

SMI-S Reference Guide SMI-S Reference Guide SMI-S Reference Guide

# Quantum Scalar Intelligent Libraries

# Scalar Series

Scalar Intelligent Libraries SMI-S Reference Guide, 6-01317-06, Ver. A, July 2008, Made in USA.

Quantum Corporation provides this publication “as is” without warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability or fitness for a particular purpose. Quantum Corporation may revise this publication from time to time without notice.

#### **COPYRIGHT STATEMENT**

Copyright 2008 by Quantum Corporation. All rights reserved.

Your right to copy this manual is limited by copyright law. Making copies or adaptations without prior written authorization of Quantum Corporation is prohibited by law and constitutes a punishable violation of the law.

#### **TRADEMARK STATEMENT**

Quantum, the Quantum logo, and Scalar are all registered trademarks of Quantum Corporation.

Other trademarks may be mentioned herein which belong to other companies.

# Contents

---

1 About This Guide and Your Product	1
Explanation of Symbols and Notes	1
Other Documents you Might Need	1
Contact Information	2
Getting More Information or Help	2

---

2 Understanding SMI-S, CIM, and MOF	3
Managing SANs With SMI-S	3
Your Library, SMI-S, and CIM	4
Understanding CIM	4
WBEM Initiative	4
Unified Modeling Language	4
Understanding MOF	5

---

3 CIM Profiles and Subprofiles	7
Server Profile, Version 1.2	7
Storage Library Profile, Version 1.2	7
Storage Library Subprofiles	9
Library Capacity Subprofile	9
Software Subprofile	9
Location Subprofile	9
Access Points Subprofile	9
Limited Access Port Subprofile	9
Physical Package Subprofile	9
Fibre Channel Target Port Subprofile	9
Element Counting Subprofile	10

---

4 Service Discovery and CIM Indications	11
Setting up Authentication and Privacy	11
Enabling the CIM Server	11
CIM Clients	11
Discovering the CIM Server	12
Using CIM Indications	13

---

A MOF Files	17
MOF File for the Scalar i2000 and the Scalar i500 .....	17

---

B References	35
Web Resources .....	35
CIM-XML Tools .....	35

# 1

## 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 Scalar<sup>®</sup> i2000 and Scalar<sup>®</sup> i500 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](http://www.snia.org).

## Explanation of Symbols and Notes

---

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



**WARNING**

**INDICATES A POTENTIALLY HAZARDOUS SITUATION WHICH, IF NOT AVOIDED, COULD RESULT IN DEATH OR BODILY INJURY.**



**CAUTION**

**Indicates a situation that may cause possible damage to equipment, loss of data, or interference with other equipment.**



**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 or at [www.quantum.com/support](http://www.quantum.com/support).

- *Scalar i500 User's Guide* (6-01210-xx)

- *Scalar i2000 User's Guide (6-00421-xx)*
- *ADIC Management Console User's Guide (6-00064-xx)*

## Contact Information

---

### Quantum Corporate Headquarters

Quantum Corporation Headquarters  
1650 Technology Drive, Suite 700  
San Jose, CA 95110-1382

### Technical Publications

Provide documentation feedback at:  
[docs-comments@quantum.com](mailto:docs-comments@quantum.com)

## Getting More Information or Help

---

More information about this product is available on the Service and Support website at [www.quantum.com/support](http://www.quantum.com/support). The Service and Support Website contains a collection of information, including answers to frequently asked questions (FAQs). You can also access software, firmware, and drivers through this site.

For further assistance, or if training is desired, contact Quantum:

Global Call Handling	1-800-284-5101
For additional contact information:	<a href="http://www.quantum.com/support">www.quantum.com/support</a>
To open a Service Request:	<a href="http://www.quantum.com/osr">www.quantum.com/osr</a>
Quantum Corporation	<a href="http://www.quantum.com">www.quantum.com</a>

For the most updated information on Quantum Global Services, please visit [www.quantum.com/support](http://www.quantum.com/support).

# 2

## Understanding SMI-S, CIM, and MOF

---

The Scalar i500 and Scalar i2000 libraries support 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 Scalar i500 and Scalar i2000 libraries. 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 Quantum 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 Quantum Managed Object Format (MOF) file for the Scalar i2000 and the Scalar i500 library, see [MOF Files](#) on page 17.



# 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

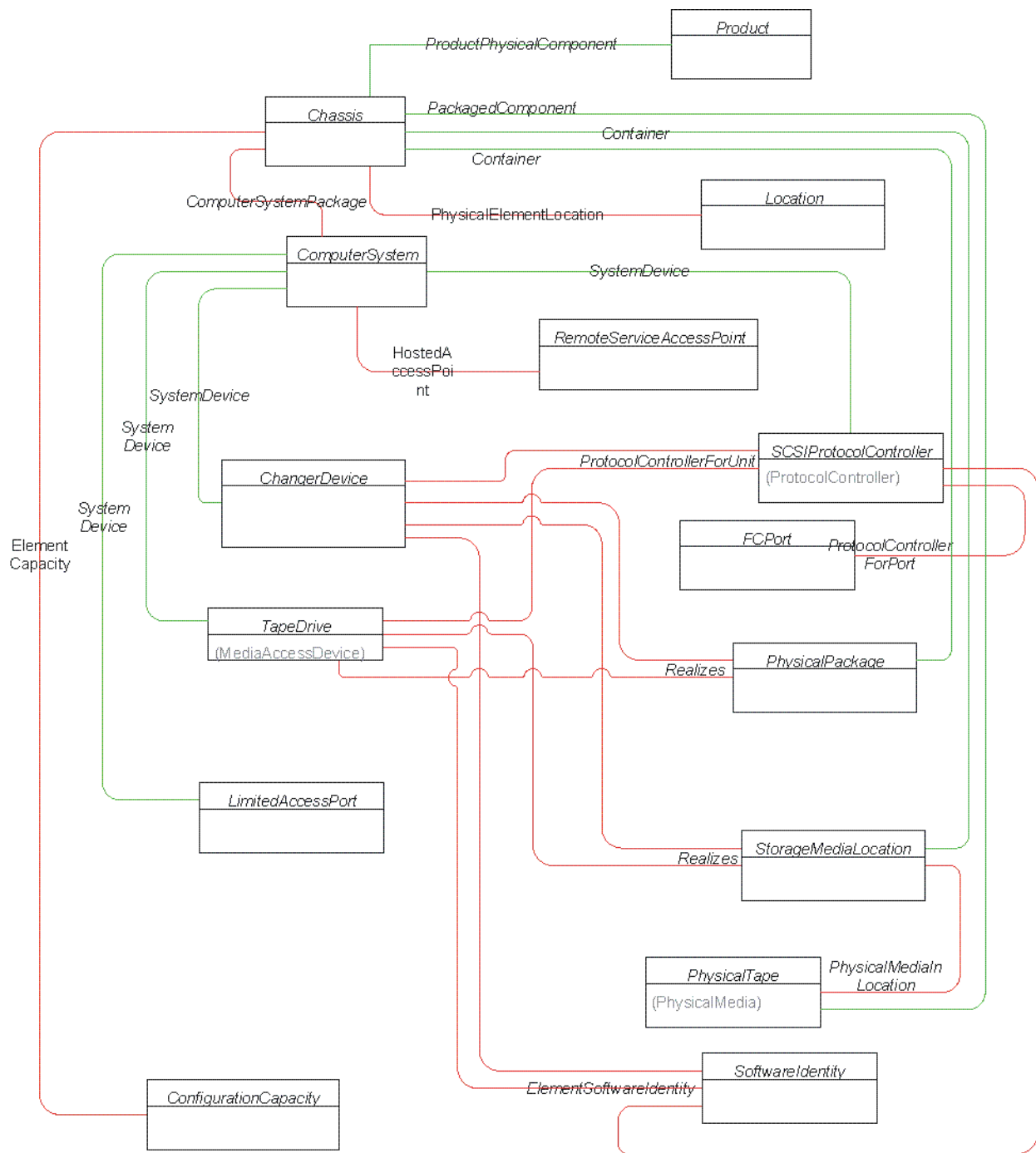
---

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.

# 4

## 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 Scalar i500 and Scalar i2000 library, while the HTTPS port 5989 can be selectively either enabled or disabled on both libraries.

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.



#### Note

You can disable the CIM server by disabling HTTP and/or HTTPS.

On both the Scalar i2000 and the Scalar i500, you can disable both HTTP and HTTPS (which also disables SLP). For more information about performing this task, see library's *User's Guide*.

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

### CIM Clients

CIM clients can connect to the library only by using the admin user and appropriate password on both the Scalar i2000 library and the Scalar i500 library. The interop namespace is `root/cimv2`.

Both the Scalar i500 and Scalar i2000 libraries support Secure Socket Layer (SSL) 3.0 authentication. To use SSL, however, you must have an SSL certificate key. The SSL certificate key for SMI-S clients is provided in the `.pem` file on the library's *Documentation CD*. For more information about the SSL certificate key, refer to the library's *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
<code>enumerateInstances()</code>	Enumerates instances of a CIM class
<code>enumerateInstanceNames()</code>	Enumerates names of instances of a CIM class
<code>getInstance()</code>	Gets a CIM instance
<code>associators()</code>	Enumerates associators of a CIM object
<code>associatorName()</code>	Enumerates names of associators of a CIM object
<code>references()</code>	Enumerates references to a CIM object
<code>referenceName()</code>	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 1.1 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 Scalar i500 and i2000.

**Table 2** Supported Queries for CIM Indications

<b>Area of Library</b>	<b>Query</b>	<b>Purpose</b>
<b>Library</b>	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.
<b>Robotics</b>	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.
<b>I/E Station and Tape Media</b>	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.
<b>Tape Drives/ I/O Blades</b>	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)

<b>Area of Library</b>	<b>Query</b>	<b>Purpose</b>
	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 SNC 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 SNC 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 SNC FC I/O blade.





## MOF Files

The Managed Object Format (MOF) files for the Scalar i2000 and Scalar i500 tape libraries are reproduced below.

The MOF files for the Scalar i2000 and Scalar i500 library are identical.

### MOF File for the Scalar i2000 and the Scalar i500

```
// =====  
// 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::cmplianceelementconformstoprofile")
]
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::cmpliancelementsoftwareidentity")
]
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>
- Standards Based Linux Instrumentation for Manageability (SBLIM): <http://www-124.ibm.com/sblim/clientapps.html>



# Index

<hr/>	
<b>C</b>	
<hr/>	
CIM	
about	4
CIM indications	13
CIM-XML	35
clients	11
Discovering the CIM Server	12
enabling	11
methods	12
profiles and subprofiles	7
contact information	2
<hr/>	
<b>I</b>	
<hr/>	
Indications	
receiving from the library	13
<hr/>	
<b>P</b>	
<hr/>	
Profile	
Server profile	7
Storage Library profile	7
<hr/>	
<b>S</b>	
<hr/>	
safety	
intended use	1
statements	1
symbols and notes	1
SAN management	3
Subprofile	
Fibre Channel Target Port	9
Library Capacity	9
Limited Access Port	9
Location	9
Physical Package	9
Software	9
Subprofiles	9
Symbols and Notes, explained	1
<hr/>	
<b>T</b>	
<hr/>	
technical assistance	2
<hr/>	
Traps	4
<hr/>	
<b>U</b>	
<hr/>	
UML	
about	4
using	
CIM indications	13
<hr/>	
<b>W</b>	
<hr/>	
WBEM	
about	4
queries	13

