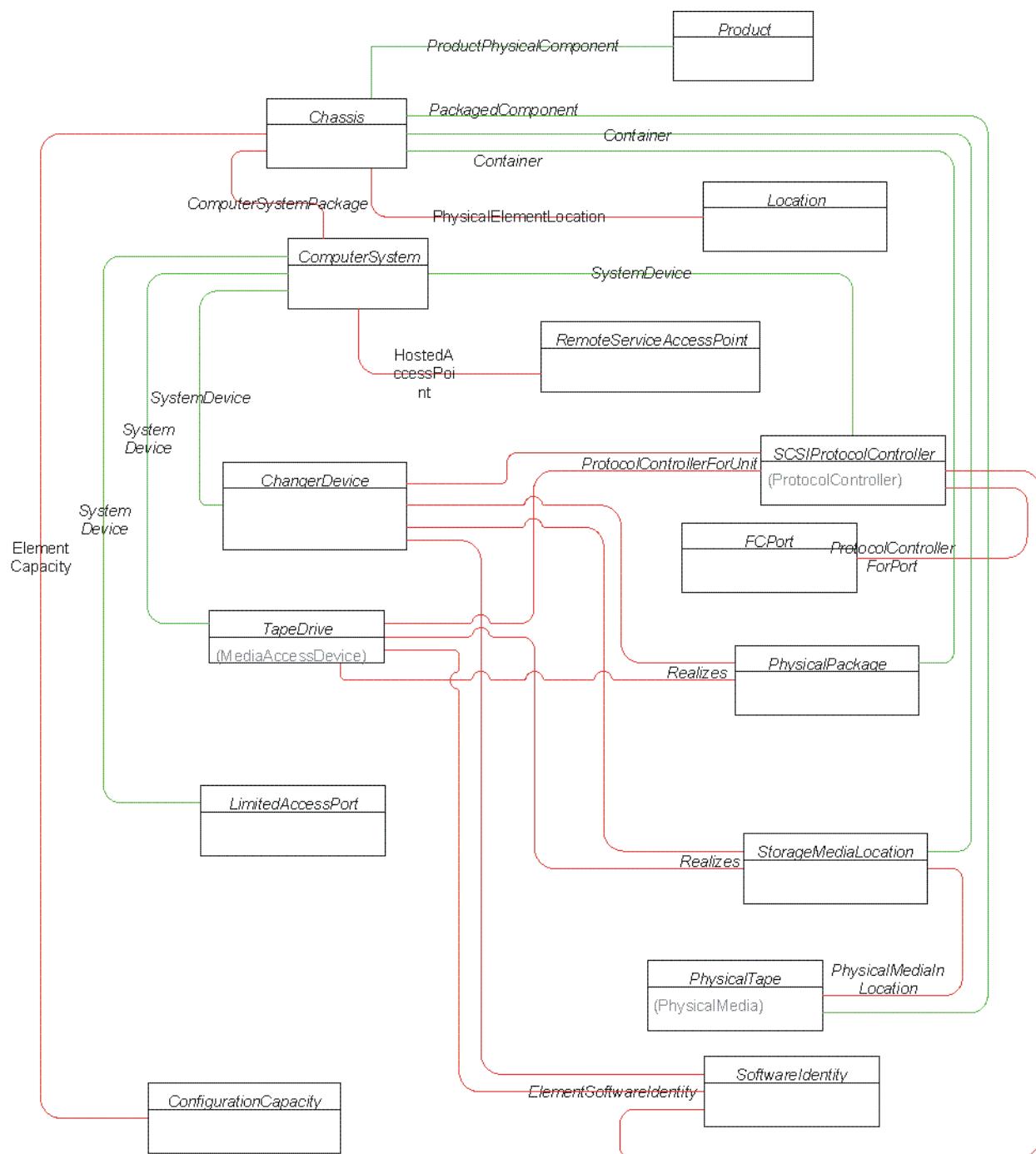
Storage Library Subprofiles

The following subprofiles are available for the storage library profile.

Library Capacity Subprofile

Using the `ConfigurationCapacity` and `ElementCapacity` classes, servers publish the minimum and maximum number of slots, drives, magazines, media changers, and other elements associated with the library.

Software Subprofile

Using the `SoftwareIdentity` class, provides information on the installed controller software. This is linked to the controller using an `InstalledSoftwareIdentity` association.

Location Subprofile

Using an instance of a `Location` class and the `PhysicalElementLocation` association, provides the location of a SAN element. Associated with product information, a `PhysicalPackage` may also have a location.

Access Points Subprofile

Provides remote access points for management tools.

Devices with a web GUI allow device-specific configurations. This is modeled using a `RemoteServiceAccessPoint`. This is linked to the managed element using a `HostedAccessPoint` association. Only the `HostedAccessPoint` association needs to be instantiated. The `HostedAccessPoint` associates the service to the System on which it is hosted.

Limited Access Port Subprofile

Libraries contain Limited Access Ports elements (mail slots, cartridge access ports, or import/export elements). This subprofile defines the required classes necessary to publish information about these common components.

Physical Package Subprofile

CIM has a strong separation between the physical and logical sides of the model. A system is "realized" using a `SystemPackaging` association to a `PhysicalPackage` (or one of its subclasses such as `Chassis`). The physical containment model can then be built using `Container` associations and subclasses.

Physical elements can be described as products by using the `Product` and `ProductPhysicalComponent` associations.

Fibre Channel Target Port Subprofile

The Fibre Channel Target Port Subprofile models the Fibre Channel-specific aspects of a target storage system. For Fibre Channel ports, the concrete subclass of `CIM_LogicalPort` is `CIM_FCPort`. `CIM_FCPort` is always associated 1 to 1 with a `CIM_SCSIProtocolEndpoint` instance.

Element Counting Subprofile

The Element Counting Subprofile defines methods to count the number of physical tapes, storage media locations, and other classes within a storage library (or other system type). Such methods allow clients to avoid retrieving all instances of physical element classes simply to count them. Therefore, network traffic will be saved between client applications and storage library providers. These methods are modeled by the Configuration Reporting Service hosted by the storage library's (or other system type's) top-level Computer System.

Service Discovery and CIM Indications

This chapter explains how to discover the CIM server embedded within your library, and how to use CIM indications to help you monitor the status of the library.

Setting up Authentication and Privacy

To use the CIM server embedded within your library, the CIM server must be enabled and your CIM clients must connect to it using the appropriate credentials.

Enabling the CIM Server

The CIM server uses port 5988 for HTTP and port 5989 for HTTPS. To enable the CIM server in your library, HTTP and/or HTTPS must be enabled. By default, HTTP port 5988 is enabled on the ML6000 library, while the HTTPS port 5989 can be selectively either enabled or disabled.

Enabling HTTP or HTTPS also enables the Service Location Protocol (SLP) service. The SLP service is also enabled by default. SLP is disabled if both HTTP and HTTPS are disabled.

You can disable the CIM server by disabling HTTPS; HTTP is always enabled.

If you disable the CIM server and need to enable it again in the future, refer to the library's *Dell PowerVault ML6000 Tape Library User's Guide*.

CIM Clients

CIM clients can connect to the library using the same identity and credential information as the remote web client of the ML6000. The interop namespace is `root/cimv2`.

The ML6000 library supports Secure Socket Layer (SSL) 3.0 authentication. To use SSL, you must have an SSL certificate. The SSL certificate is self generated. When you enable SSL from the library interface, the certificate is presented the first time you attempt to connect to the library via SSL. For information about enabling SSL on the library, see the *Dell PowerVault ML6000 User's Guide*.

Discovering the CIM Server

When you discover the CIM server that is embedded within your library, you not only determine its location within your SAN, but you also determine which services it supports (as defined by the server's profiles).

Discovering CIM servers provides information about the physical and logical entities within your SAN. This information changes dynamically as SAN entities are added, moved, or removed. This discovery process also helps you discover object classes and related association classes, as well as return status codes that are provided by servers in the managed environment.

You can discover the CIM server in your library by using the following:

- IP address and hostname of the library

- SLP

    SLP supports the WBEM SLP template, v1.0.0. You can use the following profiles and subprofiles for discovery and performance monitoring:

    - Storage Library Profile (Storage Library Profile, Version 1.2)
    - Library Capacity Subprofile
    - Software Subprofile
    - Location Subprofile
    - Access Points Subprofile
    - Limited Access Port Subprofile
    - Physical Package Subprofile
    - Fibre Channel Target Port Subprofile
    - Element Counting Subprofile

- Various methods defined by CIM

    Use the following methods, which are intrinsic methods defined by CIM, to retrieve information about your library.

    The location of the CIM server, which is identified by the library's IP address, is the target of these methods. These methods are supported for the CIM protocol only and cannot be entered as commands.

Table 1 CIM Methods

Method	How it is Used
enumerateInstances()	Enumerates instances of a CIM class
enumerateInstanceNames()	Enumerates names of instances of a CIM class
getInstance()	Gets a CIM instance
associators()	Enumerates associators of a CIM object
associatorName()	Enumerates names of associators of a CIM object
references()	Enumerates references to a CIM object
referenceName()	Enumerates names of references to a CIM object

Using CIM Indications

A CIM indication is a message that communicates a change in the status of an SMI-S managed device—in this case, a change in the status of the library. Your library can send these messages, or indications, to any CIM client that has subscribed to receive them.

To enable indications, first subscribe your CIM clients to receive indications from the library, and then use a WQL query to identify each indication that you want to receive.

To subscribe your CIM clients to receive indications from your library

1 Create a CIM_ListenerDestinationCIMXML instance for each client listening for indications. Clients can only receive indications on port 61000 over http or https. So the CIM_ListenerDestinationCIMXML. Destination should use the http://< ip address of the client>:61000 format. For example, http:// 172.16.42.122:61000

2 Use the appropriate WBEM query language (WQL) queries from Table 2 to create the supported Indication filters.

3 Create a CIM_IndicationSubscription association between the CIM_ListenerDestinationCIMXML and the CIM_IndicationFilters.

For details on how to create and receive indications, refer to the Indications subprofile in the SMI-S 1.2 specification.

The storage library profile and its subprofiles contain a list of indication filters that your CIM clients can receive, while the messages themselves are defined in the SMI-S Indications subprofile. The following table provides WQL queries for all indications supported by the ML6000.

Table 2   Supported Queries for CIM Indications

Area of Library	Query	Purpose
Library	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_ComputerSystem	Indicates that the library is available for use (after it has been powered on or restarted).
	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_ComputerSystem	Indicates that the library is shutting down.
	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ComputerSystem AND PreviousInstance.OperationalStatus <> SourceInstance.OperationalStatus	Indicates a change in the library's RAS status of overall library health.
Robotics	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_ChangerDevice	Indicates that the library's robotics system is ready for use (after library has initialized).
	CIM_SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_ChangerDevice	Indicates that the library has transitioned to a "not ready" state, and that its robotics system is currently unavailable.
	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ChangerDevice AND PreviousInstance.OperationalStatus <> SourceInstance.OperationalStatus	Indicates a change in the library's RAS status for the robotics system.
I/E Station and Tape Media	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA PhysicalMedia	Indicates that tape media has been inserted into the I/E station.
	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_PhysicalMedia	Indicates that tape media has been removed from the I/E station.
	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_PhysicalMedia AND PreviousInstance.OperationalStatus <> SourceInstance.OperationalStatus	Indicates a change in the library's RAS status for tape media.
Tape Drives/ I/O Blades	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_MediaAccessDevice	Indicates that a tape drive has been added to the library.
	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_MediaAccessDevice	Indicates that a tape drive has been removed from the library.

Table 2 Supported Queries for CIM Indications (Continued)

Area of Library	Query	Purpose
	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_MediaAccessDevice AND PreviousInstance.OperationalStatus <> SourceInstance.OperationalStatus	Indicates a change in the library's RAS status for tape drives.
	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_FCPort	Indicates that a Fibre Channel tape drive or FC I/O blade has been added to the library.
	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_FCPort	Indicates that a Fibre Channel tape drive or FC I/O blade has been removed from the library.
	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_FCPort AND PreviousInstance.OperationalStatus <> SourceInstance.OperationalStatus	Indicates a change in the RAS status of the Fibre Channel port on a tape drive. or FC I/O blade.

A
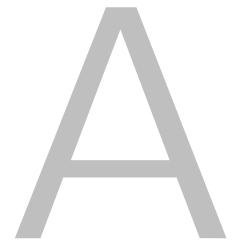
MOF Files

The Managed Object Format (MOF) files for the PowerVault ML6000 library are reproduced below.

MOF File for the PowerVault ML6000

```
// ===================================================================
// ComputerSystem
// ===================================================================
 [Description("CMPI SMIS ComputerSystem provider"),
  provider("cmpi::cmpicomputersystem")
 ]
class SMIS_ComputerSystem : CIM_ComputerSystem
{
      [Description ("Generate indication")]
   uint32 GenerateIndication();


};


// ===================================================================
// ComponentCS
// ===================================================================
 [Description("CMPI SMIS ComponentCS association provider"),
      provider("cmpi::cmpicomponentcs")
 ]
class SMIS_ComponentCS : CIM_ComponentCS
{
};
```

```
// ======================================================================
// Chassis
// ======================================================================
  [Description("CMPI SMIS Chassis provider"),
   provider("cmpi::cmpichassis")
  ]
class SMIS_Chassis : CIM_Chassis
{
};


// ======================================================================
// Product
// ======================================================================
  [Description("CMPI SMIS Product provider"),
   provider("cmpi::cmpiproduct")
  ]
class SMIS_Product : CIM_Product
{
};


// ======================================================================
// SoftwareIdentity
// ======================================================================
  [Description("CMPI SMIS Softwareidentity provider"),
   provider("cmpi::cmpisoftwareidentity")
  ]
class SMIS_SoftwareIdentity : CIM_SoftwareIdentity
{
};


// ======================================================================
// ConfigurationCapacity
// ======================================================================
  [Description("CMPI SMIS ConfigurationCapacity provider"),
   provider("cmpi::cmpiconfigurationcapacity")
  ]
class SMIS_ConfigurationCapacity : CIM_ConfigurationCapacity
```

```
{
};


// =================================================================
// FCPort
// =================================================================
 [Description("CMPI SMIS FCPort provider"),
  provider("cmpi::cmpifcport")
 ]
class SMIS_FCPort : CIM_FCPort
{
        [Description ("Generate indication")]
    uint32 GenerateIndication();
};


// =================================================================
// SCSIProtocolController
// =================================================================
 [Description("CMPI SMIS SCSIProtocolController provider"),
  provider("cmpi::cmpiscsiprotocolcontroller")
 ]
class SMIS_SCSIProtocolController : CIM_SCSIProtocolController
{
};


// =================================================================
// InstalledSoftwareIdentity
// =================================================================
 [Description("CMPI SMIS InstalledSoftwareIdentity provider"),
  provider("cmpi::cmpiinstalledsoftwareidentity")
 ]
class SMIS_InstalledSoftwareIdentity : CIM_InstalledSoftwareIdentity
{
};


// =================================================================
// ElementConformsToProfile
// =================================================================
```

```
 [Description("CMPI SMIS ElementConformsToProfile provider"),
      provider("cmpi::cmpielementconformstoprofile")
 ]
class SMIS_ElementConformsToProfile : CIM_ElementConformsToProfile
{
};


// ====================================================================
// HostedService
// ====================================================================
 [Description("CMPI SMIS HostedService provider"),
      provider("cmpi::cmpihostedservice")
 ]
class SMIS_HostedService : CIM_HostedService
{
};


// ====================================================================
// RegisteredProfile
// ====================================================================
 [Description("CMPI SMIS RegisteredProfile provider"),
     provider("cmpi::cmpiregisteredprofile")
 ]
class SMIS_RegisteredProfile : CIM_RegisteredProfile
{
};


// ====================================================================
// RegisteredSubProfile
// ====================================================================
 [Description("CMPI SMIS RegisteredSubProfile provider"),
     provider("cmpi::cmpiregisteredsubprofile")
 ]
class SMIS_RegisteredSubProfile : CIM_RegisteredSubProfile
{
};


// ====================================================================
```

```
// ObjectManager
// ====================================================================
  [Description("CMPI SMIS ObjectManager provider"),
      provider("cmpi::cmpiobjectmanager")
  ]
class SMIS_ObjectManager : CIM_ObjectManager
{
};


// ====================================================================
// ComputerSystemPackage
// ====================================================================
  [Description("CMPI SMIS ComputerSystemPackage provider"),
      provider("cmpi::cmpicomputersystempackage")
  ]
class SMIS_ComputerSystemPackage : CIM_ComputerSystemPackage
{
};


// ====================================================================
// ProductPhysicalComponent
// ====================================================================
  [Description("CMPI SMIS ProductPhysicalComponent provider"),
      provider("cmpi::cmpiproductphysicalcomponent")
  ]
class SMIS_ProductPhysicalComponent : CIM_ProductPhysicalComponent
{
};


// ====================================================================
// ElementCapacity
// ====================================================================
  [Description("CMPI SMIS ElementCapacity provider"),
      provider("cmpi::cmpielementcapacity")
  ]
class SMIS_ElementCapacity : CIM_ElementCapacity
{
};
```

```
// ===================================================================
// CommMechanismForManager
// ===================================================================
 [Description("CMPI SMIS CommMechanismForManager provider"),
      provider("cmpi::cmpicommmechanismformanager")
 ]
class SMIS_CommMechanismForManager : CIM_CommMechanismForManager
{
};


// ===================================================================
// CIMXMLCommunicationMechanism
// ===================================================================
 [Description("CMPI SMIS CIMXMLCommunicationMechanism provider"),
      provider("cmpi::cmpicimxmlcommunicationmechanism")
 ]
class SMIS_CIMXMLCommunicationMechanism : CIM_CIMXMLCommunicationMechanism
{
};


// ===================================================================
// SystemDevice
// ===================================================================
 [Description("CMPI SMIS SystemDevice provider"),
      provider("cmpi::cmpisystemdevice")
 ]
class SMIS_SystemDevice : CIM_SystemDevice
{
};


// ===================================================================
// Namespace
// ===================================================================
 [Description("CMPI SMIS Namespace provider"),
      provider("cmpi::cmpinamespace")
 ]
class SMIS_Namespace : CIM_Namespace
```

```
{
};


// ================================================================
// NamespaceInManager
// ================================================================
 [Description("CMPI SMIS NamespaceInManager provider"),
      provider("cmpi::cmpinamespaceinmanager")
 ]
class SMIS_NamespaceInManager : CIM_NamespaceInManager
{
};


// ================================================================
// Realizes
// ================================================================
 [Description("CMPI SMIS Realizes provider"),
      provider("cmpi::cmpirealizes")
 ]
class SMIS_Realizes : CIM_Realizes
{
};


// ================================================================
// ChangerDevice
// ================================================================
 [Description("CMPI SMIS ChangerDevice provider"),
      provider("cmpi::cmpichangerdevice")
 ]
class SMIS_ChangerDevice : CIM_ChangerDevice
{
      [Description ("Generate indication")]
   uint32 GenerateIndication();
};


// ================================================================
// TapeDrive
// ================================================================
```

```
  [Description("CMPI SMIS TapeDrive provider"),
        provider("cmpi::cmpitapedrive")
  ]
class SMIS_TapeDrive : CIM_TapeDrive
{
        [Description ("Generate indication")]
    uint32 GenerateIndication();
};


// ===================================================================
// Magazine
// ===================================================================
// [Description("CMPI SMIS Magazine provider"),
//       provider("cmpi::cmpimagazine")
// ]
//class SMIS_Magazine : CIM_Magazine
//{
//};


// ===================================================================
// StorageMediaLocation
// ===================================================================
  [Description("CMPI SMIS StorageMediaLocation provider"),
        provider("cmpi::cmpistoragemedialocation")
  ]
class SMIS_StorageMediaLocation : CIM_StorageMediaLocation
{
};
// ===================================================================
// PhysicalTape
// ===================================================================
  [Description("CMPI SMIS PhysicalTape provider"),
        provider("cmpi::cmpiphysicaltape")
  ]
class SMIS_PhysicalTape : CIM_PhysicalTape
{
        [Description ("Generate indication")]
    uint32 GenerateIndication();
```

```
};


// ====================================================================
// PhysicalMediaInLocation
// ====================================================================
 [Description("CMPI SMIS PhysicalMediaInLocation"),
      provider("cmpi::cmpiphysicalmediainlocation")
 ]
class SMIS_PhysicalMediaInLocation : CIM_PhysicalMediaInLocation
{
};


// ====================================================================
// PhysicalPackage
// ====================================================================
 [Description("CMPI SMIS PhysicalPackage provider"),
      provider("cmpi::cmpiphysicalpackage")
 ]
class SMIS_PhysicalPackage : CIM_PhysicalPackage
{
};


// ====================================================================
// Container
// ====================================================================
[Description("CMPI SMIS Container provider"),
      provider("cmpi::cmpicontainer")
 ]
class SMIS_Container : CIM_Container
{
};


// ====================================================================
// PackageInChassis
// ====================================================================
/*
 [Description("CMPI SMIS PackageInChassis provider"),
      provider("cmpi::cmpipackageinchassis")
```

```
  ]
class SMIS_PackageInChassis : CIM_PackageInChassis
{
};
*/
// ===================================================================
// ProtocolControllerForUnit
// ===================================================================
 [Description("CMPI SMIS ProtocolControllerForUnit provider"),
      provider("cmpi::cmpiprotocolcontrollerforunit")
 ]
class SMIS_ProtocolControllerForUnit : CIM_ProtocolControllerForUnit
{
};


// ===================================================================
// ProtocolControllerForPort
// ===================================================================
 [Description("CMPI SMIS ProtocolControllerForPort provider"),
      provider("cmpi::cmpiprotocolcontrollerforport")
 ]
class SMIS_ProtocolControllerForPort : CIM_ProtocolControllerForPort
{
};


// ===================================================================
// PackagedComponent
// ===================================================================
 [Description("CMPI SMIS PackagedComponent provider"),
      provider("cmpi::cmpipackagedcomponent")
 ]
class SMIS_PackagedComponent : CIM_PackagedComponent
{
};


// ===================================================================
// ElementSoftwareIdentity
// ===================================================================
```

```
[Description("CMPI SMIS ElementSoftwareIdentity provider"),
     provider("cmpi::cmpielementsoftwareidentity")
 ]
class SMIS_ElementSoftwareIdentity : CIM_ElementSoftwareIdentity
{
};


// ===================================================================
// LimitedAccessPort
// ===================================================================
 [Description("CMPI SMIS LimitedAccessPort provider"),
     provider("cmpi::cmpilimitedaccessport")
 ]
class SMIS_LimitedAccessPort : CIM_LimitedAccessPort
{
};



// ===================================================================
// Location
// ===================================================================
 [Description("CMPI SMIS Location provider"),
     provider("cmpi::cmpilocation")
 ]
class SMIS_Location : CIM_Location
{
};


// ===================================================================
// PhysicalElementLocation
// ===================================================================
 [Description("CMPI SMIS PhysicalElementLocation provider"),
     provider("cmpi::cmpiphysicalelementlocation")
 ]
class SMIS_PhysicalElementLocation : CIM_PhysicalElementLocation
{
};
// ===================================================================
```

```
// RemoteServiceAccessPoint
// ==================================================================
 [Description("CMPI SMIS RemoteServiceAccessPoint provider"),
     provider("cmpi::cmpiremoteserviceaccesspoint")
 ]
class SMIS_RemoteServiceAccessPoint : CIM_RemoteServiceAccessPoint
{
};


// ==================================================================
// HostedAccessPoint
// ==================================================================
 [Description("CMPI SMIS HostedAccessPoint provider"),
     provider("cmpi::cmpihostedaccesspoint")
 ]
class SMIS_HostedAccessPoint : CIM_HostedAccessPoint
{
};


// ==================================================================
// SubProfileRequiresProfile
// ==================================================================
 [Description("CMPI SMIS SubProfileRequiresProfile provider"),
     provider("cmpi::cmpisubprofilerequiresprofile")
 ]
class SMIS_SubProfileRequiresProfile : CIM_SubProfileRequiresProfile
{
};


// ==================================================================
// ObjectManagerAdapter
// ==================================================================
 [Description("CMPI SMIS ObjectManagerAdapter provider"),
     provider("cmpi::cmpiobjectmanageradapter")
 ]
class SMIS_ObjectManagerAdapter : CIM_ObjectManagerAdapter
{
};
```

```
// ===================================================================
// CommMechanismForManagerAdapter
// ===================================================================
  [Description("CMPI SMIS CommMechanismForManagerAdapter provider"),
      provider("cmpi::cmpicommmechanismformanageradapter")
  ]
class SMIS_CommMechanismForObjectManagerAdapter :
CIM_CommMechanismForObjectManagerAdapter
{
};


// ===================================================================
// SAPAvailableForElement
// ===================================================================
  [Description("CMPI SMIS SAPAvailableForElement provider"),
      provider("cmpi::cmpisapavailableforelement")
  ]
class SMIS_SAPAvailableForElement : CIM_SAPAvailableForElement
{
};


// ===================================================================
// SCSIProtocolEndpoint
// ===================================================================
  [Description("CMPI SMIS SCSIProtocolEndpoint provider"),
      provider("cmpi::cmpiscsiprotocolendpoint")
  ]
class SMIS_SCSIProtocolEndpoint : CIM_SCSIProtocolEndpoint
{
};


// ===================================================================
// PortImplementsEndpoint
// ===================================================================
  [Description("CMPI SMIS PortImplementsEndpoint provider"),
      provider("cmpi::cmpiportimplementsendpoint")
  ]
```

```
class SMIS_PortImplementsEndpoint : CIM_PortImplementsEndpoint
{
};


// ====================================================================
//   CIM_DeviceServicesLocation
// ====================================================================
 [Description("CMPI SMIS DeviceServicesLocation provider"),
      provider("cmpi::cmpideviceserviceslocation")
 ]
class SMIS_DeviceServicesLocation : CIM_DeviceServicesLocation
{
};


// ====================================================================
// ConfigurationReportingService
// ====================================================================
 [Description("CMPI SMIS ConfigurationReportingService provider"),
      provider("cmpi::cmpiconfigurationreportingservice")
 ]
class SMIS_ConfigurationReportingService : CIM_ConfigurationReportingService
{
};


// ====================================================================
//   CIM_StorageHardwareID
// ====================================================================
 [Description("CMPI SMIS StorageHardwareID provider"),
      provider("cmpi::cmpistoragehardwareid")
 ]
class SMIS_StorageHardwareID : CIM_StorageHardwareID
{
};
// ====================================================================
//   CIM_AuthorizedPrivilege
// ====================================================================
 [Description("CMPI SMIS AuthroizedPrivilege provider"),
      provider("cmpi::cmpiauthorizedprivilege")
```

```
  ]

class SMIS_AuthorizedPrivilege : CIM_AuthorizedPrivilege

{

};



// ====================================================================

//  CIM_AuthorizedSubject

// ====================================================================

 [Description("CMPI SMIS AuthroizedSubject provider"),

      provider("cmpi::cmpiauthorizedsubject")

 ]

class SMIS_AuthorizedSubject : CIM_AuthorizedSubject

{

};



// ====================================================================

//  CIM_AuthorizedTarget

// ====================================================================

 [Description("CMPI SMIS AuthroizedTarget provider"),

      provider("cmpi::cmpiauthorizedtarget")

 ]

class SMIS_AuthorizedTarget : CIM_AuthorizedTarget

{

};



// ====================================================================

//  CIM_ProtocolControllerMaskingCapabilities

// ====================================================================

 [Description("CMPI SMIS ProtocolControllerMaskingCapabilities provider"),

      provider("cmpi::cmpiprotocolcontrollermaskingcapabilities")

 ]

class SMIS_ProtocolControllerMaskingCapabilities :
CIM_ProtocolControllerMaskingCapabilities

{

};



// ====================================================================

//  CIM_StorageClientSettingData
```

```
// ====================================================================
 [Description("CMPI SMIS StorageClientsettingData provider"),
      provider("cmpi::cmpistorageclientsettingdata")
 ]
class SMIS_StorageClientSettingData : CIM_StorageClientSettingData
{
};


// ====================================================================
//   CIM_ElementSettingData
// ====================================================================
 [Description("CMPI SMIS ElementSettingData provider"),
      provider("cmpi::cmpielementsettingdata")
 ]
class SMIS_ElementSettingData : CIM_ElementSettingData
{
};


// ====================================================================
//   CIM_ElementCapabilities
// ====================================================================
 [Description("CMPI SMIS ElementCapabilities provider"),
      provider("cmpi::cmpielementcapabilities")
 ]
class SMIS_ElementCapabilities : CIM_ElementCapabilities
{
};


// ====================================================================
// SASPort
// ====================================================================
 [Description("CMPI SMIS SASPort provider"),
  provider("cmpi::cmpisasport")
 ]
class SMIS_SASPort : CIM_SASPort
{
};
```

```
// ====================================================================
// SPIPort
// ====================================================================
 [Description("CMPI SMIS SPIPort provider"),
  provider("cmpi::cmpispiport")
 ]
class SMIS_SPIPort : CIM_SPIPort
{
};
```

B

References

This Appendix lists some web resources and popular Common Information Model (CIM)-XML tools.

Web Resources

Information about the following standards organizations are accessible on the internet:

- Website for the Storage Networking Industry Association (SNIA): http://www.snia.org
- Website for the Distributed Management Task Force (DMTF): http://www.dmtf.org

CIM-XML Tools

The following tools are available from these websites:

- Open Web Based Enterprise Management (WBEM): http://www.openwbem.com
- CimNavigator: http://www.cimnavigator.com
- OpenPegasus: http://www.openpegasus.org
- WBEM Services: http://wbemservices.sourceforge.net

C

SMI-S Common Protocol Interface Specification

This section provides a detailed list of the SMI-S feature set that is supported on PowerVault ML6000 libraries and provides updates for the SMI-S1.2.0 and other extensions.

Resources

Web

information about the following standards organizations are accessible on the Internet:

- Website for the Storage Networking Industry Association (SNIA): http://www.snia.org
- Website for the Distributed Management Task Force (DMTF): http://www.dmtf.org

Notes and Issues

Refer to SMI-S 1.2 for more information

Overview

To meet the SNIA SMI-S 1.2 CTP standards, SMI-S implementation on the Dell PowerVault ML6000 provides support for the following profiles:

- Server profile
- Profile registration subprofile
- Indication subprofile
- Access points subprofile
- Storage library profile
- Element counting subprofile
- Fibre channel target port subprofile (for libraries)
- Library Capacity subprofile

- Limited Access Port subprofile
- Software subprofile

The following subprofiles are used in modeling; however this version does not claim support of such profiles as they are not supported subprofiles of the SML profile. As of SMI-S 1.3.0, these will be supported by SML and then they will be officially supported by the implementers of this spec.

- SAS Target Port subprofile
- SPI Target Port subprofile
- Multiple Computer System subprofile
- Masking and Mapping profile

Optional support for the following will be provided in the future:

- Partitioned Library subprofile
- LibraryAlert Events/Indications for library devices
- Media Movement subprofile

Future SMI-S server releases planned to be SNIA SMI-S compliant.

Architecture

The diagram below shows the interaction between CIM Server / Providers and Software Components of the MCB/library.

Figure 2   SMI-S Implementation Architecture on the PowerVault ML6000
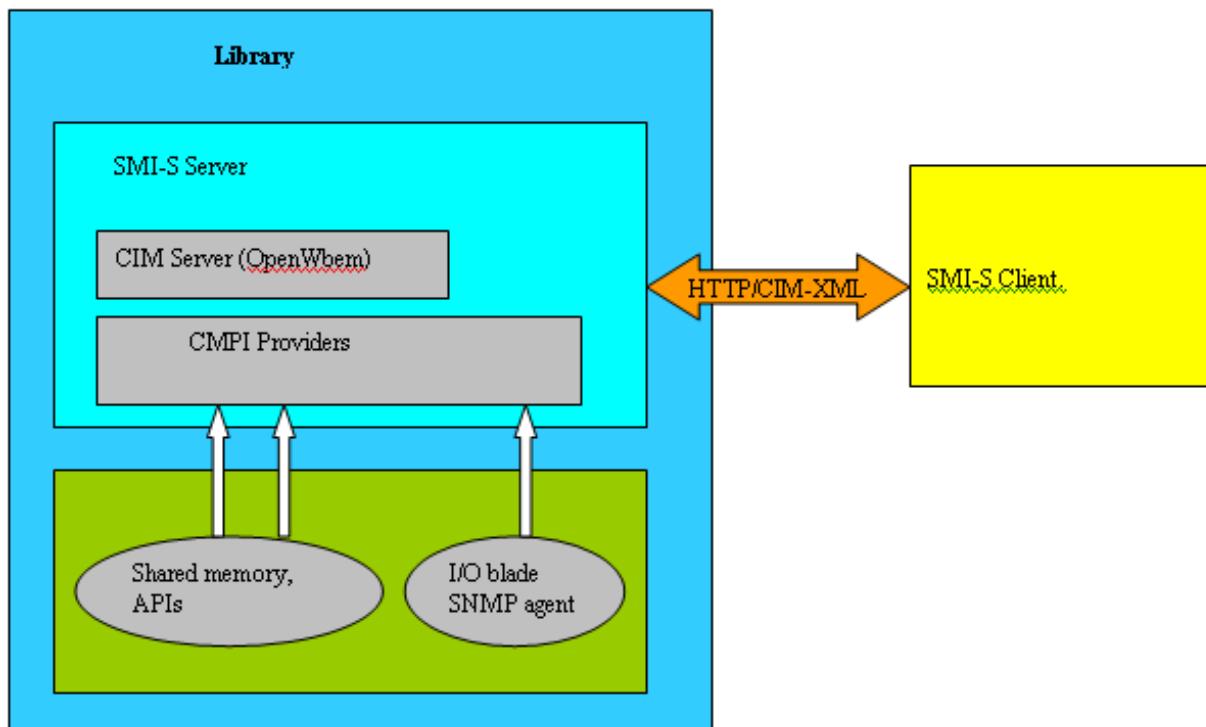
Requirements

General Purpose SMI-S Server

Overview

The General Purpose SMI-S Server role can provide a hosting environment for the plug-in instrumentation of host-based resources and management proxies for resources with remote management protocols. This plug-ins are called providers and considered sub roles of the General Purpose SMI-S Server.

SLP

The General Purpose SMI-S Server role MUST implement SLP Service agent (SA) functionality using the required template: www.dmtf.org/standards/wbem/wbem.1.0.en

Communication Interface

The General Purpose SMI-S Server role MUST implement CIM-Server functionality as specified by the CIM-XML standard.

Security Considerations

The General Purpose SMI-S Server role MUST support simple authentication scheme.

Schema Considerations

The General Purpose SMI-S Server supports CIM 2.9 schema.

Profiles

Server Profile Content

Table 3   Required Functional Profiles

Functional Group	Required
Basic Read	Yes
Basic Write	No
Instance Manipulation	No
Schema Manipulation	No
Association Traversal	Yes
Query Execution	No
Qualifier Declaration	No
Indication	Yes

Figure 3 Instance Diagram for Server Profile
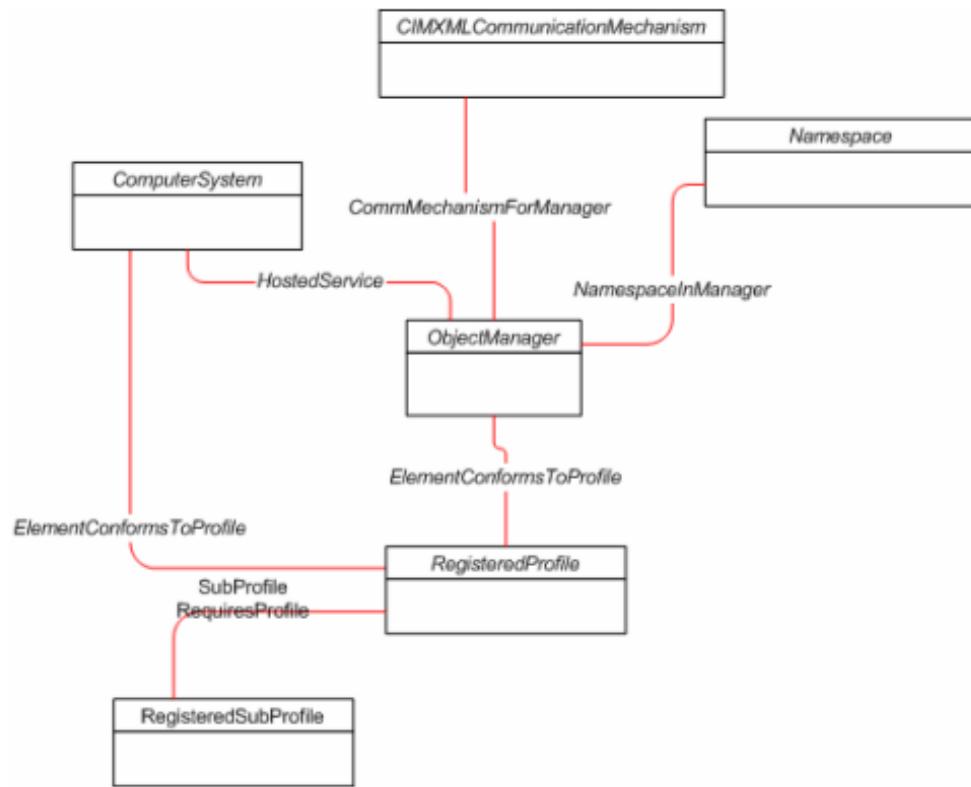
Table 4 Required Properties for ObjectManager

Property	Type	Description/Notes
SystemCreationClassName[key]	string, MaxLen(256)	
SystemName[key]	string	Host name of the library
CreationClassName[key]	string	
Name[key]	String, MaxLen(256)	
ElementName	String	
Description	String	
OperationalStatus	Uint16[]	
StatusDescriptions	String[]	MUST NOT be NULL if Other is identified in OperationalStatus
Started	Boolean	

Table 5   Required Server Level Instances of ObjectManager

Property	Instance 1
SystemCreationClassName[key]	"CIM_ComputerSystem"
SystemName[key]	<host name of the library>
CreationClassName[key]	"CIM_ObjectManager"
Name[key]	"owcimomd"
ElementName	"CIM_ObjectManager"
Description	"owcimomd"
OperationalStatus	Example : OK(2)
StatusDescriptions	OK
Started	TRUE

Table 6   Required Properties for HostedService

Property	Type	Description/Notes
Antecedent	REF	The hosting system
Dependent	REF	The service hosted on the system

Table 7   Required Server Level Instances of HostedService

Property	Instance 1
Antecedent	Reference to "CIM_ComputerSystem"
Dependent	Reference to "CIM_ObjectManager"

Table 8   Required Properties for RegisteredProfile

Property	Type	Description/Notes
InstanceID[key]	String	This is a unique value for the profile instance.
RegisteredOrganization	String, MAXLEN(256) Required	This is the official name of the organization that created the Profile. For SMI-S profiles, this would be SNIA.
OtherRegisteredOrganization	String, MAXLEN(256)	
RegisteredName	String, MAXLEN(256) Required	This is the name assigned by the organization that created the profile.
RegisteredVersion	String Required	This is the version number of the organization that defined the profile.
AdvertiseTypes	Uint16[] Required	Defines the advertisement of this profile. If the property is null then no advertisement is defined.
AdvertiseTypeDescriptions	String[]	This MUST NOT be NULL if "Other" is identified in AdvertiseType.

Table 9   Required Server Level Instances of RegisteredProfile

Property	Instance 1	Instance 2	Instance 3
InstanceID[key]	"SNIA:Server"	"SNIA:Storage Library"	"SMIS"
RegisteredOrganization	SNIA(11)	SNIA(11)	SNIA(11)
OtherRegisteredOrganization			
RegisteredName	"Server"	"Storage library"	"SMI-S"
RegisteredVersion	"1.2.0"	"1.2.0"	"1.2.0"
AdvertiseTypes	{ SLP(3) }	{ SLP(3) }	{ SLP(3) }
AdvertiseTypeDescriptions	{ "SLP" }	{ "SLP" }	{ "SLP" }

Table 10   Required Properties for RegisteredSubProfile

Property	Type	Description/Notes
InstanceID[key]	String	This is a unique value for the profile instance.
RegisteredOrganization	String, MAXLEN(256) Required	This is the official name of the organization that created the Profile. For SMI-S profiles, this would be SNIA.
OtherRegisteredOrganization	String, MAXLEN(256)	
RegisteredName	String, MAXLEN(256) Required	This is the name assigned by the organization that created the profile.
RegisteredVersion	String Required	This is the version number of the organization that defined the profile.
AdvertiseTypes	Uint16[] Required	Defines the advertisement of this profile. If the property is null then no advertisement is defined.
AdvertiseTypeDescriptions	String[]	This MUST NOT be NULL if "Other" is identified in AdvertiseType.

Table 11   Required Server Level Instances of RegisteredSubProfile

Property	Instance 1	Instance 2	Instance 3	Instance 4
InstanceID[key]	"SNIA:Storage Library:Software"	"SNIA:Storage Library:Location"	"SNIA:Storage Library:Limited Access Port Elements"	"SNIA:Storage Library:Access Points"
RegisteredOrganization	SNIA(11)	SNIA(11)	SNIA(11)	SNIA(11)
OtherRegisteredOrganization				
RegisteredName	"SNIA:Software"	"SNIA:Location"	"SNIA:Limited Access Port Elements"	"SNIA:Access Points"
RegisteredVersion	"1.2.0"	"1.2.0"	"1.2.0"	"1.2.0"
AdvertiseTypes	{ SLP(3) }	{ SLP(3) }	{ SLP(3) }	{ SLP(3) }
AdvertiseTypeDescriptions	{ "SLP" }	{ "SLP" }	{ "SLP" }	{ "SLP" }

Property	Instance 5	Instance 6	Instance 7	Instance 8
InstanceID[key]	"SNIA:Storage Library:Storage Library Capacity"	"SNIA:Storage Library:FC Target Ports"	"SNIA:Storage Library:Element Counting"	"SNIA:Server:Indication"
RegisteredOrganization	SNIA(11)	SNIA(11)	SNIA(11)	SNIA(11)
OtherRegisteredOrganization				
RegisteredName	"SNIA:Storage Library:Capacity"	"SNIA:FC Target Ports"	"SNIA:Element Counting"	"SNIA:Indication"
RegisteredVersion	"1.2.0"	"1.2.0"	"1.2.0"	"1.2.0"
AdvertiseTypes	{ SLP(3) }	{ SLP(3) }	{ SLP(3) }	{ SLP(3) }
AdvertiseTypeDescriptions	{ "SLP" }	{ "SLP" }	{ "SLP" }	{ "SLP" }

Property	Instance 9	Instance 10
InstanceID[key]	"SNIA:Server:Object Manager Adapter"	"SNIA:Server:Profile Registration"
RegisteredOrganization	SNIA(11	SNIA(11
OtherRegisteredOrganization		
RegisteredName	"SNIA:Object Manager Adapter"	"SNIA:Profile Registration"
RegisteredVersion	"1.2.0"	"1.2.0"
AdvertiseTypes	{ SLP(3) }	{ SLP(3) }
AdvertiseTypeDescriptions	{ "SLP" }	{ "SLP" }

Table 12   Required Properties for SubProfileRequiresProfile

Property	Type	Description/Notes
Antecedent	REF	The RegisteredProfile that is referenced/required by the subprofile.
Dependent	REF	A RegisteredSubProfile that requires a scoping profile, for context..

Table 13   Required Server Level Instances of SubProfileRequiresProfile

Property	Instance 1
Antecedent	CIM_RegisteredProfile
Dependent	CIM_RegisteredSubProfile

Table 14   Required Properties for ElementConformsToProfile

Property	Type	Description/Notes
ConformantStandard	REF	The RegisteredProfile to which the ManagedElement conforms.
ManagedElement	REF	The ManagedElement that conforms to the RegisteredProfile.

Table 15   Required Server Level Instances of ElementConformsToProfile

Property	Instance 1	Instance 2	Instance 3
ConformantStandard	Server	Library	SNIA:SMIS Registered
ManagedElement	CIM_ObjectManager	CIM_ComputerSystem	CIM_RegisteredProfile

Table 16   Required Properties for Namespace

Property	Type	Description/Notes
SystemCreationClassName[key]	String, MAXLEN(256)	
SystemName[key]	String, MAXLEN(256)	Host name of the library.
ObjectManagerCreationClassName	String, MAXLEN(256)	
ObjectManagerName	String, MAXLEN(256)	
CreationClassName	String, MAXLEN(256)	
Name	String, MAXLEN(256)	
ClassInfo	Uint16 Required	

Table 17   Required Server Level Instances of Namespace

Property	Instance 1
SystemCreationClassName[key]	"CIM_System"
SystemName[key]	<host name of the library>
ObjectManagerCreationClassName	"CIM_ObjectManager"
ObjectManagerName	"owcimomd"
CreationClassName	"CIM_Namespace"
Name	"root/cimv2"
ClassInfo	CIM 2.9 (11)

Table 18   Required Properties for NamespaceInManager

Property	Type	Description/Notes
Antecedent	REF	The ObjectManager containing a Namespace.
Dependent	REF	The Namespace in an ObjectManager.

Table 19   Required Server Level Instances of NamespaceInManager

Property	Instance 1
Antecedent	"CIM_ObjectManager"
Dependent	"CIM_Namespace"

Table 20   Required Properties for CIMXMLCommunicationMechanism

Property	Type	Description/Notes
ElementName	String	
CreationClassName[key]	String, MaxLen(256)	
SystemCreationClassName[key]	String, MaxLen(256)	
SystemName[key]	String	Host name of the library.
Name[key]	String, MaxLen(256)	
OperationalStatus	Uint16[	
StatusDescriptions	String[]	MUST NOT be NULL if "Other" is identified in OperationalStatus.
CommunicationMechanism	Uint16, Required	Must be 2
OtherCommunicationMechanismDescription	String	MUST NOT be NULL if "Other" is identified in CommunicationMechanism.
FunctionalProfilesSupported	Uint16[], Required	
FunctionalProfileDescriptions	String[]	
MultipleOperationsSupport	Boolean, Required	
AuthenticationMechanismsSupported	Uint16[], Required	
AuthenticationMechanismDescriptions	String[]	MUST NOT be NULL if "Other" is identified in AuthenticationMechanismSupported.
CIMXMLProtocolVersion	Uint16, Required	1.0(1)
CIMValidated	Boolean, Required	
Version	string	Must be 1.0, 1.1, or 1.2

Table 21   Required Server Level Instances for CIMXMLCommunicationMechanism

Property	Instance 1
ElementName	"CIM_CIMXMLCommunicationMechanism"
CreationClassName[key]	"CIM_CIMXMLCommunicationMechanism"
SystemCreationClassName[key]	"CIM_ObjectManager"
SystemName[key]	<host name of the library>
Name[key]	"CIM_CIMXMLCommunicationMechanism"
OperationalStatus	OK(2)
StatusDescriptions	"OK"
CommunicationMechanism	CIM-XML (2)
OtherCommunicationMechanismDescription	
FunctionalProfilesSupported	{ Basic Read (2), Association Traversal (7), Indications (9) }
FunctionalProfileDescriptions	{ "Basic Read", "Association Traversal ", "Indications" }
MultipleOperationsSupport	FALSE
AuthenticationMechanismsSupported	Basic(3)
AuthenticationMechanismDescriptions	"Basic"
CIMXMLProtocolVersion	1.0(1)
CIMValidated	TRUE
Version	1.2

Table 22   Required Properties for CommMechanismForManager

Property	Type
ObjectManager	REF
ObjectManagerCommunicationMechanism	REF

Table 23   Required Server Level Instances of CommMechanismForManager

Property	Instance 1
ObjectManager	Reference to "CIM_ObjectManager"
ObjectManagerCommunicationMechanism	Reference to "CIM_CIMXMLCommunicationMechanism"

Table 24   Required Properties for ComputerSystem

Property	Type	Description/Notes
CreationClassName[key]	String	Name of Class
Name[key]	String	For component computer systems, the provider MUST provide a unique name using one of the NameFormats.
Dedicated	Uint16[]	Container type (for example, SML).
OperationalStatus	Uint16[]	Container status.
ElementName	String	User-friendly name.
NameFormat	String	"HID" or "IP"
PrimaryOwnerName	String	
PrimaryOwnerContact	String	
Caption	String	
Description	String	

Table 25   Required Properties for ComputerSystem

Property	Instance 1
CreationClassName[key]	"CIM_ComputerSystem"
Name[key]	IP Example: 172.16.41.26
Dedicated	Storage (4).
OperationalStatus	Example: ok(2)
ElementName	"CIM_ComputerSystem"
NameFormat	"IP"
PrimaryOwnerName	"DELL"
PrimaryOwnerContact	" "
Caption	IP
Description	"Quantum WBEM Server"

Storage Library Profile Content

Figure 4   ComputerSystem-Centric Instance Diagram of StorageLibrary Profile
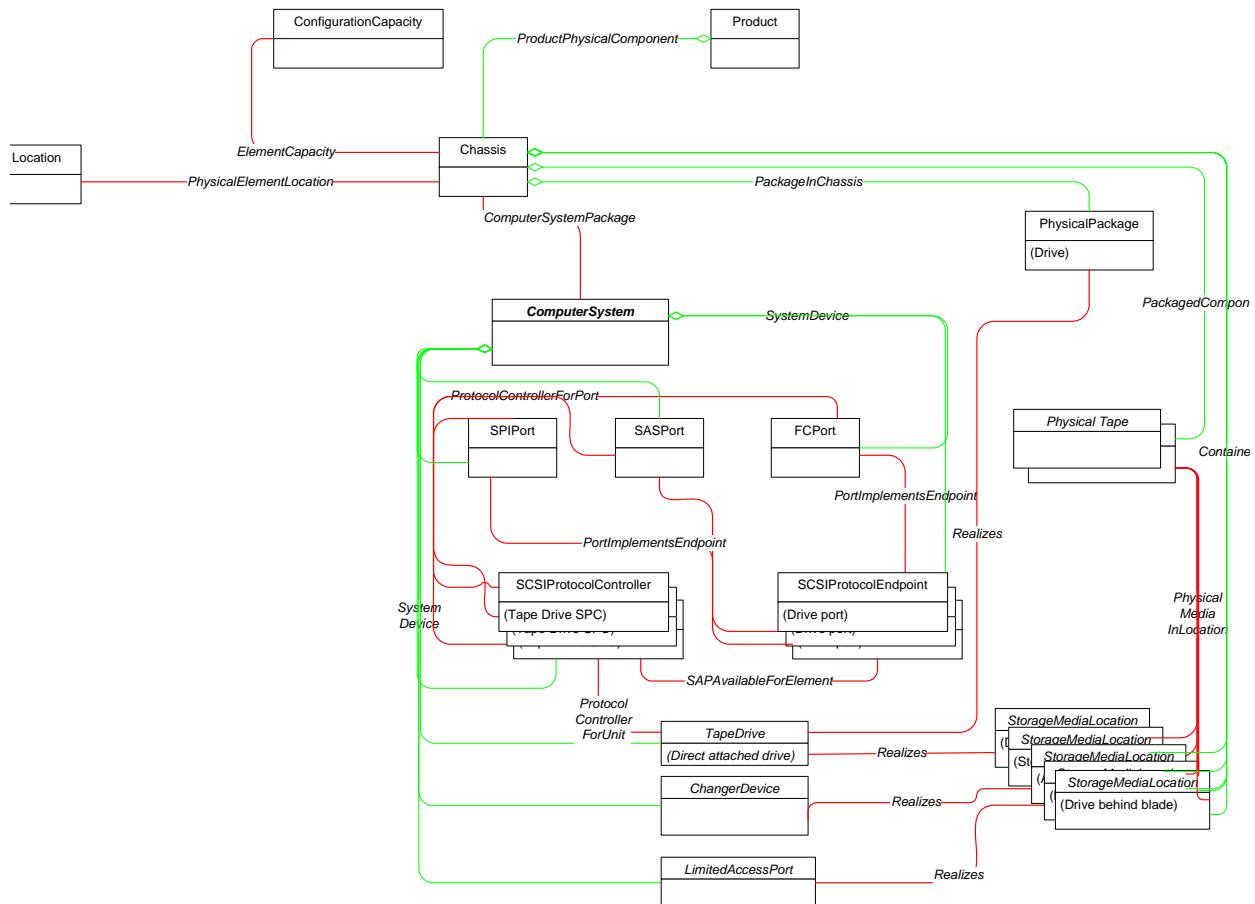
Table 26   Required Container Level Instances of ComputerSystem

Property	Instance 1
CreationClassName[key]	"CIM_ComputerSystem"
Name[key]	Vendor + Product + Serial Example: DELL ML6000 SNA12345678
Dedicated	sml (22)
OperationalStatus	Example : ok(2)
ElementName	"CIM_ComputerSystem"
NameFormat	"HID"
Automated	True
PrimaryOwnerName	Owner Name from contact info
PrimaryOwnerContact	Owner Phone + Owner E-mail from contact info
Caption	ComputerSystem
Description	Tape Library Backup Unit

Table 27   Required Properties for Chassis

Property	Type	Description/Notes
Tag[key]	String	Module (CM/EM) location number (0, -1, 1, etc.).
CreationClassName[key]	String	Indicates the name of the class or subclass used in the creation of an instance.
Serial Number	String	Module serial number
Model	String	Module display name (ControlModuleFrame, ExpansionModuleFrame).
Description	String	Module display name/model description.
Manufacturer	String	Module manufacturer.
LockPresent	Boolean	Boolean indicating whether the module is protected with a lock.
SecurityBreach	Uint16	Boolean indicating whether the aggregated door status of the library (any door) is open (breach successful) or closed (no breach).
IsLocked	Boolean	Boolean indicating that the module is currently locked.
ElementName	string	

Table 28   Required Container Level Instances of Chassis

Property	Instance (one per module)
Tag[key]	Module (CM/EM) location number (0, -1, 1, etc.).
CreationClassName[key]	"CIM_Chassis"
Serial Number	Module serial number
Model	"ControlModuleFrame" or "ExpansionModuleXU" module where XU is the module size
Manufacturer	"DELL"
LockPresent	False
SecurityBreach	Breach Successful if any door open or No Breach(3) if all doors closed.
IsLocked	False
ElementName	"Chassis X" where X is module number.

Table 29   Required Properties for Location

Property	Type	Description/Notes
Name[key]	String	Position is a free-form string indicating the placement of a PhysicalElement.
PhysicalPosition[key]	String	Address is a free-form string indicating a street, building, or other type of address for the PhysicalElement's Location.
Caption	String	

Table 30   Required Container Level Instances of Location

Property	Instance
Name[key]	IP
PhysicalPosition[key]	Location from contact info
Caption	IP

Table 31   Required Properties for PhysicalElementLocation

Property	Type	Description/Notes
Element	REF	The reference to the chassis.
PhysicalLocation	REF	The reference to the location.

Table 32   Required Server Level Instances of PhysicalElementLocation

Property	Instance 1
Element	CIM_Chassis
PhysicalLocation	CIM_Location

Figure 5   Instance Diagram of ComputerSystemPackage

IdentifyingNumber is the library serial number

SerialNumber is the module serial number, not the library serial number.

Note:
The Dell PowerVault ML6000 is modeled to have a chassis for each module.
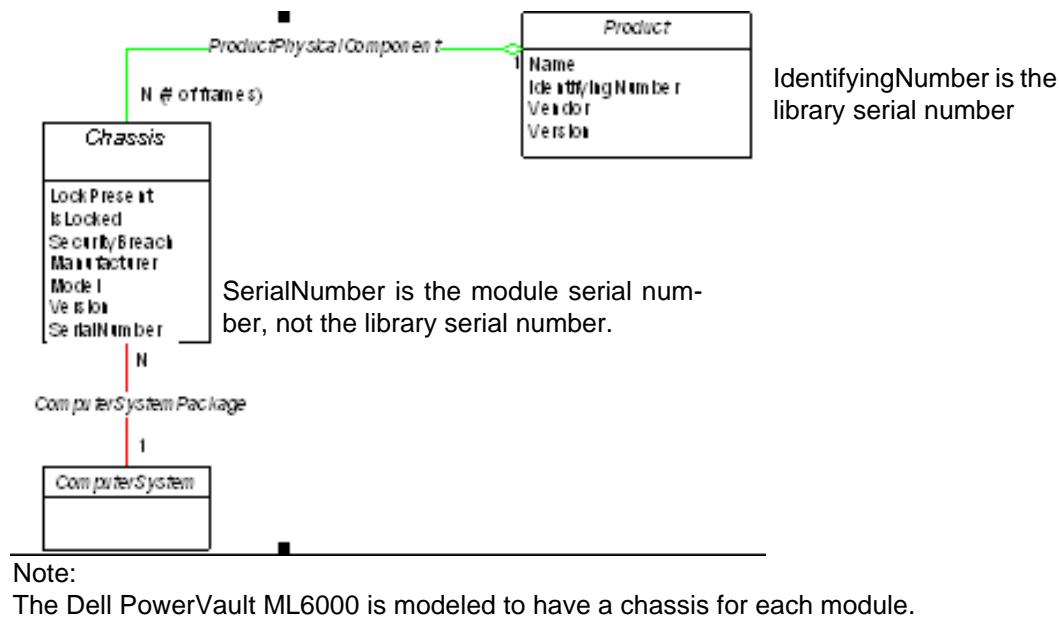
Table 33   Required Properties for ComputerSystemPackage

Property	Type	Description/Notes
Antecedent	REF	The reference to the PhysicalPackage(s) that realize a UnitaryComputerSystem.
Dependent	REF	The reference to the UnitaryComputerSystem.
PlatformGUID	PlatformGUID	A globally unique identifier for the system's package.

Table 34   Required Server Level Instances of ComputerSystemPackage

Property	Instance 1
Antecedent	CIM_Chassis
Dependent	CIM_ComputerSystem
PlatformGUID	Library-wide unique ID

Table 35 Required Properties for Product

Property	Type	Description/Notes
Name[key]	String, MaxLen(256)	Commonly used product name.
IdentifyingNumber[key]	String, MaxLen(64)	Product identification such as a serial number on software, a die number on a hardware chip, or a project number.
Vendor[key]	String, MaxLen(256)	The name of the product supplier, or entity selling the product.
Version[key]	String, MaxLen(64)	Firmware version information.
ElementName	String	User-friendly name. Suggested use is vendor, version, and product name.

Table 36 Required Container Level Instances of Product

Property	Instance 1
Name[key]	<company> <product> (for example, DELL PowerVault ML6000)
IdentifyingNumber[key]	<serial number> (for example: A0C0085826)
Vendor[key]	DELL
Version[key]	Example: 200A.TS01003
ElementName	SMIS_Product

Table 37 Required Properties for ProductPhysicalComponent

Property	Type	Description/Notes
GroupProduct	REF	The product.
PartComponent	REF	The physical element which is part of the product.

Table 38 Required Server Level Instances of ProductPhysicalComponent

Property	Instance 1
Product	CIM_Product
Component	CIM_Chassis

Table 39 Required Properties for SoftwareIdentity

Property	Type	Description/Notes
InstanceID[key]	String	Instance name.
VersionString	String	Version.
Manufacturer	String	Manufacturer of this software.

Table 40 Required Container Level Instances of SoftwareIdentity

Property	Firmware Instance (one for system)	Tape Drive Instances (per drive)	Providers Instance (one for system)	Changer Device Instance (one for system)
InstanceID[key]	"PowerVault ML6000 Firmware"	"TapeDrive:<drive serial>:<Vendor> <FWR>"	"Providers Revision"	"Changer Device:<Version>"
VersionString	Example : "1.2.3"	Drive <FWR> Example : "6B20"	Same as Firmware	Same as Firmware
Manufacturer	DELL	Example: IBM or HP	DELL	DELL
Caption	"S101Core"	"<serial>:<Vendor> <FWR>"	"Providers Revision"	"Providers Revision"

Table 41 Required Properties for InstalledSoftwareIdentity

Property	Type	Description/Notes
System	REF	System on which software is installed.
InstalledSoftware	REF	Software identity that is installed.

Table 42   Required Server Level Instances of InstalledSoftwareIdentity

Property	Library System Instance	Drive Instances
System	CIM_ComputerSystem for Library (HID)	CIM_TapeDrive for drive.
InstalledSoftware	CIM_SoftwareIdentity "PowerVault ML6000"	CIM_SoftwareIdentity for corresponding drive software instance.

Property	Server Instance (one for System)	Changer Instances (one for each changer)
System	CIM_ComputerSystem for Server	CIM_ChangerDevice.
InstalledSoftware	CIM_SoftwareIdentity "Providers Revision"	CIM_SoftwareIdentity (the single changer device software instance).

Table 43   Required Properties for ElementCapacity

Property	Type	Description/Notes
Capacity	REF	Physical capacity describes the minimum and maximum requirements.
Element	REF	Physical element being described.

Table 44   Required Server Level Instances of ElementCapacity

Property	Instance 1
Capacity	CIM_ConfigurationCapacity
Element	CIM_Chassis

Table 45   Required Properties for ConfigurationCapacity

Method	Type	Description/Notes
Name[key]	String	Instance Name.
ObjectType[key]	Uint16	Object type.
MaximumCapacity	Uint64	Maximum capacity. Should not be compressed capacity.
OtherTypeDescription	String	A string describing the object type - used when the ObjectType property is set to "Other(0)".

Table 46 Required Container Level Instances of ConfigurationCapacity

Property	Instance 1	Instance 2	Instance 3
Name[key]	"Tape Library Maximum Slots"	"Tape Library Number of Tapes"	"Tape Library Number of Tape Drives"
ObjectType[key]	StorageMediaLocation Slots (8)	Other (0)	MediaAccessDevices (Drives) (7)
MaximumCapacity	Depends on library configuration	Same as instance 1	Depends on library configuration
OtherTypeDescription	"Estimated number of maximum slots"	"Estimated number of maximum tapes"	"Estimated number of maximum tape drives"

Property	Instance 4	Instance 5	Instance 6
Name[key]	"Tape Library Total RAW Capacity"	"Tape Library Total RAW Free Capacity"	"Tape Library Total RAW Used Capacity"
ObjectType[key]	Other (0)	Other (0)	Other (0)
MaximumCapacity	Example: 400000 in GB	Example: 400000 in GB	Example: 0 in GB
OtherTypeDescription	"Estimated total raw capacity"	"Estimated total raw Free capacity"	"Estimated total raw used capacity"

Note: Total RAW capacity is computed as "number of slots" multiplied by "tape size". "Tape size" is the size of the tape type with maximum capacity. Example, If the library supports LTO-1 and LTO-2, tape size would be the RAW capacity of the LTO-2 tape.

Table 47 Required Properties for SystemDevice

Property	Type	Description/Notes
GroupComponent	REF	Parent system in the association.
PartComponent	REF	Logical device that is component of the association product.

Table 48　Required Library Level Instances of SystemDevice

Property	FCPort Instances (one for each FC Port)	SASPort Instances (one for each SAS Port)	SPI Port Instances (one for each FC Port)
GroupComponent	CIM_ComputerSystem*	CIM_ComputerSystem	CIM_ComputerSystem
PartComponent	CIM_FCPort	CIM_SASPort	CIM_SPIPort

* For FCPort, SCSIProtocolEndpoint and SCSIProtocolController, they may have a SystemDevice association to the
　　ComputerSystem of the blade, not the library (for example, if the object represents a blade FC port). See Fibre Channel
　　Blade Support on page 77 for more detail on blade support.

Table 49　Required Library Level Instances of SystemDevice

Property	SPC Instances (one for each SPC)SPE	Instances (one for each SPE)	IE Instances (one for each IE slot)
GroupComponent	CIM_ComputerSystem*	CIM_ComputerSystem*	CIM_ComputerSystem
PartComponent	CIM_SCSIProtocolController	CIM_SCSIProtocolEndpoint	CIM_LimitedAccessPort

Property	Drive Instances (one for each tape drive)	Changer Instances (one for each partition + physical)
GroupComponent	CIM_ComputerSystem	CIM_ComputerSystem
PartComponent	CIM_TapeDrive	CIM_ChangerDevice

* For FCPort, SCSIProtocolEndpoint and SCSIProtocolController, they may have a SystemDevice association to the
　　ComputerSystem of the blade, not the library (for example, if the object represents a blade FC port). See Fibre Channel
　　Blade Support on page 77 for more detail on blade support.

Table 50　Required Properties for Realizes

Property	Type 1	Description/Notes
Antecedent	REF	The physical component that implements the device.
Dependent	REF	The LogicalDevice.

Table 51   Required Library Level instances of Realizes

Property	Accessor Instances (one for each partition + physical)	Drive Instances (one for each drive)	IE Instances (one for each IE slot)
Antecedent	CIM_StorageMediaLocation	CIM_StorageMediaLocation	CIM_StorageMediaLocation
Dependent	CIM_ChangerDevice	CIM_TapeDrive	CIM_LimitedAccessPort

Property	Changer Package (one for system)	Drive Package Instances (one for each drive)	Chassis IE (one for each IE slot)
Antecedent	CIM_PhysicalPackage	CIM_PhysicalPackage	CIM_Chassis
Dependent	CIM_ChangerDevice (only the physical changer)	CIM_TapeDrive	CIM_LimitedAccessPort

Table 52   Required Properties for FCPort

Property	Type	Description/Notes
SystemCreationClassName[key]	String	
SystemName[key]	String	Host name of the library.
CreationClassName[key]	String	
DeviceID[key]	String	
PortNumber	Uint16	
PermanentAddress	String	Port WWN.
ElementName	String	
OperationalStatus	Uint16	Port Status.
NetworkAddress	String	FCID.
PortType	Uint16	Port Type.
ActiveFC4Types	Uint16	Supported FC types.
OtherNetworkPortType	String	Describes the type of module, when PortType is set to 1 ("Other").
LinkTechnology	Uint16	
OtherLinkTechnology	String	A string value describing LinkTechnology when it is set to 1, "Other".
ActiveCOS	Uint16	An array of integers indicating the class of service that is active.
UsageRestriction	Uint16	LogicalPort may be identifiable as a front end or back end port.

Table 53   Required Direct Attached Drive Front-End Ports Instances of FCPort

Property	Description/Notes
SystemCreationClassName[key]	"CIM_ComputerSystem"
SystemName[key]	<HID (Vendor + Product + Serial of the library>
CreationClassName[key]	"CIM_FCPort"
DeviceID[key]	"D:<GUID>:<Location>:<Serial>:<WWPN>"
PortNumber	Example: 1
PermanentAddress	WW Port Name Format is 16 octets 1234567887654321
ElementName	"FC Port"
OperationalStatus	Example : ok(2)
NetworkAddress	0
PortType	Example: N(10)
ActiveFC4Types	SCSI-FCP(8)
OtherNetworkPortType	"" since portType is N(10)
LinkTechnology	Example: FC(4)
OtherLinkTechnology	""
ActiveCOS	3
UsageRestriction	Front-end only(2)

Table 54   Required Properties for SASPort

Property	Type	Description/Notes
SystemCreationClassName[key]	String	
SystemName[key]	String	Host name of the library.
CreationClassName[key]	String	
DeviceID[key]	String	
PortNumber	Uint16	
PermanentAddress	String	SAS UID.
ElementName	String	
OperationalStatus	Uint16[]	Port status.

Table 55 Required Library Level Instances of SASPort

Property	Value (one for each SAS drive)
SystemCreationClassName[key]	CIM_ComputerSystem
SystemName[key]	<Vid> <Pid> <Serial>
CreationClassName[key]	CIM_SCSIProtocolController
DeviceID[key]	<DriveGUID>
PermanentAddress	SAS UID
Description	"SCSI port on drive <serial>"
OperationalStatus	Status of drive
Caption	<drive GUID>

Table 56 Required Properties for SPIPort

Property	Type	Description/Notes
SystemCreationClassName[key]	String	
SystemName[key]	String	Host name of the library.
CreationClassName[key]	String	
DeviceID[key]	String	
PortNumber	Uint16	
PermanentAddress	String	SAS UID.
NetworkAddress	String Array	
ElementName	String	
OperationalStatus	Uint16[]	Port status.

Table 57   Required Library Level Instances of SPIPort

Property	Value
SystemCreationClassName[key]	CIM_ComputerSystem
SystemName[key]	<Vid> <Pid> <Serial>
CreationClassName[key]	CIM_SPIPort
DeviceID[key]	<drive GUID>
PermanentAddress	""
NetworkAddresses	{<target ID>}
Description	"SCSI port on drive <serial>"
OperationalStatus	Status of drive
MaxDataWidth	Maximum bus width
MaxTransferRate	Maximum data transfer rate
Caption	<drive GUID>

Table 58   Required Properties for ChangerDevice

Property	Type	Description/Notes
SystemCreationClassName[key]	string, MaxLen(256)	The scoping system's CreationClassName.
SystemName[key]	string	The scoping system's name.
CreationClassName[key]	string	Indicates the name of the class or subclass used in the creation of an instance.
DeviceID[key]	string , MaxLen(256)	
MediaFlipSupported	Boolean	
ElementName	String	User-friendly name.
OperationalStatus[valuemap]	Uint16[]	
Caption	String	
Description	String	
Availability[valuemap]	Uint16	

Table 59 Required Server Level Instances of ChangerDevice

Property	Physical Library Instance 1	Partition Instances (one for each partition)
SystemCreationClassName[key]	"CIM_ComputerSystem"	"CIM_ComputerSystem"
SystemName[key]	<HID of the library>	<HID of the library>
CreationClassName[key]	"CIM_Computersystem"	"CIM_Computersystem"
DeviceID[key]	"<GUID>:<Serial>:<Version>"	"<GUID>:<Serial>:<Version>"
MediaFlipSupported	False	False
ElementName	"SMIS_ChangerDevice"	"SMIS_ChangerDevice"
OperationalStatus[valuemap]	Robotics RAS status	{OK(2)}
Caption	Physical Library	Library name (library_a or library_b, etc.)
Description	Physical Medium Changer	"Partition <HID>"
Availability[valuemap]	Running/Full Power(3) or Offline (8)	Running/Full Power(3) or Offline (8)

Table 60 Required Properties for ProtocolControllerForUnit

Property	Type	Description/Notes
Antecedent	REF	The ProtocolController.
Dependent	REF	The controlled device.
DeviceNumber	string	Address of associated device in context of the antecedent ProtocolController. Formatted as unseparated uppercase hexadecimal digits, with no leading "0x".

Table 61 Required Server Level Instances of ProtocolControllerForUnit

Property	Drive Instances (one for each direct-attached drive)	Changer Instance (one for each changer which has a control-path drive)
Antecedent	CIM_SCSIProtocolController (corresponding to the drive)	CIM_SCSIProtocolController (corresponding to control path drive)
Dependent	CIM_TapeDrive	CIM_ChangerDevice
DeviceNumber	"0"	"1"

Table 62   Required Properties for SCSIProtocolController

Property	Type	Description/Notes	Comments/ Remarks
SystemCreationClassName[key]	string, MaxLen(256)	The scoping System's CreationClassName.	
SystemName[key]	string	The scoping System's Name.	Library's host name.
CreationClassName[key]	string	Indicates the name of the class or subclass used in the creation of an instance.	
DeviceID[key]	string , MaxLen(256)	Opaque	
MaxUnitsControlled	Uint32	Maximum number of units controlled by this ProtocolController (optional)	
ElementName	String	User friendly name	
OperationalStatus[valuemap]	Uint16[]	Status of device (optional)	
Caption	String	(optional)	
Description	String	(optional)	
OtherIdentifyingInfo]	String[]	Captures additional data, beyond System Name information, that could be used to identify a LogicalDevice.	WWN

Table 63   Required Server Level Instances of SCSIProtocolController

Property	Drive Instances (one for each direct-attached drive)
SystemCreationClassName[key]	"CIM_ SCSIProtocolController"
SystemName[key]	IP
CreationClassName[key]	"CIM_ComputerSystem"
DeviceID[key]	"D:<GUID>:<serial>:<vendor> <fwr>:<SPE DeviceID>"
MaxUnitsControlled	2048
ElementName	"SMIS_SCSIProtocolController"
OperationalStatus[valuemap]	{Unknown(0)}
Caption	Same as DeviceID
Description	"Tape Drive <Type>" where <Type> is SCSI/FC/SAS
OtherIdentifyingInfo	{<serial>}
IdentifyingInfoDescriptions	{"Serial Number"}

Table 64   Required Properties of SCSIProtocolEndpoint

Property	Type	Description/Notes	Comments/ Remarks
SystemCreationClassName[key]	string, MaxLen(256)	The scoping System's CreationClassName.	
SystemName[key]	string	The scoping System's Name.	Library's host name.
CreationClassName[key]	string	Indicates the name of the class or subclass used in the creation of an instance.	
DeviceID[key]	string , MaxLen(256)	Opaque	
MaxUnitsControlled	Uint32	Maximum number of units controlled by this ProtocolController (optional)	
ElementName	String	User-friendly name	
OperationalStatus[valuemap]	Uint16[]	Status of device (optional)	
Caption	String	(optional)	
Description	String	(optional)	
ConnectionType [valuemap]	Uint16	Fibre/ISDN/…	
Role [valuemap]	Uint16	Target/Initiator…	
ProtocolIFType [valuemap]	Uint16	Protocol Interface.	

Table 65 Required Library Instances of SCSIProtocolEndpoint

Property	Drive Instances (one for each direct-attached drive)
SystemCreationClassName[key]	"CIM_ComputerSystem"
SystemName[key]	HID
CreationClassName[key]	"CIM_SCSIProtocolEndpoint"
DeviceID[key]	"<WWPN>"
ConnectionType	Fibre Channel (2)
Role	Target (3)
OperationalStatus[valuemap]	{Unknown(0)}
ProtocolIFType	Fibre Channel (56)
Description	"SMIS_SCSIProtocolEndpoint"

Table 66 Required Properties of PortImplementsEndpoint

Property	Type	Description/Notes
Antecedent	REF	The LogicalPort.
Dependdent	REF	The ProtocolEndpoint.

Table 67 Required Library Instances of PortImplementsEndpoint

Property	FCPort Instances (one for each direct-attached FC drive)	SAS Instances (one for each SAS drive)	SPI Instances (one for each SCSI drive)
Antecedent	CIM_FCPort	CIM_SASPort.	CIM_SPIPort
Dependent	CIM_SCSIProtocolEndpoint	CIM_SCSIProtocolEndpoint.	CIM_SCSIProtocolEndpoint

Table 68 Required Properties of SAPAvailableForElement

Property	Type	Description/Notes
AvailableSAP	REF	The ServiceAccessPoint.
ManagedElement	REF	The ManagedElement that is accessible by the SAP.

Table 69 Required Library Instances of SAPAvailableForElement

Property	Instance (one for each **SCSIProtocolController**)
AvailableSAP	CIM_SCSIProtocolEndpoint
ManagedElement	CIM_SCSIProtocolController

Table 70 Required Properties for ElementSoftwareIdentity

Property	Type	Description/Notes
Antecedent	REF	The SoftwareIdentity.
Dependdent	REF	The ManagedElement that requires or uses the software.

Table 71 Required Library/Server Level Instances of ElementSoftwareIdentity

Property	Drive Instance (one for each drive)	Changer Instance (one for each partition + physical)	Profile Instances (one for each Registered Profile or Subprofile)
Antecedent	CIM_SoftwareIdentity (corresponding to drive)	CIM_SoftwareIdentity (the changer software instance).	CIM_SoftwareIdentity ("Providers Revision")
Dependdent	CIM_TapeDrive	CIM_ChangerDevice.	CIM_RegisteredProfile or CIM_RegisteredSubprofile

Table 72   Required Properties for TapeDrive

Property	Type	Description/Notes	Comments/Remarks
SystemCreationClassName[key]	string, MaxLen(256)	The scoping System's CreationClassName.	
SystemName[key]	string	The scoping System's Name.	Library's host name.
CreationClassName[key]	string	Indicates the name of the class or subclass used in the creation of an instance.	
DeviceID[key]	string , MaxLen(256)	Opaque	
Availability	Uint16		
NeedsCleaning	Boolean	Boolean indicating that the MediaAccessDevice needs cleaning. Whether manual or automatic cleaning is possible is indicated in the Capabilities array property.	
PowerOnHours	Uint64	counter, units("hours")	
TotalPowerOnHours	Uint64	counter, units("hours")	
OperationalStatus [valuemap]	Uint16[]	Status of device (optional)	
MountCount	Uint64	For a MediaAccessDevice that supports removable media, the number of times that media have been mounted for data transfer or to clean the device. For devices accessing nonremovable media, such as hard disks, this property is not applicable and should be set to 0.	

Table 73   Required Server Level Instances of TapeDrive

Property	Type
SystemCreationClassName[key]	"CIM_TapeDrive"
SystemName[key]	<IP>
CreationClassName[key]	"CIM_ComputerSystem"
DeviceID[key]	"<GUID>:<serial>:<Vendor> <version>"
Availability	Running/Full Power(3) or Offline (8)
NeedsCleaning	False
PowerOnHours	Hours since last drive power
TotalPowerOnHours	Total hours the drive has been on
OperationalStatus [valuemap]	{OK(2)}
MountCount	Number of times the drive has been mounted

Table 74   Required Properties for PackageInChassis

Property	Type	Description/Notes
GroupComponent	REF	The chassis that contains other PhysicalPackages.
PartComponent	REF	The PhysicalPackage which is contained in the chassis.

Table 75   Required Server Level Instances of PackageInChassis

Property	Instance 1
GroupComponent	CIM_Chassis
PartComponent	CIM_PhysicalPackage

Table 76   Required Properties for PhysicalPackage

Property	Type	Description/Notes
CreationClassName[key]	String	
Tag[key]	String	Type + SerialNumber
Manufacturer	String	Manufacturer
Model	String	Model
Caption	String	

Table 77   Required Library Level Instances of PhysicalPackage

Property	Tape Drive Instances (one for each drive)	Control Module Instances (one for the system)
CreationClassName[key]	"CIM_PhysicalPackage"	"CIM_PhysicalPackage"
Tag[key]	"<GUID>:<Location>"	"<Physical Changer GUID>:<CM Location>"
Manufacturer	Vendor (example: "IBM")	Branding
Model	Product ID (example: "ULTRIUM-TD3")	"ControlModule5U"
Caption	<Serial>	Same as Tag
Serial Number	<Serial>	<Serial>

Table 78   Required Properties for PhysicalTape

Property	Type	Description/Notes
Tag[key]	String, MaxLen(256)	An arbitrary string that uniquely identifies the Physical Element.
CreationClassName[key]	String	The name of the concrete subclass.
Capacity	Uint64	Units ("bytes").
MediaType	Uint16	
MediaDescription	String	Additional detail related to the MediaType enumeration.
CleanerMedia	Boolean	
DualSided	Boolean	
PhysicalLabels	string[]	One or more strings on 'labels' on the PhysicalMedia.

Table 79   Required Server Level Instances of PhysicalTape

Property	Instance (one for each present cartridge)
Tag[key]	"<Changer GUID>:<Tape GUID>:0"
CreationClassName[key]	"SMIS_PhysicalTape"
Capacity	Size of data on tape12345
MediaType	LTO Ultrium(7)
MediaDescription	"LTO 1"
CleanerMedia	Boolean
DualSided	False
PhysicalLabels	Barcode {"12345"}]
EncryptionState	0 Unknown, 1 Unencrypted, 2 Encrypted

Table 80   Required Properties for PackagedComponent

Property	Type	Description/Notes
GroupComponent	REF	PhysicalPackageReference
PartComponent	REF	PhysicalComponentReference

Table 81   Required Server Level Instances of PackagedComponent

Property	Type
GroupComponent	CIM_Chassis
PartComponent	CIM_PhysicalTape

Table 82   Required Properties for StorageMediaLocation

Property	Type	Description/Notes
Tag[key]	String, MaxLen(256)	An arbitrary string that uniquely identifies the Physical Element.
CreationClassName[key]	String	Indicated the name of the class or subclass.
LocationType	Uint16	The type of location.
LocationCoordinates	String	General location information about the physical location of the StorageMediaLocation.
MediaTypesSupported	Uint16[]	Complete list of accepted media types.
MediaCapacity	Uint32	The maximum number of physical media that this StorageMediaLocation can hold.

Table 83 Required Server Level Instances of StorageMediaLocation

Property	Type
Tag[key]	<elementaddress>.<chassis>.<type>
CreationClassName[key]	"CIM_ComputerSystem"
LocationType	Slot(2) or MediaAccessDevice (4) or LimitedAccessPort (6)
LocationCoordinates	"<Location>"
MediaTypesSupported	{ LTO Ultrium(56)}
MediaCapacity	1
Manufacturer *	Branding for slots and IE, vendor for drives
Model *	"LTO" for slots and IE, product ID for drives

 * Properties inherited from the superclass PhysicalPackage.

Table 84 Required Properties for Container

Property	Type	Description/Notes
GroupComponent	REF	PhysicalPackageReference
PartComponent	REF	PhysicalComponentReference

Table 85 Required Server Level Instances of Container(Chassis)

Property	StorageMediaLocation Instances (one for each SML)	PhysicalPackage Instances (one for each PP)
GroupComponent	CIM_Chassis	CIM_Chassis
PartComponent	CIM_StorageMediaLocation	CIM_PhysicalPackage

Table 86 Required Properties for PhysicalMediaInLocation

Property	Type	Description/Notes
Antecedent	REF	StorageMediaLocation Reference
Dependdent	REF	PhysicalMedia Reference

Table 87 Required Server Level Instances of PhysicalMediaInLocation

Property	Instance 1
Antecedent	StorageMediaLocation Reference
Dependent	CIM_PhysicalTape

Table 88   Required Properties of LimitedAccessPort

Property	Type	Description/Notes
SystemCreationClassName[key]	String, MaxLen(256)	The scoping system's CreationClassName.
SystemName[key]	String	The scoping system's name.
CreationClassName[key]	String	Indicates the name of the class or subclass used in the creation of an instance.
DeviceID[key]	String, MaxLen(256)	Opaque.
Locked	Boolean	

Table 89   Required Library Level Instances of LimitedAccessPort

Property	Physical Library Instance 1
SystemCreationClassName[key]	"CIM_ComputerSystem"
SystemName[key]	<IP of the library>
CreationClassName[key]	"<GUID>:<Location>"
DeviceID[key]	True if locked.
Locked	

Associated Indications

Table 90   Creation/Deletion of a ComputerSystem

Query	Comments/Remarks
SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_ComputerSystem SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_ComputerSystem	Event to notify when library is ready to operate when it is initialized (startup) or de-initialized (shutdown). Corresponding events for FC I/O blades will be supported in future.

Table 91   Creation/Deletion of a PhysicalMedia

Query	Comments/Remarks
SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA PhysicalMedia SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_PhysicalMedia SourceInstance ISA CIM_ComputerSystem	Event to notify when a physical tape comes into the library or leaves the library.

Table 92   Creation/Deletion of a TapeDrive

Query	Comments/Remarks
SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_TapeDrive SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_TapeDrive	Event to notify when a tape drive comes into the library or leaves the library.

Table 93   Creation/Deletion of a ChangerDevice

Query	Comments/Remarks
SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_ChangerDevice CIM_SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_ChangerDevice	Event to notify that the robotics is ready for use after library is initialized (startup). Event to notify when the library is getting de-intialized (shutdown) and robotics becomes unavailable. Event to notify when a logical library is created or deleted.

Table 94   Creation/Deletion of an FCPort

Query	Comments/Remarks
SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_FCPort SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_FCPort	Event to notify when an FC port comes into the library or leaves the library. It is the same as when an FC tape drive or FC I/O blade enters or leaves the library.

Table 95   Change in Operational Status of a ComputerSystem

Query	Comments/Remarks
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ComputerSystem AND PreviousInstance.OperationalStatus <> SourceInstance.OperationalStatus	Event to notify change in operational status of overall library status.

Table 96   Change in Operational Status of a PhysicalMedia

Query	Comments/Remarks
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_PhysicalMedia AND PreviousInstance.OperationalStatus <> SourceInstance.OperationalStatus	Event to notify change in operational status of physical media.

Table 97   Change in Operational Status of a TapeDrive

Query	Comments/Remarks
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_TapeDrive AND PreviousInstance.OperationalStatus <> SourceInstance.OperationalStatus	Event to notify change in operational status of a tape drive.

Table 98   Change in Operational Status of a ChangerDevice

Query	Comments/Remarks
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ChangerDevice AND PreviousInstance.OperationalStatus <> SourceInstance.OperationalStatus	Event to notify change in operational status of a changer device.

Table 99   Change in Operational Status of an FCPort

Query	Comments/Remarks
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_FCPort AND PreviousInstance.OperationalStatus <> SourceInstance.OperationalStatus	Event to notify change in operational status of an FCPort.

Fibre Channel Blade Support

This support for Fibre Channel blades is provided through objects belonging to the following SMI-S subprofiles:

- The Multiple Computer System Subprofile
- The Masking Mapping Profile

As of 1.2.0, these subprofiles are not supported by the Storage Media Library profile, and so the Dell PowerVault ML6000 does not show support for them in the Profile Registration Subprofile.

The diagram below shows the library object model with blade support. The objects or instances that were added for blade support are shown in blue, and they are discussed in more detail in the following sections.

Figure 6   Library Object Model with Blade Support
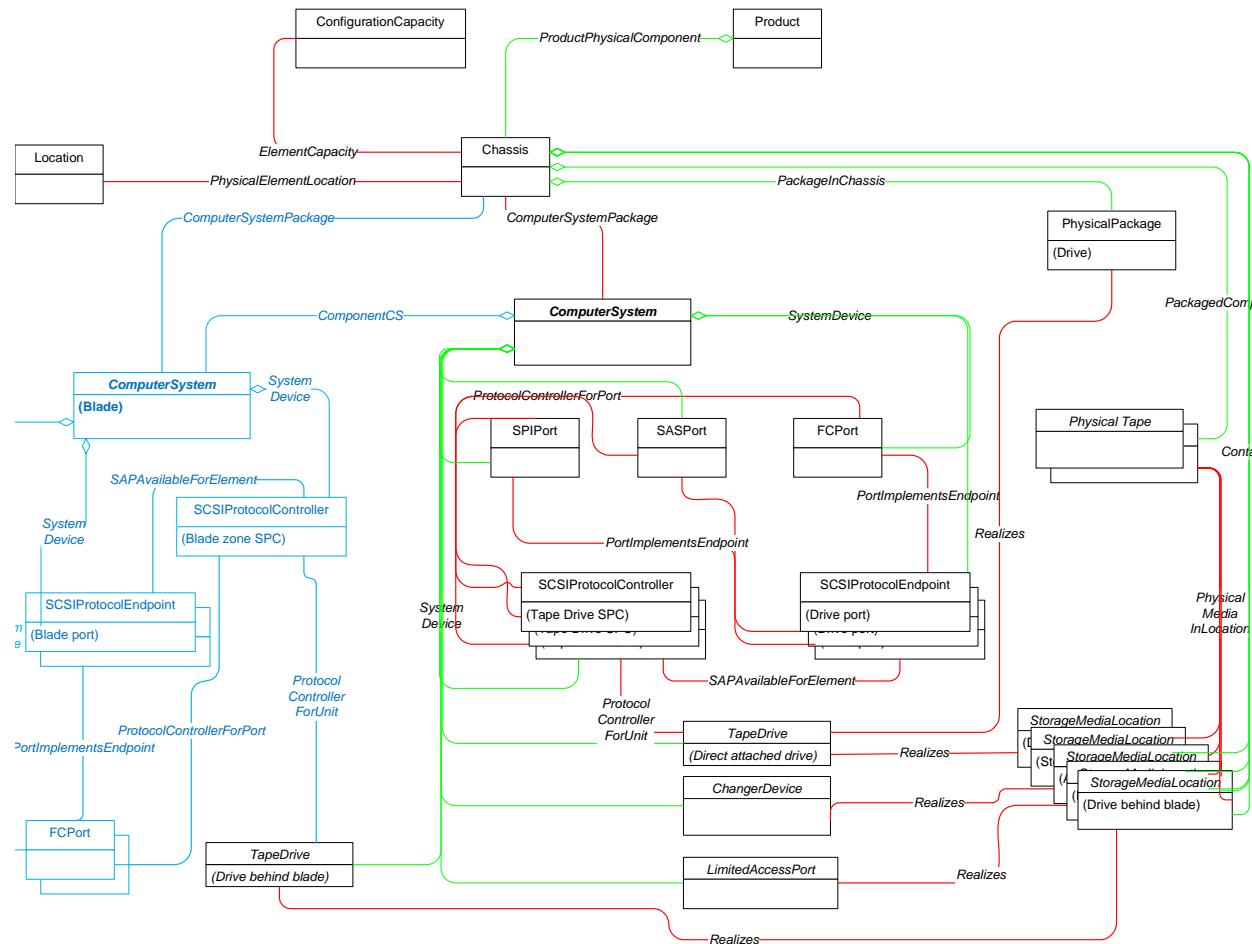
Table 100 Fibre Channel I/O Blade Instances of Computer System

Property	Instances (one per Fibre Channel I/O Blade)
CreationClassName[key]	"CIM_ComputerSystem"
Name[key]	Blade WWNN
Dedicated	Router (4)
OperationalStatus	Blade state, example: ok(2)
OtherIdentifyingInfo	"ComputerSystem"
ElementName	"SMIS_FCBComputerSystem"
NameFormat	"WWN"
PrimaryOwnerName	"DELL"
PrimaryOwnerContact	" "
CaptionBlade	WWN
Description	This is a Fibre Channel I/O blade

Table 101 Required Properties for ComponentCS

Property	Type	Description/Notes
GroupComponent	REF	ComputerSystem Reference
PartComponent	REF	ComputerSystem Reference

Table 102 Required Server Level Instances of ComponentCS

Property	Instances (one for each blade)
GroupComponent	CIM_ComputerSystem (for the library)
PartComponent	CIM_ComputerSystem (for the library)

Table 103 Required Blade Instances of FCPort

Property	Blade Instances (one for each blade target FC port)
SystemCreationClassName[key]	"CIM_ComputerSystem"
SystemName[key]	<HID>
CreationClassName[key]	"SMIS_FCPort"
DeviceID[key]	"B:<bladeID>:<portID>:<location>:WWN"
PortNumber	Example: 1
PermanentAddress	WW Port Name Format is 16 octets 1234567887654321
ElementName	"CIM_FCPort"
OperationalStatus	Port Status. Example: ok(2)
NetworkAddress	0
PortType	Example: N(10)
ActiveFC4Types	SCSI-FCP(8)
OtherNetworkPortType	"" since portType is N(10)
LinkTechnology	FC(4)
OtherLinkTechnology	""
ActiveCOS	3
UsageRestriction	Front-end only(2)

Table 104 Required Blade Instances of SCSIProtocolController (LUN Mapping Disabled)

When LUN Mapping is disabled, the blade has a SCSIProtocolController representing each zone on the blade, and since zoning is tied to target ports, this ties the blade SPC object to a blade target port.

Property	Instances (one for each port on a Fibre Channel I/O blade)
SystemCreationClassName[key]	CIM_ComputerSystem
SystemName[key]	<Vid> <Pid> <Serial>
CreationClassName[key]	CIM_SCSIProtocolController
DeviceID[key]	"B":<bladeIP>:.<port>
OtherIdentifyingInfo	{blade serial}
Caption	"B":<bladeIP>:.<port>
Description	"FC Blade <IP>, port <port>"
ElementName	CIM_SCSIProtocolController
IdentifyingInfoDescriptions	{"Serial Number"}

Table 105 Required Blade Instances of SCSIProtocolEndpoint

Property	Drive Instances (one for each direct-attached drive)
SystemCreationClassName[key]	"CIM_ComputerSystem"
SystemName[key]	HID
CreationClassName[key]	"CIM_SCSIProtocolEndpoint"
DeviceID[key]	"<WWPN>"
ConnectionType	Fibre Channel (2)
Role	Target (3)
OperationalStatus [valuemap]	{Unknown(0)}
ProtocolIFType	Fibre Channel (56)
Description	"SMIS_SCSIProtocolEndpoint"

Table 106 Required Blade Instances of PortImplementsEndpoint

Property	FCPort Instances (one for each blade target port)
Antecedent	CIM_FCPort
Dependent	CIM_SCSIProtocolEndpoint

Table 107 Required Blade Instances of SAPAvailableForElement

Property	Instance (one for each SCSIProtocolController)
AvailableSAP	CIM_SCSIProtocolEndpoint
ManagedElement	CIM_SCSIProtocolController

Table 108 Required Blade Instances of ProtocolControllerForUnit

Property	Drive Instances (one for each drive and each SPC that has access to it through zoning)	Changer Instances (partition) x (blade ports)
Antecedent	CIM_SCSIProtocolController (represents a port that has zoning access to this drive)	CIM_SCSIProtocolController (corresponding to blade port)
Dependent	CIM_TapeDrive	CIM_ChangerDevice
DeviceNumber	Internal LUN of the drive	Internal LUN of changer

Table 109 Required Blade Instances of SystemDevice

Property	FCPort Instances (one for each target port)	SPC Instances (one for each SPC)	SPE Instances (one for each target port)
GroupComponent	CIM_ComputerSystem	CIM_ComputerSystem	CIM_ComputerSystem
PartComponent	CIM_FCPort	CIM_SCSIProtocolController	CIM_SCSIProtocolEndpoint

Table 110 Required Blade Instances of InstalledSoftwareIdentity

Property	Instance 1 (one for each Fibre Channel I/O blade)
System	CIM_ComputerSystem
InstalledSoftware	CIM_SoftwareIdentity

Table 111 Required Blade Instances of InstalledSoftwareIdentity

Property	Blade Instance (1 per Fibre Channel I/O blade)
InstanceID[key]	"FCB:blade id:Version WWN"
VersionString	Blade firmware version
Manufacturer	Same as instance ID

Table 112 Required Blade Instances of ComputerSystemPackage

Property	Instance 1
Antecedent	CIM_PhysicalPackage
Dependent	CIM_ComputerSystem

LUN Mapping (EVPS)

The implementation of LUN Mapping is done for blade systems and is applicable for when the LUN Mapping (EVPS) feature is enabled on the library. The implementation is done using the objects of the SMI-S 1.2.0 Mapping and Masking Subprofile, but the library does not claim support for it because the SML Profile does not support it as of 1.2.0. The diagram below shows the object model with LUN mapping. The classes that are only used for LUN Mapping are shown in orange, while new instances of existing classes are shown in blue.

Figure 7   Object Model with LUN Mapping
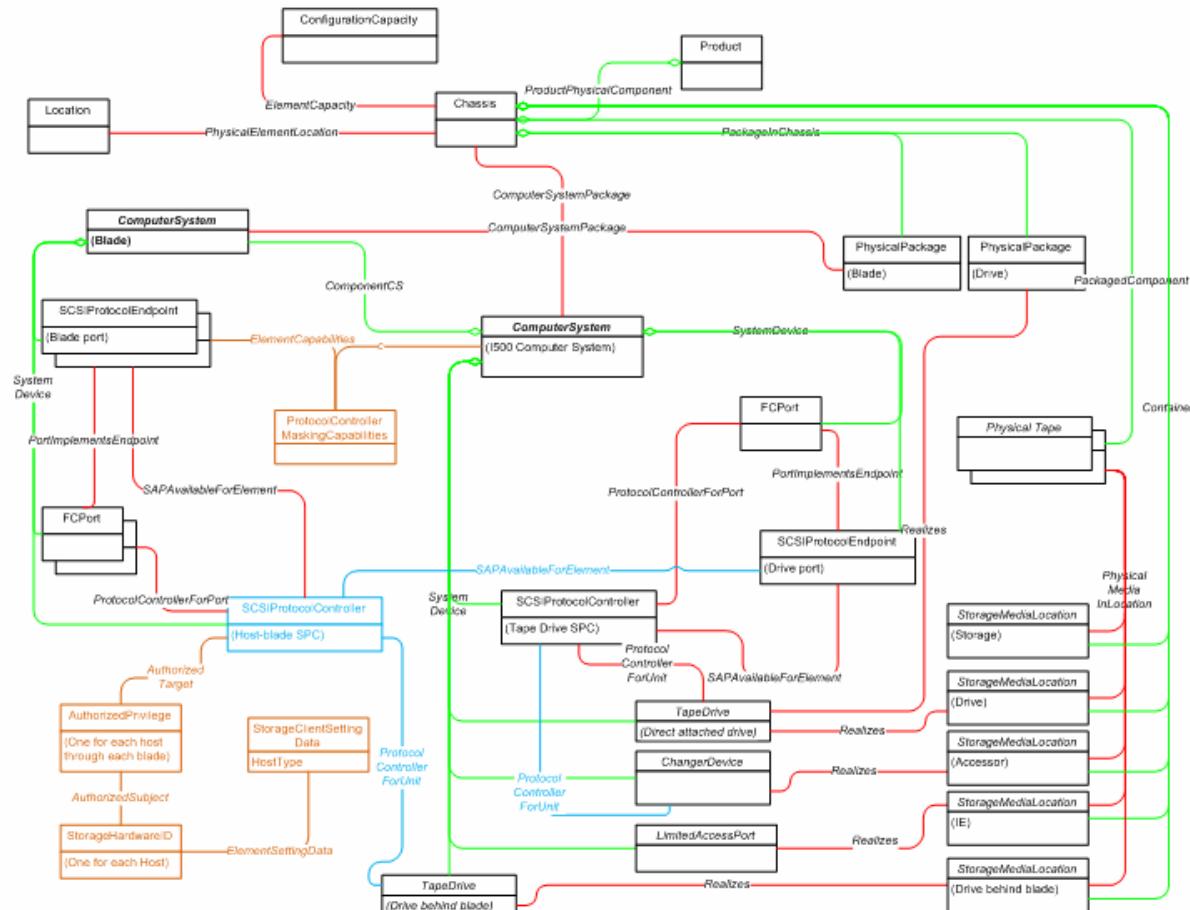
Table 113 Required Properties for StorageHardwareID

Property	Type	Description/Notes
InstanceID[key]	String	This is a unique value for the instance
CurrentlyAuthenticated	Boolean	Whether the identity has been authenticated (optional)
StorageID	String	The hardware ID
IDType	Uint16	Type of ID PortWWN, iSCSI, etc.

Table 114 Required Instances for StorageHardwareID

Property	Instances (one for each host)
InstanceID[key]	<Vendor> <Product> <Serial> <WWN hi: lo>:<blade IP>:<hostIndex>
CurrentlyAuthenticated	"true"
StorageID	WW Port Name Format is 16 octets 12345678:87654321
IDType	2 (WWPN)

Table 115 Required Properties for StorageClientSettingData

Property	Type	Description/Notes
InstanceID[key]	String	This is a unique value for the instance
ClientTypes [valuemap]	Uint16 array	Operating system, version of client (optional)
ElementName	String	The hardware ID
OtherClientTypeDescriptions	String	String for the types that have other client types

Table 116 Required Instances for StorageClientSettingData

Property	Instance (one for each present OS as chosen from host type config)
InstanceID[key]	# (used for internal lookup)
ClientTypes	1 (Other: UNISYS, FCR-2, PV-136T-FC, SWITCH, GATEWAY, GENERIC), 4 (HPUX, HP-V2), 7 (NETWARE), 9(AIX), 14 (LINUX), 15 (Windows), 16 (AS400)
ElementName	String type shown in GUI
OtherClientTypeDescriptions	String type shown in GUI for the ones that have 1 (other) client types

Table 117   Required Properties for AuthorizedPrivilege

Property	Type	Description/Notes
InstanceID[key]	String	This is a unique value for the instance
PrivilegeGranted	Boolean	Whether privilege has been granted or denied (optional)
Activities [valuemap]	Uint16 array	Read/write/create…

Table 118   Required Instances for AuthorizedPrivilege

Property	Instance (one for each host-blade combination)
InstanceID[key]	<Vendor> <Product> <Serial>:<blade IP>:<hostIndex>:<WWN hi: lo>
PrivilegeGranted	True
Activities	5,6 (read,write)

Table 119   Required Properties for ElementSettingData

Property	Type	Description/Notes
SettingData	REF	The SettingData associated with the element.
ManagedElement	REF	The ManagedElement.

Table 120   Required Instances for ElementSettingData

Property	Instance (one for each host)
SettingData	CIM_SettingData
ManagedElement	CIM_StorageHardwareID

Table 121   Required Properties for AuthroizedSubject

Property	Type	Description/Notes
Privilege	REF	Privilege grant or denial.
PrivilegedElement	REF	The subject which has been granted or denied access.

Table 122   Required Instances for AuthorizedSubject

Property	Value (one for each blade-host combination)
Privilege	CIM_AuthorizedPrivilege describing blade-host combination
PrivilegedElement	CIM_StorageHardwareID of the host

Table 123   Required Properties for AuthorizedTarget

Property	Type	Description/Notes
Privilege	REF	Privilege grant or denial
TargetElement	REF	The object to be accessed

Table 124   Required Instances for AuthorizedTarget

Property	Value (one for each blade-host-port combination)
Privilege	CIM_AuthorizedPrivilege of the blade-initiator combination
TargetElement	CIM_SCSIProtocolController for blade-host-port combination

Table 125   Required Properties for ProtocolControllerMaskingCapabilities

Property	Type	Description/Notes
InstanceID[key]	String	This is a unique value for the instance
ElementName	String	
ValidHardwareIDTypes	Uint16 array	Supported ID's for masking
PortsPerView	Uint16	Integer enumeration for one port or multiple ports per view
ClientSelectableDeviceNumbers	Boolean	True if the user selects the device numbers (LUNs)
OneHardwareIDPerView	Boolean	Whether multiple ID's can have the same view
UniqueUnitNumbersPerPort	Boolean	Whether a unit number (LUN) is unique across each port for an SPC
PrivilegeDeniedSupported	Boolean	True if the a view can be created with privilege denied
ProtocolControllerSupportsCollections	Boolean	True if the system has multiple hosts treated together in one view
SPCAllowsNoInitiators	Boolean	Creation related (read-write)
AttachDeviceSupported	Boolean	Creation related (read-write)
ProtocolControllerRequiresAuthorizedIdentity	Boolean	Creation related (read-write)
ExposePathsSupported	Boolean	Creation related (read-write)
CreateProtocolControllerSupported	Boolean	Creation related (read-write)
SPCAllowsNoLUs	Boolean	Creation related (read-write)
SPCAllowsNoTargets	Boolean	Creation related (read-write)
SPCSupportsDefaultViews	Boolean	Creation related (read-write)

Table 126 Required Instances for ProtocolControllerMaskingCapabilities

Property	Top-level ComputerSystem	Blade LUN Mapping On	Blade LUN Mapping Off
InstanceID[key]	<Vid> <Pid> <Serial>:"L"	<Vid> <Pid> <Serial>:"E"	<Vid> <Pid> <Serial>:"B"
ElementName	"CS SPC Masking Capabilities"	"EVPS Enabled SPC Masking Capabilities"	"EVPS Disabled SPC Masking Capabilities"
ValidHardwareIDTypes	{2} (wwpn)	{2} (wwpn)	{2} (wwpn)
PortsPerView	1	1	Multiple
ClientSelectableDeviceNumbers	False	True	False
OneHardwareIDPerView	False	True	False
UniqueUnitNumbersPerPort	False	False	False
PrivilegeDeniedSupported	False	False	False
ProtocolControllerSupportsCollections	False	False	False
SPCAllowsNoInitiators	False	False	False
AttachDeviceSupported	False	False	False
ProtocolControllerRequiresAuthorizedIdentity	False	False	False
ExposePathsSupported	False	False	False
CreateProtocolControllerSupported	False	False	False
SPCAllowsNoLUs	False	False	False
SPCAllowsNoTargets	False	False	False
SPCSupportsDefaultViews	False	False	False

Table 127 Required Properties of ElementCapabilities

Property	Type	Description/Notes
Capabilities	REF	The capabilities supported by the ManagedElement
ManagedElement	REF	The ManagedElement

Table 128 Required Instances for ElementCapabilities

Property	Instance (one for each computer system)
Capabilities	CIM_ProtocolControllerMaskingCapabilities
ManagedElement	CIM_ComputerSystem

Table 129 Lun Mapping Instances of SCSIProtocolController

When LUN mapping is enabled, there is a an instance for each host-blade-port combination. A single host has a single StorageHardwareID, which is then associated through AuthorizedSubject to multiple AuthorizedPrivileges, one for each blade that the host is attached to (or has ever been), and that, in turn, is associated through AuthorizedTarget to two SCSIProtocolControllers (one for each port on the blade).

Property	Instances (one for each host-blade-port combination)
SystemCreationClassName[key]	CIM_ComputerSystem
SystemName	<Vid> <Pid> <Serial>
CreationClassName[key]	CIM_SCSIProtocolController
DeviceID	"B":<bladeIP>:.<port>:<blade host ID>:WWPN_HI:WWPNLO
OtherIdentifyingInfo	{WWPN, hostname, blade serial}
Caption	"B":<bladeIP>:.<port>:<blade host ID>:WWPN_HI:WWPNLO
Description	"FC Blade <IP>, port <port>, host: <hostname> (<WWPN>)"
ElementName	CIM_SCSIProtocolController
IdentifyingInfoDescriptions	{"Host WWPN", "Host name", "Serial Number"}

Table 130 LUN Mapping Instances of ProtocolControllerForUnit

Property	Changer Instances (one for each time that a partition is exposed)	Drive Instances (one for each time that a drive is exposed)
Antecedent	CIM_SCSIProtocolController.	CIM_SCSIProtocolController
Dependent	CIM_ChangerDevice	CIM_TapeDrive
DeviceAccess	2 (Read/Write)	2 (Read/Write)
DeviceNumber	Host LUN	Host LUN
AccessState	2 (online) if host is Online in GUI, 3 (offline) if host is offline in GUI	

SMI-S Common Protocol Interface Specification

Index