



Storage Management Technical Specification, Part 8 Media Libraries

Version 1.2.0, Revision 6

"This document has been released and approved by the SNIA. The SNIA believes that the ideas, methodologies and technologies described in this document accurately represent the SNIA goals and are appropriate for widespread distribution. Suggestion for revision should be directed to the Technical Council Managing Director at tcmd@snia.org."

SNIA Technical Position

22 October, 2007

Errata/Change Log

20071022

No errata have been identified for 1.2.0.

The SNIA hereby grants permission for individuals to use this document for personal use only, and for corporations and other business entities to use this document for internal use only (including internal copying, distribution, and display) provided that:

- 1) Any text, diagram, chart, table or definition reproduced must be reproduced in its entirety with no alteration, and,
- 2) Any document, printed or electronic, in which material from this document (or any portion hereof) is reproduced must acknowledge the SNIA copyright on that material, and must credit the SNIA for granting permission for its reuse.

Other than as explicitly provided above, you may not make any commercial use of this document, sell any or this entire document, or distribute this document to third parties. All rights not explicitly granted are expressly reserved to SNIA.

Permission to use this document for purposes other than those enumerated above may be requested by e-mailing tcmd@snia.org please include the identity of the requesting individual and/or company and a brief description of the purpose, nature, and scope of the requested use.

Copyright © 2003-2007 Storage Networking Industry Association.

INTENDED AUDIENCE

This document is intended for use by individuals and companies engaged in developing, deploying, and promoting interoperable multi-vendor SANs through the SNIA organization.

DISCLAIMER

The information contained in this publication is subject to change without notice. The SNIA makes no warranty of any kind with regard to this specification, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The SNIA shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this specification.

Suggestions for revisions should be directed to <http://www.snia.org/feedback/>.

Copyright © 2003-2007 SNIA. All rights reserved. All other trademarks or registered trademarks are the property of their respective owners.

Portions of the CIM Schema are used in this document with the permission of the Distributed Management Task Force (DMTF). The CIM classes that are documented have been developed and reviewed by both the Storage Networking Industry Association (SNIA) and DMTF Technical Working Groups. However, the schema is still in development and review in the DMTF Working Groups and Technical Committee, and subject to change.

CHANGES TO THE SPECIFICATION

Each publication of this specification is uniquely identified by a three-level identifier, comprised of a version number, a release number and an update number. The current identifier for this specification is version 1.2.0. Future publications of this specification are subject to specific constraints on the scope of change that is permissible from one publication to the next and the degree of interoperability and backward compatibility that should be assumed between products designed to different publications of this standard. The SNIA has defined three levels of change to a specification:

- **Major Revision:** A major revision of the specification represents a substantial change to the underlying scope or architecture of the SMI-S API. A major revision results in an increase in the version number of the version identifier (e.g., from version 1.x.x to version 2.x x). There is no assurance of interoperability or backward compatibility between releases with different version numbers.
- **Minor Revision:** A minor revision of the specification represents a technical change to existing content or an adjustment to the scope of the SMI-S API. A minor revision results in an increase in the release number of the specification's identifier (e.g., from x.1.x to x.2.x). Minor revisions with the same version number preserve interoperability and backward compatibility.
- **Update:** An update to the specification is limited to minor corrections or clarifications of existing specification content. An update will result in an increase in the third component of the release identifier (e.g., from x.x.1 to x.x.2). Updates with the same version and minor release levels preserve interoperability and backward compatibility.

TYPOGRAPHICAL CONVENTIONS

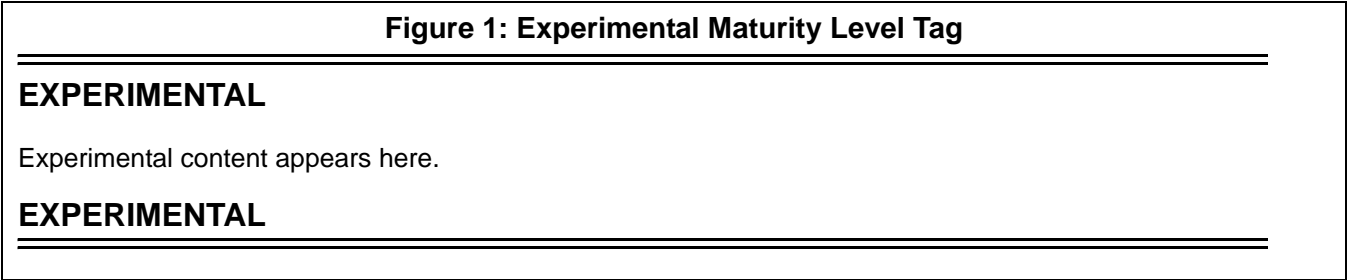
This specification has been structured to convey both the formal requirements and assumptions of the SMI-S API and its emerging implementation and deployment lifecycle. Over time, the intent is that all content in the specification will represent a mature and stable design, be verified by extensive implementation experience, assure consistent support for backward compatibility, and rely solely on content material that has reached a similar level of maturity. Unless explicitly labeled with one of the subordinate maturity levels defined for this specification, content is assumed to satisfy these requirements and is referred to as "Finalized". Since much of the evolving specification

content in any given release will not have matured to that level, this specification defines three subordinate levels of implementation maturity that identify important aspects of the content's increasing maturity and stability. Each subordinate maturity level is defined by its level of implementation experience, its stability and its reliance on other

emerging standards. Each subordinate maturity level is identified by a unique typographical tagging convention that clearly distinguishes content at one maturity model from content at another level.

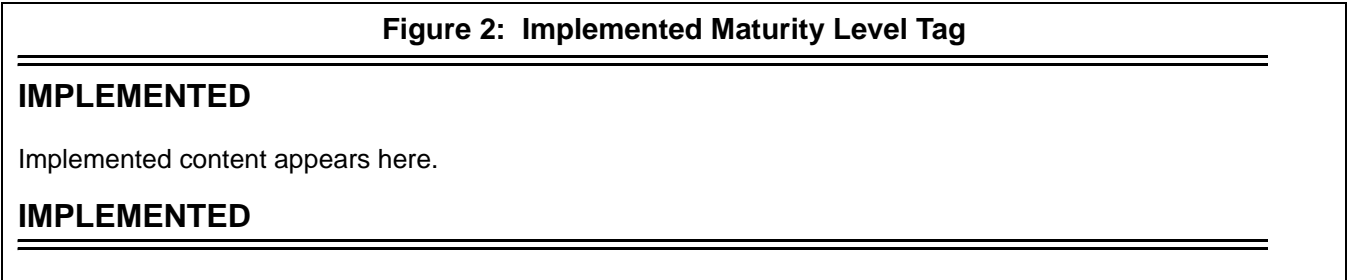
Experimental Maturity Level

No material is included in this specification unless its initial architecture has been completed and reviewed. This material is referred to as “Experimental”. It is presented here as an aid to implementers who are interested in likely future developments within the SMI specification. Some content included in this specification has complete and reviewed design, but lacks implementation experience and the maturity gained through implementation experience. This content is included in order to gain wider review and to gain implementation experience. The contents of an Experimental profile may change as implementation experience is gained. There is a high likelihood that the changed content will be included in an upcoming revision of the specification. Experimental material can advance to a higher maturity level as soon as implementations are available. Figure 1 is a sample of the typographical convention for Experimental content.



Implemented Maturity Level

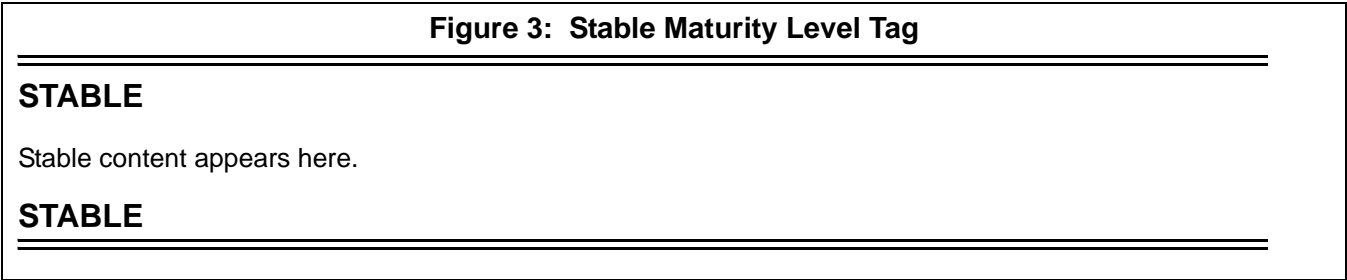
Profiles for which initial implementations have been completed are classified as “Implemented”. This indicates that at least two different vendors have implemented the profile, including at least one provider implementation. At this maturity level, the underlying architecture and modeling are stable, and changes in future revisions will be limited to the correction of deficiencies identified through additional implementation experience. Should the material become obsolete in the future, it must be deprecated in a minor revision of the specification prior to its removal from subsequent releases. Figure 2 is a sample of the typographical convention for Implemented content.



Stable Maturity Level

Once content at the Implemented maturity level has garnered additional implementation experience, it can be tagged at the Stable maturity level. Material at this maturity level has been implemented by three different vendors, including both a provider and a client. Should material that has reached this maturity level become obsolete, it may only be deprecated as part of a minor revision to the specification. Material at this maturity level that has been deprecated may only be removed from the specification as part of a major revision. A profile that has reached this maturity level is guaranteed to preserve backward compatibility from one minor specification revision to the next.

As a result, Profiles at or above the Stable maturity level shall not rely on any content that is Experimental. Figure 3 is a sample of the typographical convention for Implemented content.



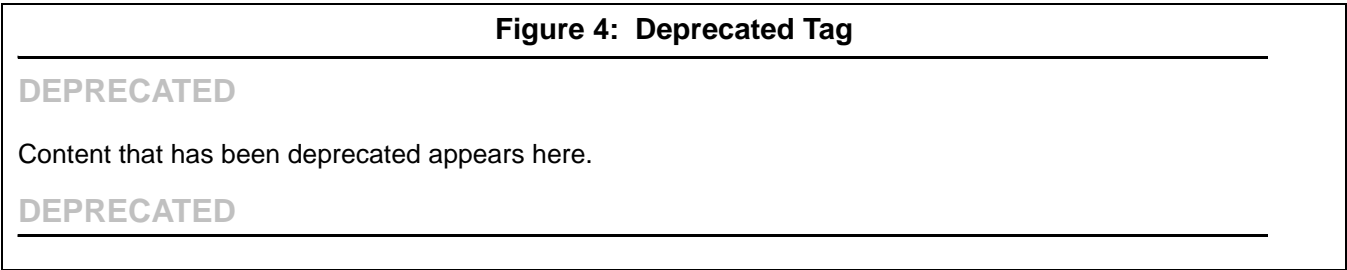
Finalized Maturity Level

Content that has reached the highest maturity level is referred to as “Finalized.” In addition to satisfying the requirements for the Stable maturity level, content at the Finalized maturity level must solely depend upon or refine material that has also reached the Finalized level. If specification content depends upon material that is not under the control of the SNIA, and therefore not subject to its maturity level definitions, then the external content is evaluated by the SNIA to assure that it has achieved a comparable level of completion, stability, and implementation experience. Should material that has reached this maturity level become obsolete, it may only be deprecated as part of a major revision to the specification. A profile that has reached this maturity level is guaranteed to preserve backward compatibility from one minor specification revision to the next. Over time, it is hoped that all specification content will attain this maturity level. Accordingly, there is no special typographical convention, as there is with the other, subordinate maturity levels. Unless content in the specification is marked with one of the typographical conventions defined for the subordinate maturity levels, it should be assumed to have reached the Finalized maturity level.

Deprecated Material

Non-Experimental material can be deprecated in a subsequent revision of the specification. Sections identified as “Deprecated” contain material that is obsolete and not recommended for use in new development efforts. Existing and new implementations may still use this material, but shall move to the newer approach as soon as possible. The maturity level of the material being deprecated determines how long it will continue to appear in the specification. Implemented content shall be retained at least until the next revision of the specialization, while Stable and Finalized material shall be retained until the next major revision of the specification. Providers shall implement the deprecated elements as long as it appears in the specification in order to achieve backward compatibility. Clients may rely on deprecated elements, but are encouraged to use non-deprecated alternatives when possible.

Deprecated sections are documented with a reference to the last published version to include the deprecated section as normative material and to the section in the current specification with the replacement. Figure 4 contains a sample of the typographical convention for deprecated content.



USAGE

The SNIA hereby grants permission for individuals to use this document for personal use only, and for corporations and other business entities to use this document for internal use only (including internal copying, distribution, and display) provided that:

- 1) Any text, diagram, chart, table or definition reproduced shall be reproduced in its entirety with no alteration.
- 2) Any document, printed or electronic, in which material from this document (or any portion hereof) is reproduced shall acknowledge the SNIA copyright on that material, and shall credit the SNIA for granting permission for its reuse.

Other than as explicitly provided above, you may not make any commercial use of this document, sell any or this entire document, or distribute this document to third parties. All rights not explicitly granted are expressly reserved to SNIA.

Permission to use this document for purposes other than those enumerated above may be requested by e-mailing tcmd@snia.org please include the identity of the requesting individual and/or company and a brief description of the purpose, nature, and scope of the requested use.

Contents

Errata/Change Log	iii
List of Tables	xi
List of Figures	xiii
Foreword.....	xv
1. Scope	1
2. Normative References	3
2.1 General.....	3
2.2 Approved references	3
2.3 References under development	3
2.4 Other references.....	3
3. Terms and definitions	5
3.1 General.....	5
3.2 Definitions.....	5
4. Storage Library Profile	7
4.1 Description.....	7
4.2 Health and Fault Management Considerations	11
4.3 Cascading Considerations.....	11
4.4 Supported Subprofiles and Packages	11
4.5 Methods of this Profile	11
4.6 Client Considerations and Recipes	12
4.7 Registered Name and Version	13
4.8 CIM Elements	14
5. Element Counting Subprofile	27
5.1 Description.....	27
5.2 Health and Fault Management Considerations	27
5.3 Cascading Considerations.....	27
5.4 Supported Subprofiles and Packages	27
5.5 Methods of the Profile.....	27
5.6 Client Considerations and Recipes	29
5.7 Registered Name and Version	30
5.8 CIM Elements	31
6. InterLibraryPort Connection Subprofile	33
6.1 Description.....	33
6.2 Health and Fault Management Considerations	33
6.3 Cascading Considerations.....	33
6.4 Supported Subprofiles and Packages	34
6.5 Methods of the Profile.....	34
6.6 Client Considerations and Recipes	34
6.7 Registered Name and Version	34
6.8 CIM Elements	34
7. Library Capacity Subprofile	37
7.1 Description.....	37
7.2 Health and Fault Management Considerations	37
7.3 Cascading Considerations.....	37
7.4 Supported Subprofiles and Packages	37
7.5 Client Considerations and Recipes	37
7.6 Registered Name and Version	38
7.7 CIM Elements	38
8. LibraryAlert Events/Indications for Library Devices	41
8.1 Description.....	41
8.2 Health and Fault Management Considerations	41
8.3 Cascading Considerations.....	41

8.4	Supported Subprofiles and Packages	41
8.5	Methods of the Profile.....	41
8.6	Client Considerations and Recipes	41
8.7	Registered Name and Version	57
8.8	CIM Elements	57
9.	Limited Access Port Elements Subprofile	59
9.1	Description.....	59
9.2	Health and Fault Management Considerations	60
9.3	Cascading Considerations.....	60
9.4	Supported Subprofiles and Packages	60
9.5	Methods of the Profile.....	60
9.6	Registered Name and Version	60
9.7	CIM Elements	61
10.	Media Movement Subprofile	65
10.1	Description.....	65
10.2	Health and Fault Management Considerations	66
10.3	Cascading Considerations.....	67
10.4	Supported Subprofiles and Packages	67
10.5	Methods of the Profile.....	67
10.6	Client Considerations and Recipes	68
10.7	Registered Name and Version	68
10.8	CIM Elements	68

List of Tables

Table 1.	Supported Profiles for Storage Library	11
Table 2.	CIM Elements for Storage Library	14
Table 3.	SMI Referenced Properties/Methods for CIM_ComputerSystem	16
Table 4.	SMI Referenced Properties/Methods for CIM_ComputerSystem	16
Table 5.	SMI Referenced Properties/Methods for CIM_ChangerDevice	17
Table 6.	SMI Referenced Properties/Methods for CIM_MediaAccessDevice	18
Table 7.	SMI Referenced Properties/Methods for CIM_PackagedComponent	19
Table 8.	SMI Referenced Properties/Methods for CIM_PhysicalMedia	19
Table 9.	SMI Referenced Properties/Methods for CIM_PhysicalMediaInLocation	20
Table 10.	SMI Referenced Properties/Methods for CIM_StorageMediaLocation	20
Table 11.	SMI Referenced Properties/Methods for CIM_Realizes	21
Table 12.	SMI Referenced Properties/Methods for CIM_Chassis	21
Table 13.	SMI Referenced Properties/Methods for CIM_ElementSoftwareIdentity	22
Table 14.	SMI Referenced Properties/Methods for CIM_ComputerSystemPackage	22
Table 15.	SMI Referenced Properties/Methods for CIM_SystemDevice	22
Table 16.	SMI Referenced Properties/Methods for CIM_SCSIProtocolController	23
Table 17.	SMI Referenced Properties/Methods for CIM_ProtocolControllerForUnit	23
Table 18.	SMI Referenced Properties/Methods for CIM_SoftwareIdentity	24
Table 19.	SMI Referenced Properties/Methods for CIM_ElementSoftwareIdentity	24
Table 20.	SMI Referenced Properties/Methods for CIM_StorageLibraryCapabilities	25
Table 21.	SMI Referenced Properties/Methods for CIM_ElementCapabilities	25
Table 22.	CIM Elements for Storage Library Element Counting	31
Table 23.	SMI Referenced Properties/Methods for CIM_ConfigurationReportingService	31
Table 24.	SMI Referenced Properties/Methods for CIM_HostedService	32
Table 25.	CIM Elements for Storage Library InterLibraryPort Connection	34
Table 26.	SMI Referenced Properties/Methods for CIM_InterLibraryPort	35
Table 27.	SMI Referenced Properties/Methods for CIM_LibraryExchange	35
Table 28.	CIM Elements for Storage Library Capacity	38
Table 29.	SMI Referenced Properties/Methods for CIM_ConfigurationCapacity	38
Table 30.	SMI Referenced Properties/Methods for CIM_ElementCapacity	39
Table 31.	LibraryAlert Property Settings	41
Table 32.	Vendor Specific Properties of LibraryAlert	42
Table 33.	Variable Alert Properties for LibraryAlert	42
Table 34.	SCSI TapeAlert-based Properties	42
Table 35.	LibraryAlert AlertIndication Properties	43
Table 36.	CIM Elements for SML_Events	57
Table 37.	SMI Referenced Properties/Methods for CIM_AlertIndication	57
Table 38.	CIM Elements for Storage Library Limited Access Port Elements	61
Table 39.	SMI Referenced Properties/Methods for CIM_LimitedAccessPort	62
Table 40.	SMI Referenced Properties/Methods for CIM_Magazine	62
Table 41.	SMI Referenced Properties/Methods for CIM_Realizes	63
Table 42.	SMI Referenced Properties/Methods for CIM_SystemDevice	63
Table 43.	SMI Referenced Properties/Methods for CIM_Container	64
Table 44.	Media Movement Standard Messages	66
Table 45.	CIM Elements for Storage Library Media Movement	68
Table 46.	SMI Referenced Properties/Methods for CIM_HostedService	69
Table 47.	SMI Referenced Properties/Methods for SNIA_MediaMovementService	69

List of Figures

Figure 1.	Experimental Maturity Level Tag	vi
Figure 2.	Implemented Maturity Level Tag.....	vi
Figure 3.	Stable Maturity Level Tag	vii
Figure 4.	Deprecated Tag	vii
Figure 5.	Storage Library-centric Instance Diagram	8
Figure 6.	MediaAccessDevice-centric Instance Diagram.....	9
Figure 7.	ChangerDevice-centric Instance Diagram	9
Figure 8.	Physical View Instance Diagram.....	10
Figure 9.	StorageMediaLocation Instance Diagram.....	10
Figure 10.	Instance Diagram.....	27
Figure 11.	InterLibraryPort Connection Instance Diagram.....	33
Figure 12.	Library Capacity Instance Diagram.....	37
Figure 13.	Tape Libraries with Magazines in LimitedAccessPorts.....	59
Figure 14.	Tape Libraries with no Magazines in LimitedAccessPorts.....	60
Figure 15.	Storage Library Centric View	65
Figure 16.	Media-centrc View	66

Foreword

Storage Library Profile and related subprofiles defined in this book provide a standard CIM interface to monitor and control various aspects of removable media libraries including tape libraries. Once a library supports this specification, any SMI-S client based on this standard can discover a tape library, determine its capacity, perform inventory, monitor status, move tapes and perform other configuration and control operations. This specification also standardizes library specific life-cycle and alert indications that are delivered to a client asynchronously, once a client subscribes to these indications.

This book covers Part 8 (Media Libraries) of the SMI-S standard listed below. While Part 1 describes SMI-S concepts and terms, some of the profiles and subprofiles referenced in Storage Library profile are specified in Part 2 Common Profiles book.

Parts of this Standard

This standard is subdivided in the following parts:

- *Storage Management Technical Specification, Part 1 Common Architecture*
- *Storage Management Technical Specification, Part 2 Common Profiles*
- *Storage Management Technical Specification, Part 3 Block Devices*
- *Storage Management Technical Specification, Part 4 File Systems*
- *Storage Management Technical Specification, Part 5 Fabric*
- *Storage Management Technical Specification, Part 6 Host Elements*
- *Storage Management Technical Specification, Part 7 Information Lifecycle Management*
- *Storage Management Technical Specification, Part 8 Media Libraries*

Acknowledgements

The SNIA SMI Technical Steering Group, which developed and reviewed this standard, would like to recognize the significant contributions made by the following members:

<i>Organization Represented</i>	<i>Name of Representative</i>
Brocade Communications Systems	John Crandall
EMC Corporation	Kamesh Aiyer
	Edgar St. Pierre
Hewlett-Packard	Steve Peters
Hitachi Data Systems	Steve Quinn
IBM	Duane Baldwin
	Jack Gelb
	Mike Walker
iStor Networks, Inc.	Scott Baker
Network Appliance	Alan Yoder
Sun Microsystems	Mark Carlson
Symantec	Steve Hand
	Paul von Behren

SNIA Web Site

Current SNIA practice is to make updates and other information available through their web site at <http://www.snia.org>

SNIA Address

Requests for interpretation, suggestions for improvement and addenda, or defect reports are welcome. They should be sent via the SNIA Feedback Portal at <http://www.snia.org/feedback/> or by mail to the Storage Networking Industry Association, 500 Sansome Street, Suite #504, San Francisco, CA 94111, U.S.A.

Clause 1: Scope

This version of the Storage Library profile specification models various details of the following objects of the media library for monitoring.

- Library
- Drives
- Changer Devices
- Slots
- IO Slots
- SCSI Interfaces and SCSI and FC Target Ports
- Physical Tapes
- Physical Package
- Magazines

In general, a CIM client can monitor the health and status of the above objects as well as get alert, status change and lifecycle CIM indications. In addition, a client can control the movement of media in a library using this specification.

The future versions of this specification shall address partitioned tape libraries and virtual tape libraries. Note that the experimental subprofile modelling partitioned tape libraries and virtual tape libraries in the previous version of this specification has been withdrawn and hence is now omitted from this specification.

Clause 2: Normative References

2.1 General

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

2.2 Approved references

ISO/IEC 14776-452, SCSI Primary Commands - 2 (SPC-2) [ANSI INCITS.351-2001]

ISO/IEC 24775 Storage Management

2.3 References under development

Storage Management Technical Specification, Part 1 Common Architecture

Storage Management Technical Specification, Part 2 Common Profiles

Storage Management Technical Specification, Part 3 Block Devices

ISO/IEC 14776-452, SCSI Primary Commands - 3 (SPC-3) [ANSI INCITS.351-2005]

2.4 Other references

DMTF DSP0214:2004 CIM Operations over HTTP

Normative References

Clause 3: Terms and definitions

3.1 General

For the purposes of this document, the terms and definitions given in *Storage Management Technical Specification, Part 1 Common Architecture* and the following apply.

3.2 Definitions

3.2.1 Changer Device

The robotic arm and control logic within a storage media library that moves media from one location to another.

3.2.2 Media Access Device

A device that performs read and write operations on media. In tape libraries, it is the tape drive.

3.2.3 Storage Media Location

Various locations within a media library where the physical media can be placed. These include the changer devices, the media access devices, physical slots or magazines, and I/O slots.

3.2.4 Storage Media Library

A library in which a large number of removable media can be stored and retrieved. A library also contains a limited number of media access devices for reading and writing to the media. A changer device within the library moves the media between a stored location and drive or between two locations. The drives, changers and the library are controlled by a host typically via the SCSI and/or FC ports, but other types of ports are possible. A storage media library typically is a tape library.

3.2.5 Limited Access Port

An operator-accessible window of a storage media library through which physical media is fed into the library or physical media can be retrieved out of a library. A Limited access port is also known as an I/O Port, Import Export Port, Mailslot, etc.

3.2.6 Library Capacity

The capacity of a storage media library is measured in terms of the number of physical media it can hold.

3.2.7 Magazine

A magazine is a container that holds multiple physical media. Some storage media libraries have magazines that fit into the physical slot instead of single media.

STABLE**Clause 4: Storage Library Profile****4.1 Description**

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 2.12.1 model and defines the subset of classes that supply the necessary information 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 implementers and client application developers. Detailed descriptions of classes are from the CIM 2.12.1 schema.

The relevant objects for a storage library should be instantiated in the name space of the provider (or agent) for a storage library resource. Whenever an instance of a class for a resource may exist in multiple name spaces a durable name is defined to aid clients in correlating the objects across name spaces. For storage libraries, durable names are defined for the following resources:

- ChangerDevice
- ComputerSystem
- MediaAccessDevice

The durable names are defined in a following subsection of this profile. All other objects do not require durable names and have instances within a single name space.

4.1.1 Instance Diagrams

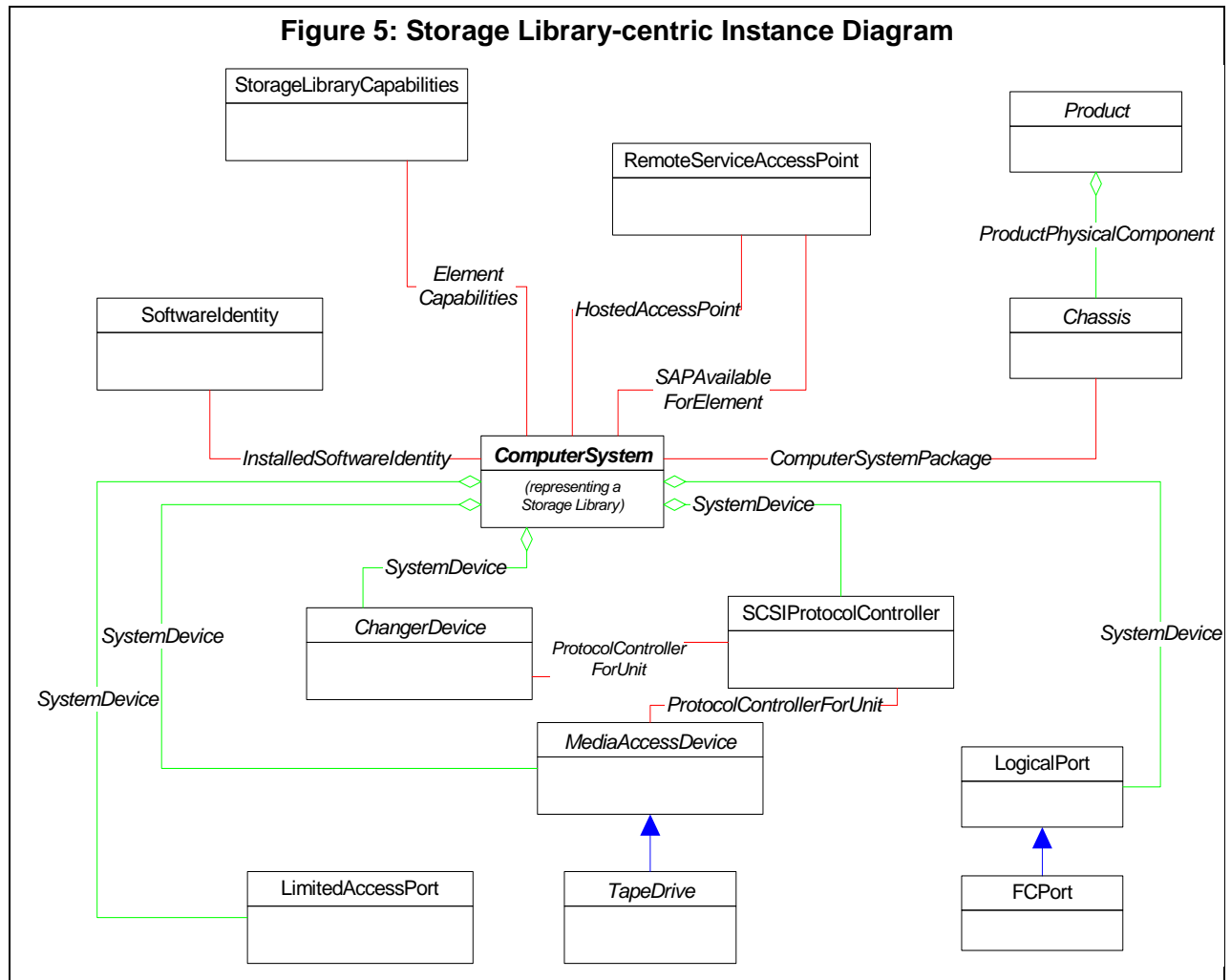
The following instance diagrams represent five related views of the storage library profile:

- a) System Level
- b) MediaAccessDevice and its physical and logical relationships
- c) ChangerDevice and its connections to SoftwareIdentity, ProtocolController, and StorageMediaLocation
- d) StorageMediaLocation and its relationship to PhysicalMedia and other physical classes
- e) StorageMediaLocation and its required Realizes relationships.

4.1.2 System Level View

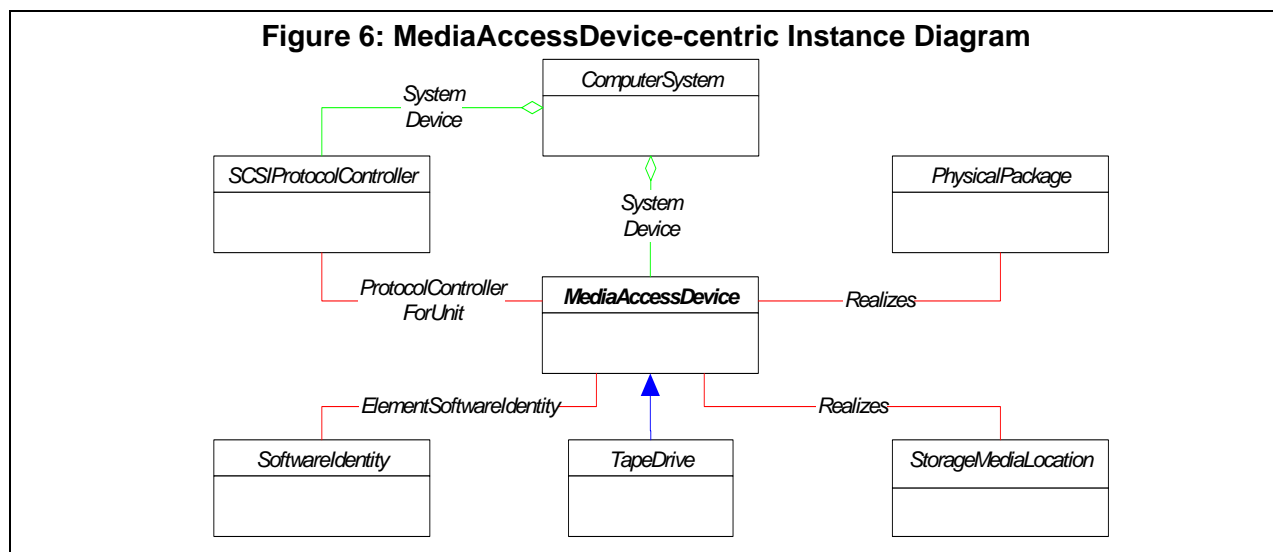
Figure 5 shows the required components for a ComputerSystem. Note that LogicalDevice subclasses shall be associated with ComputerSystem via SystemDevice.

Note: Classes using a red outline and associations using a dotted outline represent optional components that have been included in the diagram as an aid to understanding.



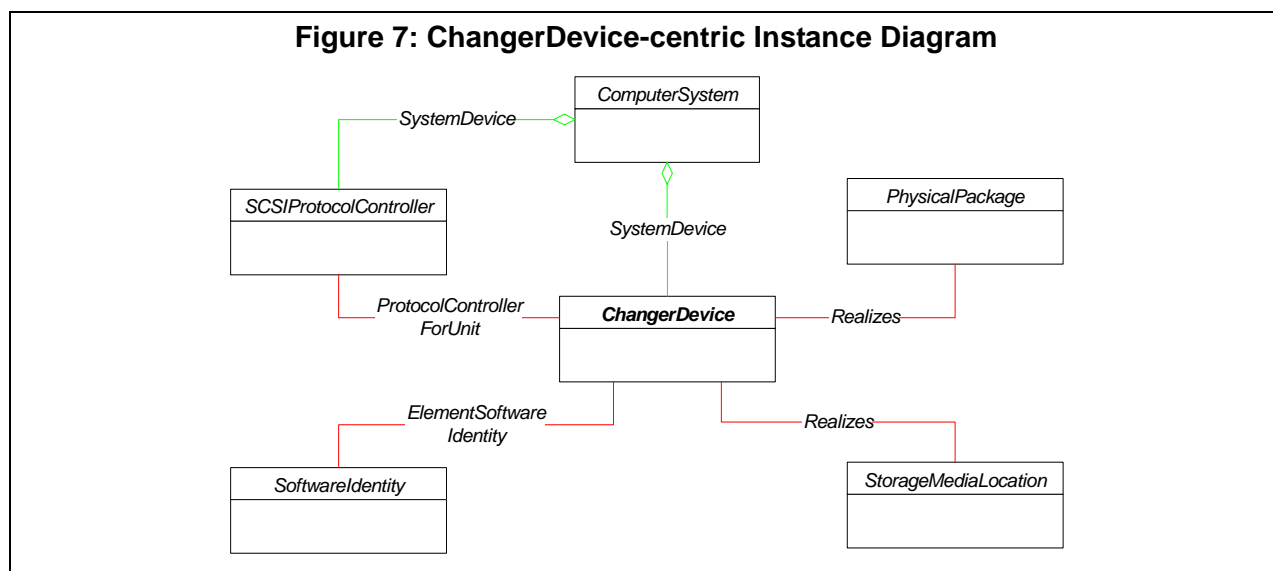
4.1.3 MediaAccessDevice-centric View

Figure 6 shows the required classes related to MediaAccessDevice. Though not shown in this figure, both MediaAccessDevice and ProtocolController are connected to a ComputerSystem instance through the SystemDevice association. In some libraries, notably small autoloaders, external hosts access a library's ChangerDevice through the ProtocolController of a MediaAccessDevice. For such libraries, an additional ProtocolControllerForUnit association should be instantiated between the MediaAccessDevice's ProtocolController and the affected ChangerDevice. ProtocolControllerForUnit is a many-to-many association, so a single ProtocolController can be connected to multiple LogicalDevices if this accurately represents a library's configuration.

Figure 6: MediaAccessDevice-centric Instance Diagram

4.1.4 ChangerDevice-centric View

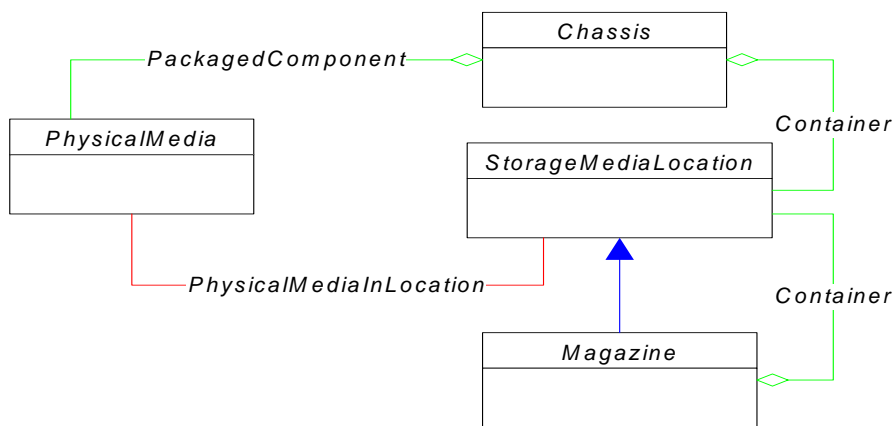
Figure 7 shows the required classes related to ChangerDevice.

Figure 7: ChangerDevice-centric Instance Diagram

4.1.5 Physical View

Figure 8 shows important physical components of a storage library and how they relate. With regard to StorageMediaLocation and Magazine, one of two implementation alternatives shall be selected:

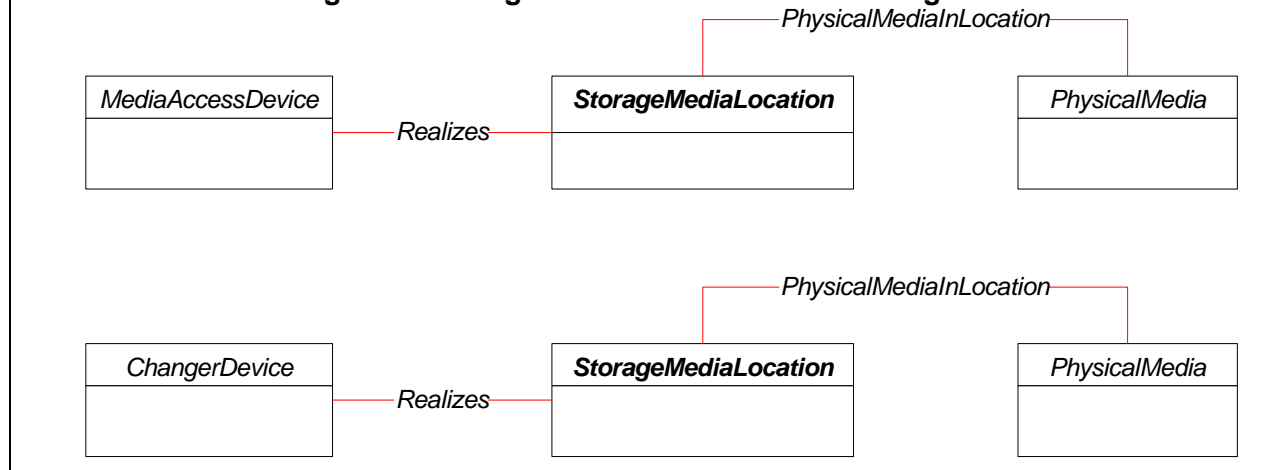
- Instantiate multiple Magazines associated to Chassis via Container, then instantiate StorageMediaLocations that are contained (again via Container) within each Magazine;
- Instantiate multiple StorageMediaLocations directly associated to Chassis via Container, without the use of Magazines. Other optional classes, such as Panel, can also be used to group StorageMediaLocations, but this is not mandatory.

Figure 8: Physical View Instance Diagram

4.1.6 StorageMediaLocation Instance Diagram

Figure 9 shows relationships between various LogicalDevices (i.e., MediaAccessDevices, LimitedAccessPort, and ChangerDevice) and StorageMediaLocation. For each LogicalDevice that can hold media, at least one StorageMediaLocation shall be associated via Realizes.

The figure also shows how PhysicalMedia is conceptually placed “inside” a LogicalDevice by associating PhysicalMedia with a StorageMediaLocation that Realizes a LogicalDevice (see Figure 9). All tapes, irrespective of the location, are associated with the chassis using PackedComponent.

Figure 9: StorageMediaLocation Instance Diagram

4.1.7 Durable Names and Correlatable IDs of the Profile

Different implementations use different approaches to uniquely identify the SCSI units pertinent to Storage Media Libraries (i.e. Changer Devices and Media Access Devices). The agent should utilize the same Durable Name techniques described for volumes in the Disk Array section. The chosen name is stored in the Name attribute of the logical device with the corresponding setting for the NameFormat attribute. Allowable name formats and device pairings for the storage library profile are:

- FCPort: FCPort.PermanentAddress = Fibre Channel Port World Wide Name. NameFormat should be set to “WWN”

- ChangerDevice.DeviceID = Vendor+Product+Serial Number+(optional instance number). Vendor, Model and Serial number should be taken from the ChangerDevice's associated ComputerSystem, Product, and/or Chassis. An option instance number may be added to uniquely denote more than one ChangerDevice "inside" a ComputerSystem
- MediaAccessDevice (or TapeDrive).DeviceID = Vendor+Product+Serial number for the MediaAccessDevice
- ComputerSystem.Name = Vendor+Product+Serial number for the storage library and/or its associated Product and Chassis. NameFormat should be set to "Vendor+Product+Serial"

Please refer to *Storage Management Technical Specification, Part 1 Common Architecture 7.6* for additional information.

4.2 Health and Fault Management Considerations

None

4.3 Cascading Considerations

None

4.4 Supported Subprofiles and Packages

Table 1: Supported Profiles for Storage Library

Registered Profile Names	Mandatory	Version
Access Points	No	1.2.0
Location	No	1.2.0
FC Target Ports	No	1.2.0
Software	No	1.2.0
Storage Library Limited Access Port Elements	No	1.2.0
Storage Library Media Movement	No	1.2.0
Storage Library Capacity	No	1.2.0
Storage Library Element Counting	No	1.2.0
Storage Library InterLibraryPort Connection	No	1.2.0
Storage Library Partitioned Library	No	1.2.0
Physical Package	Yes	1.2.0

4.5 Methods of this Profile

None

4.6 Client Considerations and Recipes

4.6.1 Recipe Overview

While no pseudo-code-based recipes have been written for this profile, this section provides some helpful information for writing management applications and suggests techniques for addressing common use cases.

4.6.2 Discover a Storage Media Library

Discovery of Storage Media Libraries is achieved by looking up instances of `ComputerSystem` which are subclassed from `System` and have a corresponding `Name` and `NameFormat` property as described above under 4.1.7. Specifically, `NameFormat` shall be set to “VendorModelSerial” and the `Name` shall be of the form Vendor+Product+Serial

4.6.3 Determine Library Physical Media Capacity

The physical media capacity of a library is the number of physical media objects that may be stored in the currently installed configuration of a Storage Media Library. This capacity may be determined by enumerating the `StorageMediaLocation` instances that are associated with each of the library’s `Chassis` objects.

In implementations that choose to include the Capacity subprofile, minimum and maximum slot capacities for a Storage Library are modeled in the `ConfigurationCapacity`, which is described earlier in the section on Capacity Constraints. Since this use case relies on an optional part of the profile, it may not be supported by each agent implementation.

4.6.4 Determine Physical Media Inventory

To determine the physical media inventory of a storage library, clients should discover the `Chassis` instance associated with a particular `ComputerSystem` (via the `ComputerSystemPackage` association), and enumerate the `PhysicalMedia` instances associated with the `Chassis` through the `PackagedComponent` association.

4.6.5 Discover Storage Library Control Type

The control mechanism to a library is either one of these:

- SCSI Media Changer Commands directed to the library’s changer device
- Library control commands directed to a Library Control service

If a library does not have a `ProtocolController` instance associated via `ProtocolControllerForUnit` to the `ChangerDevice` then the client should conclude that an alternate mechanism for controlling the library is required. This mechanism may vary, but should be represented by an instance of `Service` as described in the section on Software/Service View for a library’s hosted services

4.6.6 Determine Library Drive Capacity

The current drive capacity of a library may be determined by enumerating the `MediaAccessDevice` instances through the `SystemDevice` association of the library.

When the optional Capacity subprofile is implemented, the number of drives discovered should be within the range indicated by the minimum and maximum capacity attribute found on the library `Chassis`’ `ElementCapacity` association with `ConfigurationCapacity` for tape drives. This bounds check is not available if the Capacity subprofile is not implemented.

4.6.7 Determine Drive Data Path Technology

Clients can discover the data path protocol of each drive within a storage library by enumerating `MediaAccessDevice` instances, then following the `ProtocolControllerForUnit` association linking a `MediaAccessDevice` with a `ProtocolController`. Properties within Controller can then be queried for more information. If the `MediaAccessDevice` has a fibre channel interface, an `FCPort` instance is linked to its

ProtocolController by a ProtocolControllerForPort association. See *Storage Management Technical Specification, Part 2 Common Profiles* Clause 8: FC Target Ports Profile for more information on fibre channel connectivity.

4.6.8 Find asset Information

Information about the entire storage library is modeled in the Chassis instances associated with the ComputerSystem. Chassis properties include Manufacturer, Model, Version, and Tag. Tag is an arbitrary identifying string.

To identify asset information for the logical devices, a client should access the corresponding logical device through the ComputerSystem object's SystemDevice association. For each logical device instance the client may then check for asset information from the PhysicalElement associated through a Realizes association. Product information may also be available through the corresponding ProductPhysicalElement/ProductPhysicalComponent aggregation.

4.6.9 Discovery of Mailslots, Import/Export Elements or LimitedAccessPorts in a Storage Library

Clients may determine the number of LimitedAccessPorts in a library by enumerating the LimitedAccessPorts connected to a ComputerSystem instance via the SystemDevice association.

Note that some smaller libraries do not have the type of import/export element modeled by LimitedAccessPort. As a result, LimitedAccessPort elements are included in an (optional) subprofile (see Clause 9: Limited Access Port Elements Subprofile).

4.6.10 Counting assets in large storage libraries

Very large libraries may contain dozens of MediaAccessDevices and many thousands of StorageMediaLocations and PhysicalMedia. The intrinsic enumerateInstances() method is commonly used to count or gather CIM object instances of this type. Clients may find that using enumerateInstances() to count assets in very large libraries requires an excessive amount of time and processing resources. Providers supporting large libraries may also find that excessive time and resources are consumed attempting to return the bulk of data requested in enumerateInstances() calls. The following suggestions may be of help in situations where large libraries are of interest:

- Omit Qualifiers from enumerateInstances() or getInstance() requests;
- Request only the lowest-level child class of interest for examination or counting;
- Request only the properties of interest in enumerateInstances() or getInstance() requests. When only a count of existing objects is desired, omit all properties from the request;
- Use the intrinsic enumerateInstanceNames() or associatorNames() method instead of enumerateInstances() when only a count of existing objects is desired. The enumerateInstanceNames() and associatorNames() calls are much "lighter weight" overall than enumerateInstances();
- If the provider supports it, use the Physical Elements Count subprofile to quickly count PhysicalMedia and StorageMediaLocation instances. Note that this subprofile is optional and experimental and may not be supported by some providers.

4.7 Registered Name and Version

Storage Library version 1.2.0

4.8 CIM Elements

Table 2: CIM Elements for Storage Library

Element Name	Requirement	Description
CIM_ComputerSystem (4.8.1)	Mandatory	
CIM_ComputerSystem (4.8.2)	Mandatory	'Top level' system that represents the whole Storage Library.
CIM_ChangerDevice (4.8.3)	Mandatory	
CIM_MediaAccessDevice (4.8.4)	Mandatory	
CIM_PackagedComponent (4.8.5)	Mandatory	
CIM_PhysicalMedia (4.8.6)	Mandatory	
CIM_PhysicalMediaInLocation (4.8.7)	Mandatory	
CIM_StorageMediaLocation (4.8.8)	Mandatory	
CIM_Realizes (4.8.9)	Conditional	Conditional requirement: Support for Inter-Library Port profile.
CIM_Chassis (4.8.10)	Mandatory	
CIM_ElementSoftwareIdentity (4.8.11)	Mandatory	
CIM_ComputerSystemPackage (4.8.12)	Mandatory	
CIM_SystemDevice (4.8.13)	Conditional	Conditional requirement: Support for Inter-Library Port profile. This association links logicalDevices To the scoping system.
CIM_SCSIProtocolController (4.8.14)	Mandatory	
CIM_ProtocolControllerForUnit (4.8.15)	Mandatory	
CIM_SoftwareIdentity (4.8.16)	Mandatory	
CIM_ElementSoftwareIdentity (4.8.17)	Mandatory	
CIM_StorageLibraryCapabilities (4.8.18)	Optional	Describes the capabilities of the Storage Library represented by the top level ComputerSystem this is associated with
CIM_ElementCapabilities (4.8.19)	Optional	Class to implement the association between the top-level ComputerSystem representing a Storage Library and it's StorageLibraryCapabilities
SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_ComputerSystem	Mandatory	Creation of a storage library instance
SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_ComputerSystem	Mandatory	Deletion of a storage library instance
SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_PhysicalMedia	Mandatory	Creation of a physical media instance

Table 2: CIM Elements for Storage Library

Element Name	Requirement	Description
SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_PhysicalMedia	Mandatory	Deletion of a physical media instance
SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_MediaAccessDevice	Mandatory	Creation of a media access device instance
SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_MediaAccessDevice	Mandatory	Deletion of a media access device instance
SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_ChangerDevice	Mandatory	Creation of a Changer Device instance
SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_ChangerDevice	Mandatory	Deletion of a Changer Device instance
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ComputerSystem AND PreviousInstance.OperationalStatus <> SourceInstance.OperationalStatus	Mandatory	Deprecated WQL - Change in OperationalStatus of a storage library
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_MediaAccessDevice AND PreviousInstance.OperationalStatus <> SourceInstance.OperationalStatus	Mandatory	Deprecated WQL - Change in OperationalStatus for a media access device
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ChangerDevice AND PreviousInstance.OperationalStatus <> SourceInstance.OperationalStatus	Mandatory	Deprecated WQL - Change in OperationalStatus for a Changer Device
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ComputerSystem AND PreviousInstance.CIM_ComputerSystem::OperationalStatus <> SourceInstance.CIM_ComputerSystem::OperationalStatus	Optional	Experimental CQL - Change in OperationalStatus of a storage library
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_MediaAccessDevice AND PreviousInstance.CIM_MediaAccessDevice::OperationalStatus <> SourceInstance.CIM_MediaAccessDevice::OperationalStatus	Optional	Experimental CQL - Change in OperationalStatus for a media access device

Table 2: CIM Elements for Storage Library

Element Name	Requirement	Description
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ChangerDevice AND PreviousInstance.CIM_ChangerDevice::OperationalStatus <> SourceInstance.CIM_ChangerDevice::OperationalStatus	Optional	Experimental CQL - Change in OperationalStatus for a Changer Device

4.8.1 CIM_ComputerSystem

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 3 describes class CIM_ComputerSystem.

Table 3: SMI Referenced Properties/Methods for CIM_ComputerSystem

Properties	Flags	Requirement	Description & Notes
CreationClassName		Mandatory	
Name		Mandatory	

4.8.2 CIM_ComputerSystem

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 4 describes class CIM_ComputerSystem.

Table 4: SMI Referenced Properties/Methods for CIM_ComputerSystem

Properties	Flags	Requirement	Description & Notes
CreationClassName		Mandatory	

Table 4: SMI Referenced Properties/Methods for CIM_ComputerSystem

Properties	Flags	Requirement	Description & Notes
Name		Mandatory	Unique identifier for the storage library. This should take the form of a string consisting of Vendor+Product+SerialNumber, derived from SCSI Inquiry Pages.
Dedicated		Mandatory	Indicates that this computer system is dedicated to operation as a storage library
NameFormat		Mandatory	Format for Name property. HID is a required format. Others are optional.
OperationalStatus		Mandatory	Overall status of the library
StatusDescriptions		Optional	Additional information related to the values in OperationalStatus.
ElementName		Mandatory	User friendly name
PrimaryOwnerContact	M	Optional	Contact details for storage library owner
PrimaryOwnerName	M	Optional	Owner of the storage library

4.8.3 CIM_ChangerDevice

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 5 describes class CIM_ChangerDevice.

Table 5: SMI Referenced Properties/Methods for CIM_ChangerDevice

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	
CreationClassName		Mandatory	
SystemName		Mandatory	
DeviceID		Mandatory	
MediaFlipSupported		Mandatory	
ElementName		Mandatory	
OperationalStatus		Mandatory	Status of the changer device.

Table 5: SMI Referenced Properties/Methods for CIM_ChangerDevice

Properties	Flags	Requirement	Description & Notes
StatusDescriptions		Optional	Additional information related to the values in OperationalStatus.

4.8.4 CIM_MediaAccessDevice

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 6 describes class CIM_MediaAccessDevice.

Table 6: SMI Referenced Properties/Methods for CIM_MediaAccessDevice

Properties	Flags	Requirement	Description & Notes
SystemCreationClass sName		Mandatory	
CreationClassName		Mandatory	
SystemName		Mandatory	
DeviceID		Mandatory	
OperationalStatus		Mandatory	
StatusDescriptions		Optional	Additional information related to the values in OperationalStatus.
NeedsCleaning		Mandatory	If unknown, set to False.
MountCount		Mandatory	

4.8.5 CIM_PackagedComponent

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 7 describes class CIM_PackagedComponent.

Table 7: SMI Referenced Properties/Methods for CIM_PackagedComponent

Properties	Flags	Requirement	Description & Notes
GroupComponent		Mandatory	
PartComponent		Mandatory	

4.8.6 CIM_PhysicalMedia

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 8 describes class CIM_PhysicalMedia.

Table 8: SMI Referenced Properties/Methods for CIM_PhysicalMedia

Properties	Flags	Requirement	Description & Notes
CreationClassName		Mandatory	
Tag		Mandatory	
Capacity		Mandatory	0 = unknown. If CleanerMedia=True, then ignore Capacity value.
MediaType		Mandatory	
MediaDescription		Optional	
CleanerMedia		Mandatory	If unknown, set to False
DualSided		Mandatory	
LabelStates		Mandatory	
LabelFormats		Mandatory	
PhysicalLabels		Mandatory	
RemovalConditions		Mandatory	

4.8.7 CIM_PhysicalMediaInLocation

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 9 describes class CIM_PhysicalMediaInLocation.

Table 9: SMI Referenced Properties/Methods for CIM_PhysicalMediaInLocation

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	
Dependent		Mandatory	

4.8.8 CIM_StorageMediaLocation

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 10 describes class CIM_StorageMediaLocation.

Table 10: SMI Referenced Properties/Methods for CIM_StorageMediaLocation

Properties	Flags	Requirement	Description & Notes
CreationClassName		Mandatory	
Tag		Mandatory	
LocationType		Mandatory	
LocationCoordinates		Mandatory	
MediaTypesSupported		Mandatory	
MediaCapacity		Mandatory	

4.8.9 CIM_Realizes

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: InterLibraryPort

Table 11 describes class CIM_Realizes.

Table 11: SMI Referenced Properties/Methods for CIM_Realizes

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	
Dependent		Mandatory	

4.8.10 CIM_Chassis

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 12 describes class CIM_Chassis.

Table 12: SMI Referenced Properties/Methods for CIM_Chassis

Properties	Flags	Requirement	Description & Notes
CreationClassName		Mandatory	
Tag		Mandatory	
LockPresent		Mandatory	
SecurityBreach		Mandatory	
IsLocked		Mandatory	
ElementName		Mandatory	
Manufacturer		Mandatory	
Model		Mandatory	
SerialNumber		Mandatory	

4.8.11 CIM_ElementSoftwareIdentity

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 13 describes class CIM_ElementSoftwareIdentity.

Table 13: SMI Referenced Properties/Methods for CIM_ElementSoftwareIdentity

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	
Dependent		Mandatory	

4.8.12 CIM_ComputerSystemPackage

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 14 describes class CIM_ComputerSystemPackage.

Table 14: SMI Referenced Properties/Methods for CIM_ComputerSystemPackage

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	
Dependent		Mandatory	

4.8.13 CIM_SystemDevice

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: InterLibraryPort

Table 15 describes class CIM_SystemDevice.

Table 15: SMI Referenced Properties/Methods for CIM_SystemDevice

Properties	Flags	Requirement	Description & Notes
PartComponent		Mandatory	
GroupComponent		Mandatory	

4.8.14 CIM_SCSIProtocolController

This is only required if FC Ports claim backwards compatibility with SMI-S 1.0

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 16 describes class CIM_SCSIProtocolController.

Table 16: SMI Referenced Properties/Methods for CIM_SCSIProtocolController

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	
SystemName		Mandatory	
CreationClassName		Mandatory	
DeviceID		Mandatory	Opaque identifier
ElementName		Optional	
OperationalStatus		Mandatory	
StatusDescriptions		Optional	Additional information related to the values in OperationalStatus.
MaxUnitsControlled		Optional	

4.8.15 CIM_ProtocolControllerForUnit

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 17 describes class CIM_ProtocolControllerForUnit.

Table 17: SMI Referenced Properties/Methods for CIM_ProtocolControllerForUnit

Properties	Flags	Requirement	Description & Notes
DeviceNumber		Optional	The target device visible through the controller.
Antecedent		Mandatory	Reference to MediaAccessDevice or ChangerDevice
Dependent		Mandatory	

4.8.16 CIM_SoftwareIdentity

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 18 describes class CIM_SoftwareIdentity.

Table 18: SMI Referenced Properties/Methods for CIM_SoftwareIdentity

Properties	Flags	Requirement	Description & Notes
InstanceID		Mandatory	
VersionString		Mandatory	The software of firmware version of the device (ChangerDevice, MediaAccessDevice, or a SCSIProtocolController)
Manufacturer		Mandatory	
Classifications		Optional	4 = Application Software, 10 = Firmware
BuildNumber		Optional	
MajorVersion		Optional	
RevisionNumber		Optional	
MinorVersion		Optional	

4.8.17 CIM_ElementSoftwareIdentity

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 19 describes class CIM_ElementSoftwareIdentity.

Table 19: SMI Referenced Properties/Methods for CIM_ElementSoftwareIdentity

Properties	Flags	Requirement	Description & Notes
Dependent		Mandatory	
Antecedent		Mandatory	

4.8.18 CIM_StorageLibraryCapabilities

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Optional

Table 20 describes class CIM_StorageLibraryCapabilities.

Table 20: SMI Referenced Properties/Methods for CIM_StorageLibraryCapabilities

Properties	Flags	Requirement	Description & Notes
InstanceID		Mandatory	Unique Identifier for this Capabilities class. See MOF for specific format
ElementName		Mandatory	A user friendly name
Capabilities		Optional	Array of general capabilities for the Storage Library (see MOF)
MaxAuditTime		Optional	Number of seconds it takes for the library to complete an audit or "inventory" operations.

4.8.19 CIM_ElementCapabilities

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Optional

Table 21 describes class CIM_ElementCapabilities.

Table 21: SMI Referenced Properties/Methods for CIM_ElementCapabilities

Properties	Flags	Requirement	Description & Notes
Capabilities		Mandatory	
ManagedElement		Mandatory	

STABLE

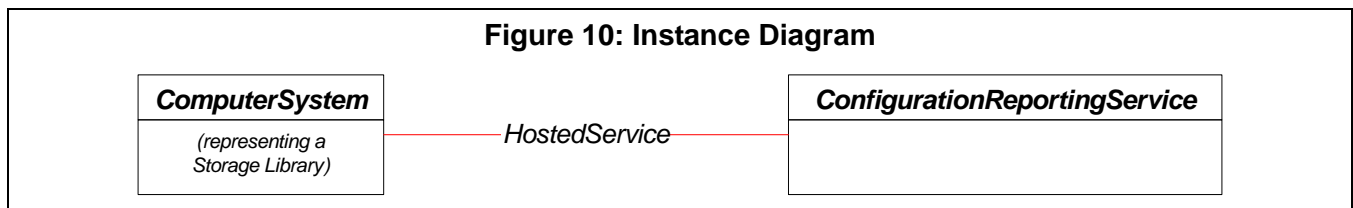
EXPERIMENTAL

Clause 5: Element Counting Subprofile

5.1 Description

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 ConfigurationReportingService hosted by the storage library's (or other system type's) top-level ComputerSystem.

Figure 10 provides a sample instance diagram.



5.1.1 Discovery

The Element counting subprofile, as currently defined, is not an advertised profile. Support for the Element Counting Subprofile can be obtained through the Storage Library Profile (or other top-level system profile as appropriate).

5.2 Health and Fault Management Considerations

Not defined in this standard.

5.3 Cascading Considerations

Not defined in this standard.

5.4 Supported Subprofiles and Packages

The Element counting subprofile requires the Storage Library profile. Other top-level device profiles may also be able to make use of this subprofile, but such compatibility is not guaranteed.

5.5 Methods of the Profile

5.5.1 GetClassTypes

GetClassTypes returns the list of class types that a given ManagedElement – typically, a storage library's top-level ComputerSystem or Chassis – supports or has installed. Calling GetClassTypes is the first step in a three step process to obtain a count of desired elements. (See 5.6 Client Considerations and Recipes for an overview and example).

The GetClassTypes method uses the following parameters:

[IN] uint16 InquiryType = “Installed” or “Supports”

When “Installed” is specified, the method will return the list of countable classes that the associated ComputerSystem currently has installed or contained within its scope. When “Supports” is specified, the method will return the list of countable classes that the associated ComputerSystem potentially supports, though no such class instances may currently be installed or contained within its scope.

[IN] boolean Recursive = true or false

For the purposes of the current subprofile, the value of the Recursive parameter is not relevant. Until defined otherwise, clients should specify “false”, and expect that the value will not affect operation of the GetClassTypes method in any way.

[IN] CIM_ManagedElement REF Target = a CIM object pointer to the to the top-level ComputerSystem to which ConfigurationReportingService is associated.

In some cases, a pointer the ComputerSystem’s Chassis may be appropriate. This parameter reinforces that the ConfigurationReportingService is returning information on the storage library’s (or other top-level profile’s) ComputerSystem or Chassis. Classes to be returned or counted are considered to be uniquely within the scope of this top-level ComputerSystem or Chassis.

[IN (false), OUT] string ClassTypes[] = an array of class types that can be counted by the service.

One value of this parameter will be selected by the client and used when calling GetUnitTypes() and ReportCapacity(), described below. The method/service provider may return a string representation of any valid CIM class which it can report a count on. For example, a storage library provider might return “CIM_PhysicalMedia” to indicate that this service allows clients to obtain a count of PhysicalMedia instances currently associated with the Target ComputerSystem or Chassis instance. Other example values would be “CIM_StorageMediaLocation” and “CIM_MediaAccessDevice”

The GetClassTypes method also returns one of the following status values:

“Success”, “Not Supported”, “Unknown”, “Timeout”, “Failed”, “DMTF Reserved”, “Vendor Specific”. In general, it is expected that “Success” will be returned on successful execution and “Failed” or “Timeout” will be returned when errors occur in executing this method on the provider/server side. If “Not Supported” is returned, the client may still attempt to call the GetUnitTypes and ReportCapacity methods, but a known value for the ClassType parameter will not be available to the client up front. “Unknown” indicates that the result cannot be determined for the given parameter combination at this time.

5.5.2 GetUnitTypes

GetUnitTypes returns the type of “unit” relationships that can be specified by the client when counting class instances associated with a top-level ComputerSystem or Chassis. Calling GetUnitTypes in the second step in a three step process to obtain a count of desired elements. (See 5.6 Client Considerations and Recipes for an overview and example).

The GetUnitTypes method uses many of the same parameters as GetClassTypes, including:

[IN] uint16 InquiryType: see details in 5.5.1 GetClassTypes. “Supported” or “Installed” are valid enumerated values.

[IN] boolean Recursive: see details in 5.5.1 GetClassTypes. Generally, a value of “false” is expected.

[IN] CIM_ManagedElement REF Target: see details in 5.5.1 GetClassTypes. A pointer to the top-level ComputerSystem associated with this ConfigurationReportingService. In some cases, a pointer to the top-level Chassis may be appropriate.

[IN] string ClassType: see details in 5.5.1 GetClassTypes. The class type to be counted.

[IN (false) OUT] uint16 UnitTypes[] = an array of “relationship types” to help specify how the class instances to be counted are associated with the top-level ComputerSystem or Chassis specified by Target. Many values are available for UnitTypes, but clients should expect that only “Contained” or “Connected” will be returned by storage library providers. Other values, such as “None”, “Front Side”, and “Memory” should not be returned until future definition of their meaning is documented. Clients will use one of the values returned in this parameter when calling ReportCapacity.

The GetUnitTypes method also returns one of the following status values:

“Success”, “Not Supported”, “Unknown”, “Timeout”, “Failed”, “DMTF Reserved”, “Vendor Specific”. In general, it is expected that “Success” will be returned on successful execution and “Failed” or “Timeout” will be returned when errors occur in executing this method on the provider/server side. If “Not Supported” is returned, the client may still attempt to call the ReportCapacity method, but a known value for the UnitType parameter will not be available to the client up front. In general, clients should attempt to specify “Contained” or “Connected” when calling ReportCapacity. “Unknown” indicates that the result cannot be determined for the given parameter combination at this time.

5.5.3 ReportCapacity

ReportCapacity returns the number or count of a given class types that the given ManagementElement – typically, a storage library’s top-level ComputerSystem or Chassis – supports or has installed. Calling ReportCapacity in the third step in a three step process to obtain a count of desired elements. (See 5.6 Client Considerations and Recipes for an overview and example).

The ReportCapacity method uses many of the same parameters as GetClassTypes and GetUnitTypes, including:

[IN] uint16 InquiryType: see details in 5.5.1 GetClassTypes. “Supported” or “Installed” are valid enumerated values.

[IN] boolean Recursive: see details in 5.5.1 GetClassTypes. Generally, a value of “false” is expected.

[IN] CIM_ManagedElement REF Target: in 5.5.1 GetClassTypes. A pointer to the top-level ComputerSystem associated with this ConfigurationReportingService. In some cases, a pointer to the top-level Chassis may be appropriate.

[IN] string ClassType: see details in 5.5.1 GetClassTypes. The class type to be counted.

[IN] uint16 UnitType: see details in 5.5.1 GetClassTypes. Generally, the “Contained” or “Connected” enumerated value will be used.

[IN (false), OUT] uint64 NumberOfUnits = the number of “supported” or “installed” ClassType instances “contained” or “connected” in a given Target ComputerSystem’s (or Chassis’s) scope. Obtaining this count is the purpose of the ConfigurationReportingService.

The ReportCapacity method also returns one of the following status values:

“Success”, “Not Supported”, “Unknown”, “Timeout”, “Failed”, “DMTF Reserved”, “Vendor Specific”. In general, it is expected that “Success” will be returned on successful execution and “Failed” or “Timeout” will be returned when errors occur in executing this method on the provider/server side. If “Not Supported” is returned, it may indicate that the Target, ClassType, or UnitType parameters are in error. Supported values for ClassType and UnitType should be obtained by calling GetClassTypes and GetUnitTypes prior to calling ReportCapacity.

5.6 Client Considerations and Recipes

ConfigurationReportingService may be used by clients interested in quickly obtaining a count or “number of” desired instances. For example, a client may want to know the number of PhysicalMedia instances associated with a particular storage library, but the time and overhead associated with enumerating the instances of these objects – through the extrinsic enumerateInstances() or enumerateInstanceNames() methods – can be excessive.

To use ConfigurationReportingService, clients call three methods in succession: GetClassTypes, GetUnitTypes, and ReportCapacity. GetClassTypes returns the list of class types that can be counted. This information is then used to call GetUnitTypes, which returns a list of “unit” relationships (e.g. “Connected” or “Contained”). This value and other information is then passed to ReportCapacity, which returns the count of desired class instances.

An example: A client wants to count the number of PhysicalMedia instances associated with a storage library (itself represented by a top-level ComputerSystem and Chassis instance). Having discovered a ConfigurationReportingService associated with the ComputerSystem of interest, the client will call:

```
uint32 GetClassTypes (
    InquiryType = "Installed",
    Recursive = "false",
    Target = CIM object path to the ComputerSystem of interest,
    &ClassTypes[] = pointer to the countable classes, as returned by the
                    provider/service)
```

Assuming that GetClassTypes returns a value of “Success”, the client may examine the ClassTypes[] array and find that it contains “CIM_MediaAccessDevice”, “CIM_PhysicalMedia”, “CIM_StorageMediaLocation”, and “CIM_MediaTransferDevice”. Since this client is interested in PhysicalMedia, it would use the “CIM_PhysicalMedia” value use to call GetUnitTypes:

```
uint32 GetUnitTypes (
    InquiryType = "Installed",
    Recursive = "false",
    Target = CIM object path to the ComputerSystem of interest,
    ClassType = "CIM_PhysicalMedia"
    &UnitTypes[] = pointer to the supported “unit” relationship types, as
                    returned by the provider/service)
```

Assuming that GetUnitTypes returns a value of “Success”, the client may examine the UnitTypes[] array and find that it contains only “Contained”. The client would then use this value to call ReportCapacity:

```
uint32 ReportCapacity (
    InquiryType = "Installed",
    Recursive = "false",
    Target = CIM object path to the ComputerSystem of interest,
    ClassType = "CIM_PhysicalMedia",
    UnitType = "Contained"
    &NumberOfUnits)
```

Assuming that ReportCapacity returns a value of “Success”, the client should examine the NumberOfUnits value to determine the number of CIM_PhysicalMedia “contained” or currently “installed” in the Target ComputerSystem.

In general, it is expected that “Success” will be returned on successful execution of these three methods, and “Failed” or “Timeout” will be returned when errors occur in executing these methods on the provider/server side. If “Not Supported” is returned, it may indicate that the Target, ClassType, or UnitType parameters are in error.

5.7 Registered Name and Version

Storage Library Element Counting version 1.1.0

5.8 CIM Elements

Table 22: CIM Elements for Storage Library Element Counting

Element Name	Requirement	Description
CIM_ConfigurationReportingService (5.8.1)	Mandatory	
CIM_HostedService (5.8.2)	Mandatory	

5.8.1 CIM_ConfigurationReportingService

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 23 describes class CIM_ConfigurationReportingService.

Table 23: SMI Referenced Properties/Methods for CIM_ConfigurationReportingService

Properties	Flags	Requirement	Description & Notes
SystemCreationClass Name		Mandatory	
CreationClassName		Mandatory	
SystemName		Mandatory	
Name		Mandatory	
GetClassTypes()		Mandatory	
GetUnitTypes()		Mandatory	
ReportCapacity()		Mandatory	

5.8.2 CIM_HostedService

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 24 describes class CIM_HostedService.

Table 24: SMI Referenced Properties/Methods for CIM_HostedService

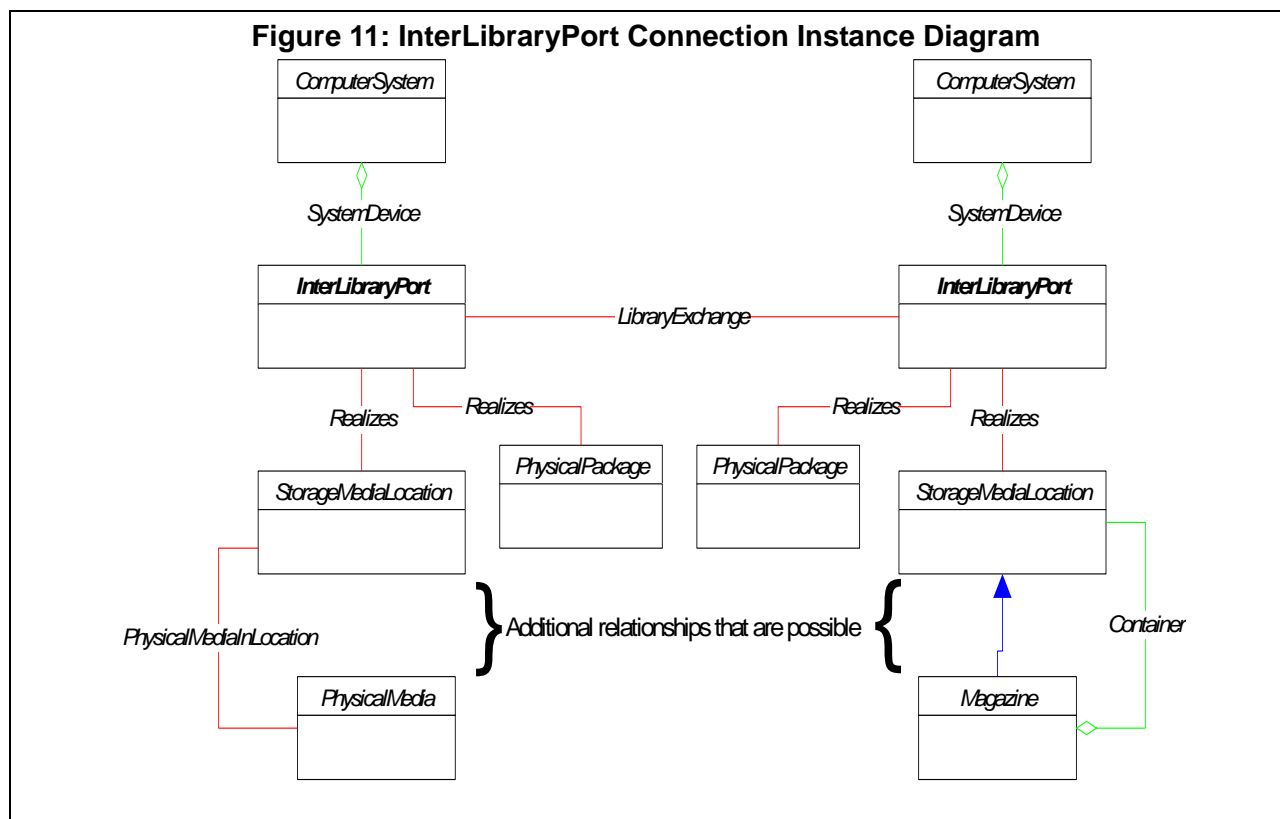
Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	
Dependent		Mandatory	

EXPERIMENTAL

EXPERIMENTAL
Clause 6: InterLibraryPort Connection Subprofile
6.1 Description

Support of InterLibraryPort devices, a.k.a. pass-thru ports or cartridge exchange mechanisms, is designated as optional in this profile. However, when such a device exists the agent representing the library should instantiate this class for each port. When one or more libraries are connected via an Inter-Library Port and the corresponding agents are working with separate name spaces a mechanism is required for correlating the LibraryExchange association that represents the port connection.

Figure 11 provides a sample instance diagram.


Durable Names and Correlatable IDs

A Durable Name is not defined by this profile for InterLibraryPort instances and remains unspecified. This is not an issue when associated InterLibraryPort instances are within the same name space.

6.2 Health and Fault Management Considerations

Not defined in this standard.

6.3 Cascading Considerations

Not defined in this standard.

6.4 Supported Subprofiles and Packages

None.

6.5 Methods of the Profile

None.

6.6 Client Considerations and Recipes

None.

6.7 Registered Name and Version

Storage Library InterLibraryPort Connection version 1.1.0

6.8 CIM Elements

Table 25: CIM Elements for Storage Library InterLibraryPort Connection

Element Name	Requirement	Description
CIM_InterLibraryPort (6.8.1)	Mandatory	InterLibraryPorts represent hardware that transports Physical Media between connected Storage Libraries. The LibraryExchange association identifies the connected Libraries, by identifying the connected InterLibraryPorts.
CIM_LibraryExchange (6.8.2)	Mandatory	This relationship identifies that two storage libraries are connected through their InterLibraryPorts.
SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_InterLibraryPort	Mandatory	Creation of an instance of InterLibraryPort
SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_InterLibraryPort	Mandatory	Deletion of an instance of InterLibraryPort
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_InterLibraryPort AND SourceInstance.OperationalStatus <> PreviousInstance.OperationalStatus	Mandatory	Deprecated WQL - Change in OperationalStatus of a InterLibraryPort
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_InterLibraryPort AND SourceInstance.CIM_InterLibraryPort::OperationalStatus <> PreviousInstance.CIM_InterLibraryPort::OperationalStatus	Optional	Experimental CQL - Change in OperationalStatus of a InterLibraryPort

6.8.1 CIM_InterLibraryPort

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 26 describes class CIM_InterLibraryPort.

Table 26: SMI Referenced Properties/Methods for CIM_InterLibraryPort

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	
CreationClassName		Mandatory	
SystemName		Mandatory	
DeviceID		Mandatory	
LastAccessed		Mandatory	Last access time of the port by the library
ImportCount		Mandatory	The number of times the port was used to move physical media into the storage library
ExportCount		Mandatory	The number of times the port was used to move physical media out of the storage library
Direction		Mandatory	Identifies whether the port can be used to import physical media, export physical media or both
OperationalStatus		Mandatory	Status of the InterLibrary port.
StatusDescriptions		Optional	Additional information related to the values in OperationalStatus.

6.8.2 CIM_LibraryExchange

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 27 describes class CIM_LibraryExchange.

Table 27: SMI Referenced Properties/Methods for CIM_LibraryExchange

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	

Table 27: SMI Referenced Properties/Methods for CIM_LibraryExchange

Properties	Flags	Requirement	Description & Notes
Dependent		Mandatory	

EXPERIMENTAL

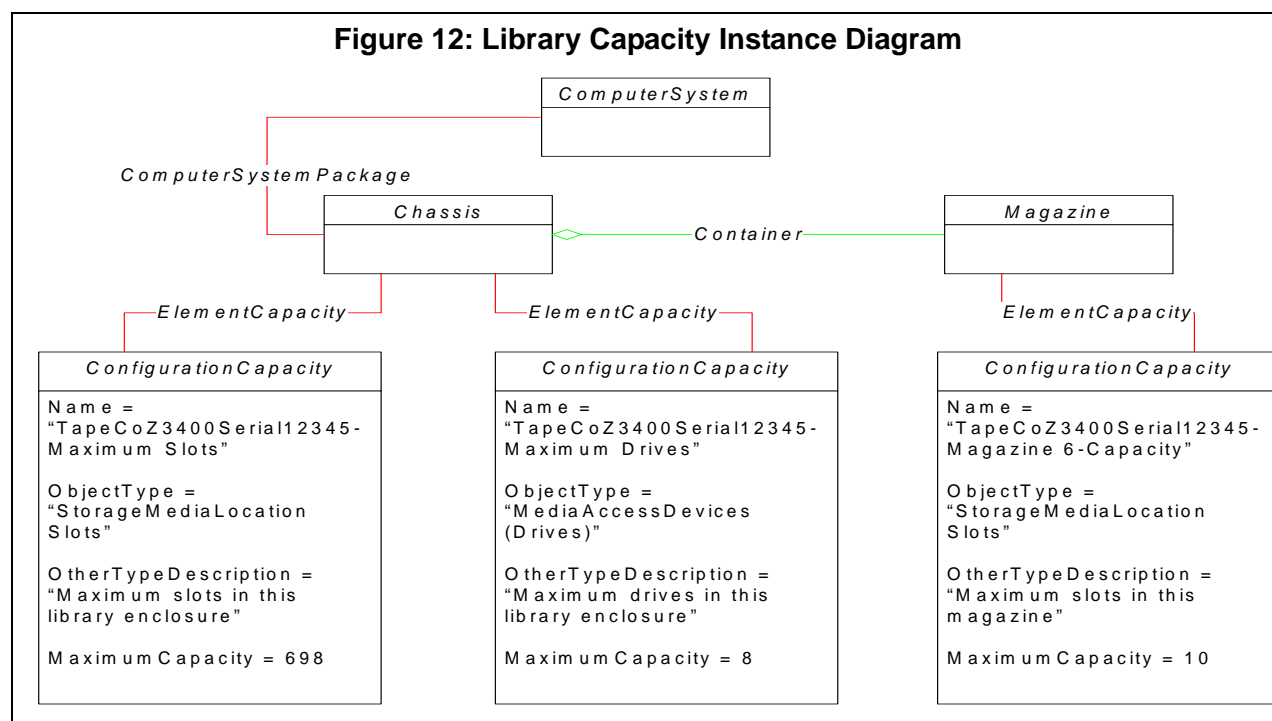
EXPERIMENTAL

Clause 7: Library Capacity Subprofile

7.1 Description

By adding two classes (ConfigurationCapacity and ElementCapacity) servers can publish the minimum and maximum number of slots, drives, magazines, media changers, and other elements associated with a given storage library.

Figure 12 illustrates the use of ConfigurationCapacity and ElementCapacity in conjunction with the basic storage library profile.



7.2 Health and Fault Management Considerations

Not defined in this standard.

7.3 Cascading Considerations

Not defined in this standard.

7.4 Supported Subprofiles and Packages

None.

7.5 Client Considerations and Recipes

None.

7.6 Registered Name and Version

Storage Library Capacity version 1.1.0

7.7 CIM Elements

Table 28: CIM Elements for Storage Library Capacity

Element Name	Requirement	Description
CIM_ConfigurationCapacity (7.7.1)	Mandatory	ConfigurationCapacity provides information on the minimum and maximum number of slots, drives, magazines, media changers, and other elements associated with a given storage library.
CIM_ElementCapacity (7.7.2)	Mandatory	

7.7.1 CIM_ConfigurationCapacity

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 29 describes class CIM_ConfigurationCapacity.

Table 29: SMI Referenced Properties/Methods for CIM_ConfigurationCapacity

Properties	Flags	Requirement	Description & Notes
Name		Mandatory	
ObjectType		Mandatory	Other, Processors, Power Supplies, see MOF
OtherTypeDescription		Optional	
MinimumCapacity		Mandatory	
MaximumCapacity		Mandatory	

7.7.2 CIM_ElementCapacity

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 30 describes class CIM_ElementCapacity.

Table 30: SMI Referenced Properties/Methods for CIM_ElementCapacity

Properties	Flags	Requirement	Description & Notes
Element		Mandatory	
Capacity		Mandatory	

EXPERIMENTAL

EXPERIMENTAL

Clause 8: LibraryAlert Events/Indications for Library Devices

8.1 Description

Historically, media libraries have been managed using both SCSI and SNMP interfaces. A number of library management standards have been defined based on these interfaces, including the “TapeAlert” error events flags. These events alert subscribing clients to current or pending error conditions related to a library, drives, or media. The SCSI implementation of TapeAlert is described in the SCSI Stream Commands (SSC-2) and SCSI Media Changer Commands (SMC-2) specifications.

In order to carry these useful asynchronous events into the WBEM/CIM domain, the TapeAlert events have been mapped into instances of the AlertIndication class. This CIM class provides a general means for communicating asynchronous events to subscribing clients and TapeAlert events/indications -- hereafter referred to more generally as “LibraryAlert” indications -- shall be specified by filling in standard values for the properties of an AlertIndication.

8.2 Health and Fault Management Considerations

Not defined in this standard.

8.3 Cascading Considerations

Not defined in this standard.

8.4 Supported Subprofiles and Packages

None.

8.5 Methods of the Profile

None.

8.6 Client Considerations and Recipes

For all LibraryAlert indications, the following properties of AlertIndication shall be static and set to the values shown in Table 31.

Table 31: LibraryAlert Property Settings

Property Name	Property type	Property Value
Description	string	“LibraryAlert Indication”
AlertType	Uint16 (enumeration)	5 = “Device Alert”
ProableCause	Uint16 (enumeration)	1 = “other”
Trending	Uint16 (enumeration)	1 = “Not Applicable”
SystemCreationClassName	string	“CIM_ComputerSystem”

Clients may identify a received AlertIndication as a LibraryAlert indication primarily by the value of "LibraryAlert Indication" in the Description property. The following Query attribute on an IndicationFilter instance should be provided by the agent for these alerts:

```
SELECT * FROM CIM_Alert
WHERE Description="LibraryAlert Indication"
```

The following AlertIndication properties for LibraryAlert indications shall be vendor-specific and no specification or restriction of values is made here:

Table 32: Vendor Specific Properties of LibraryAlert

Property Name	Property type	Property Value
OtherSeverity	string	specified by vendor
EventID	string	specified by vendor
ProviderName	string	specified by vendor

A small number of AlertIndication properties for LibraryAlert indications shall have variable values that are restricted within a small range, as follows:

Table 33: Variable Alert Properties for LibraryAlert

Property Name	Property type	Property Value
SystemName	string	Name property value for the StorageLibrary instance that is associated with this unique indication
AlertingManagedElement	string	CIMInstance in string format for element to which this indication applies: MediaAccessDevice, StorageLibrary, or PhysicalMedia

The remaining AlertIndication properties for LibraryAlert indications shall have values derived from the SCSI TapeAlert specifications: SCSI Stream Commands (SSC-2) and SCSI Media Changer Commands (SMC-2).

Note that a small number of indications apply only to Tape libraries, while all other indications apply generically to any library type. Those indications that are tape-specific may be identified by the following strings in the OtherAlertType property:

Table 34: SCSI TapeAlert-based Properties

Property Name	Property type	Property Value
OtherAlertType	string	"Tape snapped/cut in the drive where media can be de-mounted."
OtherAlertType	string	"Tape snapped/cut in the drive where media cannot be de-mounted."
OtherAlertType	string	"The drive is having severe trouble reading or writing, which will be resolved by a retention cycle."

The remaining AlertIndication properties and values for all LibraryAlert indications are shown in Table 35. Note that the OtherAlertType property, in particular, serves to uniquely identify each of the LibraryAlert indications.

Table 35: LibraryAlert AlertIndication Properties

Event/Alert Summary	AlertIndication “Mapped” Properties from SSC-2 and SMC-2 Specs			
	OtherAlert Type	Perceived Severity	ProbableCause Description	Recommended Action[]
	string	Uint16	string	string
Read Warning	“The drive is having severe trouble reading.”	“3” = “Degraded/Warning”	“The drive is having problems reading data. No data has been lost, but there has been a reduction in the performance.”	
Write Warning	“The drive is having severe trouble writing.”	“4” = “Warning”	“Worn out Media”	“1. Discard the worn out media” “2. Use a new cleaning media”
Hard Error	“The drive had a hard read or write error.”	“5” = “Warning”	“Bad Media or Drive. The operation has stopped because an error has occurred while reading or writing data that the drive cannot correct.”	
Media	“Media can no longer be written/read, or performance is severely degraded.”	“6” = “Critical”	“Bad Media”	“1. Copy any data you require from this media.” “2. Do not use this media again.” “3. Restart the operation with a different media.”
Read Failure	“The drive can no longer read data from the storage media.”	“6” = “Critical”	“Worn out media”	“1. Replace media.” “2. Call the drive supplier help line.”
Write Failure	“The drive can no longer write data to the media.”	“6” = “Critical”	“The media is from a faulty batch or the drive is faulty: “	“1. Use known-good media to test the drive. “ “2. If the problem persists, call the media drive supplier”

Table 35: LibraryAlert AlertIndication Properties (Continued)

Event/Alert Summary	AlertIndication “Mapped” Properties from SSC-2 and SMC-2 Specs			
	OtherAlert Type	Perceived Severity	ProbableCause Description	Recommended Action[]
	string	Uint16	string	string
Media Life	“The media has exceeded its specified life.”	“3” = “Degraded/Warning”	“The media has reached the end of its calculated useful life:”	“1. Copy any data you need to another media.” 2. Discard the old media.”
Not Data Grade	“The cartridge is not data-grade. Any data you write to the media is at risk. Replace the cartridge with a data-grade media.”	“3” = “Degraded/Warning”	“The cartridge is not data-grade. Any data you write to the media is at risk.”	“Replace the cartridge with a data-grade media.”
Write Protect	“Write command is attempted to a write protected media.”	“6” = “Critical”	“Replace with writable media”	“You are trying to write to a write protected cartridge. Remove the write protection or use another media.”
No Removal	“Manual or software unload attempted when prevent media removal is on.”	“2” = “Information”	“Wait until drive is not in-use”	“You cannot eject the cartridge because the drive is in use. Wait until the operation is complete before ejecting the cartridge.”
Cleaning Media	“Cleaning media loaded into drive”	“2” = “Information”	“The media in the drive is a cleaning cartridge.”	“Replace this media with writeable media”
Unsupported Format	“Attempted load of unsupported media format (e.g., DDS2 in DDS1 drive).”	“2” = “Information”	“You have tried to load a cartridge of a type that is not supported by this drive.”	“Insert media of a type supported by this drive”
Recoverable Snapped Tape	“Tape snapped/cut in the drive where media can be de-mounted.”	“6” = “Critical”	“The operation has failed because the tape in the drive has snapped.”	“1. Discard the old tape.” “2. Restart the operation with a different tape.”

Table 35: LibraryAlert AlertIndication Properties (Continued)

Event/Alert Summary	AlertIndication “Mapped” Properties from SSC-2 and SMC-2 Specs			
	OtherAlert Type	Perceived Severity	ProbableCause Description	Recommended Action[]
	string	Uint16	string	string
Unrecoverable Snapped Tape	“Tape snapped/cut in the drive where media cannot be de-mounted.”	“6” = “Critical”	“The operation has failed because the tape in the drive has snapped.”	“1. Do not attempt to extract the tape cartridge.” “2. Call the tape drive supplier help line.”
Memory Chip In Cartridge Failure	“Memory chip failed in cartridge.”	“3” = “Degraded/Warning”	“The memory in the media has failed, which reduces performance.”	“Do not use the cartridge for further write operations.”
Forced Eject	“Manual or forced eject while drive actively writing or reading.”	“6” = “Critical”	“The operation has failed because the media was manually de-mounted while the drive was actively writing or reading.”	
Read Only Format	“Media loaded that is read-only format.”	“3” = “Degraded/Warning”	“You have loaded a cartridge of a type that is read-only in this drive. The cartridge will appear as write protected.”	
Directory Corrupted On Load	“Drive powered down while loaded, or permanent error prevented the directory being updated.”	“3” = “Degraded/Warning”	“The directory on the cartridge has been corrupted. File search performance will be degraded. “	“The directory can be rebuilt by reading all the data on the cartridge.”
Nearing Media Life	“Media may have exceeded its specified number of passes.”	“2” = “Information”	“The storage media is nearing the end of its calculated life.”	“1. Use another storage media for your next backup.” “2. Store this storage media in a safe place in case you need to restore data from it.”

Table 35: LibraryAlert AlertIndication Properties (Continued)

Event/Alert Summary	AlertIndication "Mapped" Properties from SSC-2 and SMC-2 Specs			
	OtherAlert Type	Perceived Severity	ProbableCause Description	Recommended Action[]
	string	Uint16	string	string
Clean Now	"The drive thinks it has a head clog or needs cleaning."	"6" = "Critical"	"The drive needs cleaning."	<p>"1. If the operation has stopped, eject the storage media and clean the drive."</p> <p>"2. If the operation has not stopped, wait for it to finish and then clean the drive. Check the drive user's manual for device specific cleaning"</p>
Clean Periodic	"The drive is ready for a periodic cleaning."	"3" = "Degraded/Warning"	"The drive is due for routine cleaning:"	<p>"1. Wait for the current operation to finish."</p> <p>"2. Then use a cleaning cartridge.</p> <p>Check the drive user's manual for device specific cleaning instructions."</p>
Expired Cleaning Media	"The cleaning media has expired."	"6" = "Critical"	"The last cleaning cartridge used in the drive has worn out:"	<p>"1. Discard the worn out cleaning cartridge."</p> <p>"2. Wait for the current operation to finish."</p> <p>"3. Then use a new cleaning cartridge."</p>
Invalid Cleaning Media	"Invalid cleaning media type used."	"6" = "Critical"	"The last cleaning cartridge used in the drive was an invalid type:"	<p>"1. Do not use this cleaning cartridge in this drive."</p> <p>"2. Wait for the current operation to finish."</p> <p>"3. Then use a valid cleaning cartridge."</p>

Table 35: LibraryAlert AlertIndication Properties (Continued)

Event/Alert Summary	AlertIndication “Mapped” Properties from SSC-2 and SMC-2 Specs			
	OtherAlert Type	Perceived Severity	ProbableCause Description	Recommended Action[]
	string	Uint16	string	string
Retention Requested	“The drive is having severe trouble reading or writing, which will be resolved by a retention cycle.”	“3” = “Information”	“The drive has requested a retention operation.”	
Dual-Port Interface Error	“Failure of one interface port in a dual-port configuration (i.e., Fibre Channel)”	“3” = “Degraded/Warning”	“A redundant interface port on the drive has failed.”	
Cooling Fan Failure	“Fan failure inside drive mechanism or drive enclosure.”	“3” = “Degraded/Warning”	“A drive cooling fan has failed.”	“Replace cooling fan or drive enclosure”
Power Supply Failure	“Redundant power supply unit failure inside the drive enclosure or rack subsystem.”	“3” = “Degraded/Warning”	“A redundant power supply has failed inside the drive enclosure.”	“Check the enclosure user’s manual for instructions on replacing the failed power supply.”
Power Consumption	“Power consumption of the drive is outside specified range.”	“3” = “Degraded/Warning”	“The drive power consumption is outside the specified range.”	
Drive Maintenance	“The drive requires preventive maintenance (not cleaning).”	“3” = “Degraded/Warning”	“Preventive maintenance of the drive is required.”	“Check the drive users manual for device specific preventive maintenance tasks or call the drive supplier help line.”
Hardware A	“The drive has a hardware fault that requires reset to recover.”	“6” = “Critical”	“The drive has a hardware fault”	“1. Eject the media or magazine.” “2. Reset the drive.” “3. Restart the operation.”

Table 35: LibraryAlert AlertIndication Properties (Continued)

Event/Alert Summary	AlertIndication "Mapped" Properties from SSC-2 and SMC-2 Specs			
	OtherAlert Type	Perceived Severity	ProbableCause Description	Recommended Action[]
	string	Uint16	string	string
Hardware B	"The drive has a hardware fault that is not read/write related or requires a power cycle to recover."	"6" = "Critical"	"The drive has a hardware fault"	"1. Turn the drive off and then on again." "2. Restart the operation." "3. If the problem persists, call the drive supplier help line."
Interface	"The drive has identified an interface fault."	"3" = "Degraded/Warning"	"Bad cable or drive interface."	"1. Check the cables and cable connections." "2. Restart the operation."
Eject Media	"Error recovery action: Media Ejected"	"6" = "Critical"		"1. Eject the media or magazine." "2. Insert the media or magazine again." "3. Restart the operation."
Download Failure	"Firmware download failed."	"3" = "Degraded/Warning"	"The firmware download has failed because you have tried to use the incorrect firmware for this drive."	"Obtain the correct firmware and try again."
Drive Humidity	"Drive humidity limits exceeded."	"3" = "Degraded/Warning"	"Bad drive fan"	"Replace fan or drive enclosure"
Drive Temperature	"Drive temperature limits exceeded."	"3" = "Degraded/Warning"	"Bad cooling fan"	"Replace fan or drive enclosure"
Drive Voltage	"Drive voltage limits exceeded."	"3" = "Degraded/Warning"	"Bad drive power supply"	"Check the drive users manual for device specific preventive maintenance tasks or call the drive supplier help line."
Predictive Failure	"Predictive failure of drive hardware."	"6" = "Critical"		"A hardware failure of the drive is predicted. Call the drive supplier help line."

Table 35: LibraryAlert AlertIndication Properties (Continued)

Event/Alert Summary	AlertIndication "Mapped" Properties from SSC-2 and SMC-2 Specs			
	OtherAlert Type	Perceived Severity	ProbableCause Description	Recommended Action[]
	string	Uint16	string	string
Diagnostics Required	"The drive may have a hardware fault that may be identified by extended diagnostics (i.e., SEND DIAGNOSTIC command)."	"3" = "Degrading/Warning"	"The drive may have a hardware fault."	"1. Run extended diagnostics to verify and diagnose the problem. Check the drive user's manual for device specific instructions on running extended diagnostic tests."
Loader Hardware A	"Loader mechanism is having trouble communicating with the drive."	"6" = "Critical"	"The changer mechanism is having difficulty communicating with the drive."	"1. Turn the autoloader off then on." "2. Restart the operation." "3. If a problem persists, call the drive supplier help line."
Loader Stray Media	"Stray media left in loader after previous error recovery."	"6" = "Critical"	"A media has been left in the autoloader by a previous hardware fault."	"1. Insert an empty magazine to clear the fault." "2. If the fault does not clear, turn the autoloader off and then on again." "3. If the problem persists, call the drive supplier help line."
Loader Hardware B	"Loader mechanism has a hardware fault."	"3" = "Degrading/Warning"	"There is a problem with the autoloader mechanism."	

Table 35: LibraryAlert AlertIndication Properties (Continued)

Event/Alert Summary	AlertIndication “Mapped” Properties from SSC-2 and SMC-2 Specs			
	OtherAlert Type	Perceived Severity	ProbableCause Description	Recommended Action[]
	string	Uint16	string	string
Loader Door	“Changer door open.”	“6” = “Critical”	“The operation has failed because the autoloader door is open.”	<p>“1. Clear any obstructions from the autoloader door.”</p> <p>“2. Eject the magazine and then insert it again.”</p> <p>“3. If the fault does not clear, turn the autoloader off and then on again.”</p> <p>“4. If the problem persists, call the drive supplier help line.”</p>
Loader Hardware C	“The loader mechanism has a hardware fault that is not mechanically related.”	“6” = “Critical”	“The autoloader has a hardware fault:”	<p>“1. Turn the autoloader off and then on again.”</p> <p>“2. Restart the operation.”</p> <p>“3. If the problem persists, call the drive supplier help line. Check the autoloader user’s manual for device specific instructions on turning the device power on and off.”</p>
Loader Magazine	“Loader magazine not present.”	“6” = “Critical”	“The autoloader cannot operate without the magazine:”	<p>“1. Insert the magazine into the autoloader.”</p> <p>“2. Restart the operation.”</p>
Loader Predictive Failure	“Predictive failure of loader mechanism hardware”	“3” = “Degrading/Warning”		“A hardware failure of the changer mechanism is predicted. Call the drive supplier help line.”
Load Statistics	“Drive or library powered down with media loaded.”	“3” = “Degrading/Warning”	“Media statistics have been lost at some time in the past.”	

Table 35: LibraryAlert AlertIndication Properties (Continued)

Event/Alert Summary	AlertIndication “Mapped” Properties from SSC-2 and SMC-2 Specs			
	OtherAlert Type	Perceived Severity	ProbableCause Description	Recommended Action[]
	string	Uint16	string	string
Media Directory Invalid at Unload	“Error preventing the media directory being updated on unload.”	“3” = “Degrading/Warning”	“The directory on the media just unloaded has been corrupted.”	“The directory can be rebuilt by reading all the data.”
Media System area Write Failure	“Write errors while writing the system area on unload.”	“6” = “Critical”	“The media just unloaded could not write its system area successfully: “	“1. Copy data to another cartridge.” “2. Discard the old cartridge.”
Media System Area Read Failure	“Read errors while reading the system area on load.”	“6” = “Critical”	“The media system area could not be read successfully at load time: “	“1. Copy data to another cartridge.”
No Start of Data	“Media damaged, bulk erased, or incorrect format.”	“6” = “Critical”	“The start of data could not be found on the media.”	“1. Check that you are using the correct format media.” “2. Discard the media or return the media to your supplier.”
Loading Failure	“The drive is unable to load the media”	“6” = “Critical”	“The operation has failed because the media cannot be loaded and threaded.”	“1. Remove the cartridge, inspect it as specified in the product manual, and retry the operation.” “2. If the problem persists, call the drive supplier help line.”
Library Hardware A	“Changer mechanism is having trouble communicating with the internal drive”	“6” = “Critical”	“The library mechanism is having difficulty communicating with the drive: “	“1. Turn the library off then on.” “2. Restart the operation.” “3. If the problem persists, call the library supplier help line.”
Library Hardware B	“Changer mechanism has a hardware fault”	“3” = “Degrading/Warning”		“There is a problem with the library mechanism. If problem persists, call the library supplier help line.”

Table 35: LibraryAlert AlertIndication Properties (Continued)

Event/Alert Summary	AlertIndication “Mapped” Properties from SSC-2 and SMC-2 Specs			
	OtherAlert Type	Perceived Severity	ProbableCause Description	Recommended Action[]
	string	Uint16	string	string
Library Hardware C	“The changer mechanism has a hardware fault that requires a reset to recover.”	“6” = “Critical”	“The library has a hardware fault”	<p>“1. Reset the library.”</p> <p>“2. Restart the operation. Check the library user’s manual for device specific instructions on resetting the device.”</p>
Library Hardware D	“The changer mechanism has a hardware fault that is not mechanically related or requires a power cycle to recover.”	“6” = “Critical”	“The library has a hardware fault.”	<p>“1. Turn the library off then on again.”</p> <p>“2. Restart the operation.”</p> <p>“3. If the problem persists, call the library supplier help line. Check the library user’s manual for device specific instructions on turning the device power on and off.”</p>
Library Diagnostic Required	“The changer mechanism may have a hardware fault which would be identified by extended diagnostics.”	“3” = “Degrading/Warning”	“The library mechanism may have a hardware fault.”	“Run extended diagnostics to verify and diagnose the problem. Check the library user’s manual for device specific instructions on running extended diagnostic tests.”
Library Interface	“The library has identified an interface fault”	“6” = “Critical”	“Bad cable”	<p>“1. Check the cables and connections.”</p> <p>“2. Restart the operation.”</p>
Failure Prediction	“Predictive failure of library hardware”	“3” = “Degrading/Warning”		“A hardware failure of the library is predicted. Call the library supplier help line.”

Table 35: LibraryAlert AlertIndication Properties (Continued)

Event/Alert Summary	AlertIndication "Mapped" Properties from SSC-2 and SMC-2 Specs			
	OtherAlert Type	Perceived Severity	ProbableCause Description	Recommended Action[]
	string	Uint16	string	string
Library Maintenance	"Library preventative maintenance required."	"3" = "Degrading/Warning"		"Preventive maintenance of the library is required. Check the library user's manual for device specific preventative maintenance tasks, or call your library supplier help line."
Library Humidity Limits	"Library humidity limits exceeded"	"6" = "Critical"	"Library humidity range is outside the operational conditions"	
Library Temperature Limits	"Library temperature limits exceeded"	"6" = "Critical"	"Library temperature is outside the operational conditions"	
Library Voltage Limits	"Library voltage limits exceeded"	"6" = "Critical"	"Potential problem with a power supply."	
Library Stray Media	"Stray cartridge left in library after previous error recovery"	"6" = "Critical"	"Cartridge left in picker or drive"	<p>"1. Insert an empty magazine to clear the fault."</p> <p>"2. If the fault does not clear, turn the library off and then on again."</p> <p>"3. If the problem persists, call the library supplier help line."</p>
Library Pick Retry	"Operation to pick a cartridge from a slot had to perform an excessive number of retries before succeeding"	"3" = "Degrading/Warning"	"There is a potential problem with the drive ejecting cartridges or with the library mechanism picking a cartridge from a slot."	<p>"1.Run diagnostics to determine the health of the Library."</p> <p>"2. If the problem persists, call the library supplier help line."</p>

Table 35: LibraryAlert AlertIndication Properties (Continued)

Event/Alert Summary	AlertIndication "Mapped" Properties from SSC-2 and SMC-2 Specs			
	OtherAlert Type	Perceived Severity	ProbableCause Description	Recommended Action[]
	string	Uint16	string	string
Library Place Retry	"Operation to place a cartridge in a slot had to perform an excessive number of retries before succeeding"	"3" = "Degrading/Warning"	"Worn cartridge or bad storage slot/magazine"	"1. No action needs to be taken at this time." "2. If the problem persists, call the library supplier help line."
Library Load Retry	"Operation to load a cartridge in a drive had to perform an excessive number of retries before succeeding"	"3" = "Degrading/Warning"	"Worn cartridge or picker"	"Run diagnostics to determine the health of the library."
Library Door	"Library door open is preventing the library from functioning"	"6" = "Critical"	"The library has failed because the door is open."	"1. Clear any obstructions from the library door." "2. Close the library door." "3. If the problem persists, call the library supplier help line."
Library Mailslot	"Mechanical problem with import/export mailslot"	"6" = "Critical"	"There is a mechanical problem with the library media mailslot."	"1. Check for wedged storage media in import/export mailslot"
Library Magazine	"Library magazine not present"	"6" = "Critical"	"Administrator has removed the library's magazine"	"1. Insert the magazine into the library." "2. Restart the operation."
Library Security	"Library door opened then closed during operation"	"3" = "Degrading/Warning"	"Administrator is trying to remove or insert a storage media"	

Table 35: LibraryAlert AlertIndication Properties (Continued)

Event/Alert Summary	AlertIndication “Mapped” Properties from SSC-2 and SMC-2 Specs			
	OtherAlert Type	Perceived Severity	ProbableCause Description	Recommended Action[]
	string	Uint16	string	string
Library Security Mode	“Library security mode changed”	“2” = “Information”	“Administrator changed security mode”	“The library security mode has been changed. The library has either been put into secure mode, or the library has exited the secure mode. This is for information purposes only. No action is required.”
Library Offline	“Library manually turned offline”	“2” = “Information”	“The library has been manually turned offline and is unavailable for use.”	
Library Drive Offline	“Library turned internal drive offline.”	“2” = “Information”	“Drive failure”	“A drive inside the library has been taken offline. This is for information purposes only. No action is required.”
Library Scan Retry	“Operation to scan the bar code on a cartridge had to perform an excessive number of retries before succeeding”	“3” = “Degrading/Warning”	“There is a potential problem with the bar code label or the scanner hardware in the library mechanism.”	“1. No action needs to be taken at this time.” “2. If the problem persists, call the library supplier help line.”
Library Inventory	“Inconsistent media inventory”	“6” = “Critical”	“Media label has changed or bad Bar code scanner subsystem problem.”	“1. Redo the library inventory to correct inconsistency.” “2. Restart the operation. Check the applications user’s manual or the hardware user’s manual for specific instructions on redoing the library inventory.”
Library Illegal Operation	“Illegal operation detected”	“3” = “Degrading/Warning”	“A library operation has been attempted that is invalid at this time.”	

Table 35: LibraryAlert AlertIndication Properties (Continued)

Event/Alert Summary	AlertIndication “Mapped” Properties from SSC-2 and SMC-2 Specs			
	OtherAlert Type	Perceived Severity	ProbableCause Description	Recommended Action[]
	string	Uint16	string	string
Dual-Port Interface Error	“Failure of one interface port in a dual-port configuration”	“3” = “Degrading/Warning”	“A redundant interface port on the library has failed.”	
Cooling Fan Failure	“One or more fans inside the library have failed. Internal flag state only cleared when all flags are working again”	“3” = “Degrading/Warning”	“Bad cooling Fan”	
Power Supply	“Redundant power supply failure inside the library subsystem”	“3” = “Degrading/Warning”	“Bad Power Supply”	“A redundant power supply has failed inside the library. Check the library user’s manual for instructions on replacing the failed power supply. “
Power Consumption	“Power consumption of one or more devices inside the library is outside the specified range”	“3” = “Degrading/Warning”	“The library power consumption is outside the specified range.”	
Pass Through Mechanism Failure	“Error occurred in pass-through mechanism during self test or while attempting to transfer a cartridge between library modules”	“6” = “Critical”	“A failure has occurred in the cartridge pass-through mechanism between two library modules.”	
Cartridge in Pass-through Mechanism	“Cartridge left in the pass-through mechanism between two library modules”	“6” = “Critical”		“A cartridge has been left in the pass-through mechanism from a previous hardware fault. Check the library users guide for instructions on clearing this fault.”
Unreadable barcode Labels	“Unable to read a bar code label on a cartridge during library inventory/scan”	“2” = “Information”	“Bad Bar Code Labels or Scanner”	“The library was unable to read the bar code on a cartridge.”

8.7 Registered Name and Version

SML_Events version 1.1.0

8.8 CIM Elements

Table 36: CIM Elements for SML_Events

Element Name	Requirement	Description
CIM_AlertIndication (8.8.1)	Mandatory	

8.8.1 CIM_AlertIndication

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 37 describes class CIM_AlertIndication.

Table 37: SMI Referenced Properties/Methods for CIM_AlertIndication

Properties	Flags	Requirement	Description & Notes
Description		Mandatory	"LibraryAlertIndication"
AlertType		Mandatory	5 = "Device Alert"
ProbableCause		Mandatory	1 = "other"
Trending		Mandatory	1 = "Not Applicable"
SystemCreationClassName		Mandatory	CIM_ComputerSystem
OtherSeverity		Mandatory	Specified by vendor
EventID		Mandatory	Specified by vendor
ProviderName		Mandatory	Specified by vendor
SystemName		Mandatory	
AlertingManagedElement		Mandatory	
OtherAlertType		Mandatory	
PerceivedSeverity		Mandatory	
ProbableCauseDescription		Mandatory	

STABLE**Clause 9: Limited Access Port Elements Subprofile****9.1 Description**

Most libraries contain Limited Access Ports elements (a.k.a., mailslots, cartridge access ports, or import/export elements). This subprofile defines the classes necessary to publish information about these common components.

9.1.1 Instance Diagram

Figure 13 shows the relationship between LimitedAccessPorts and other portions of the Storage Library profile.

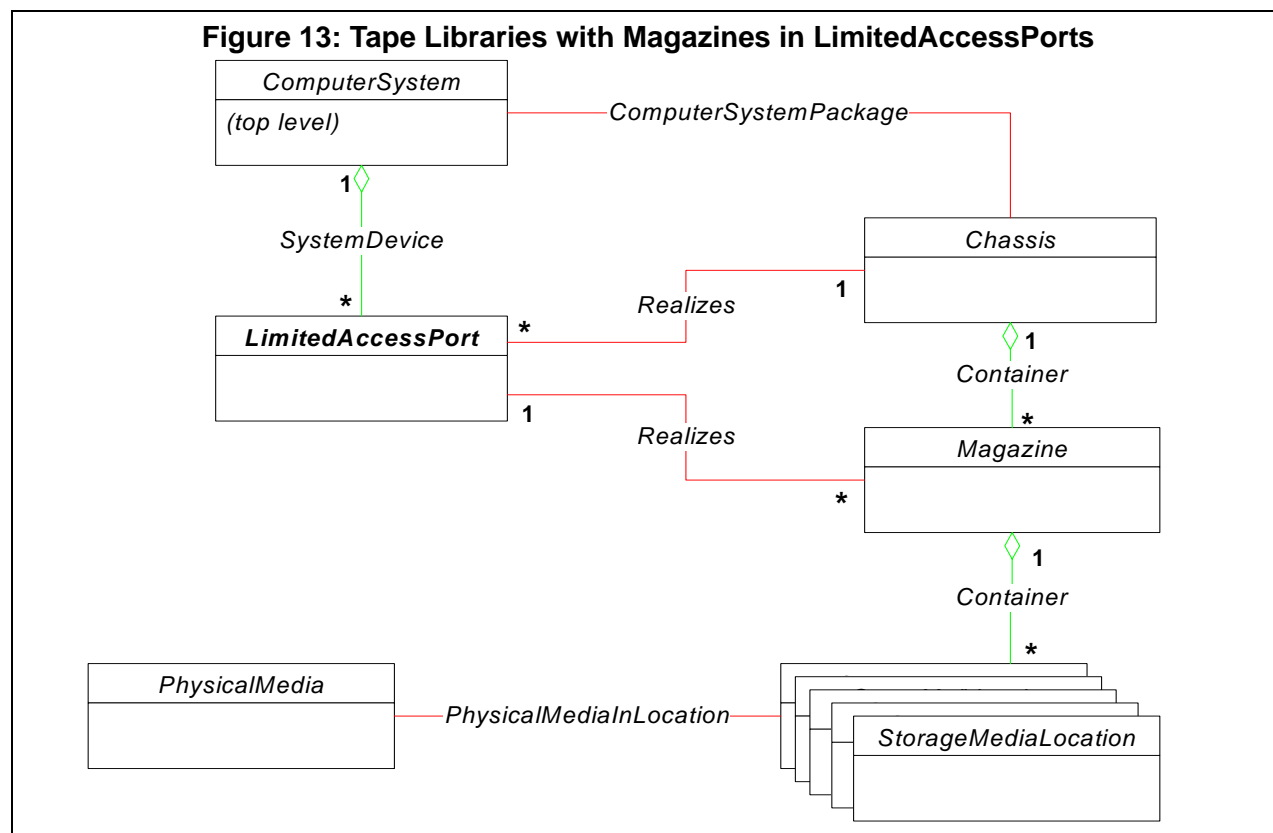
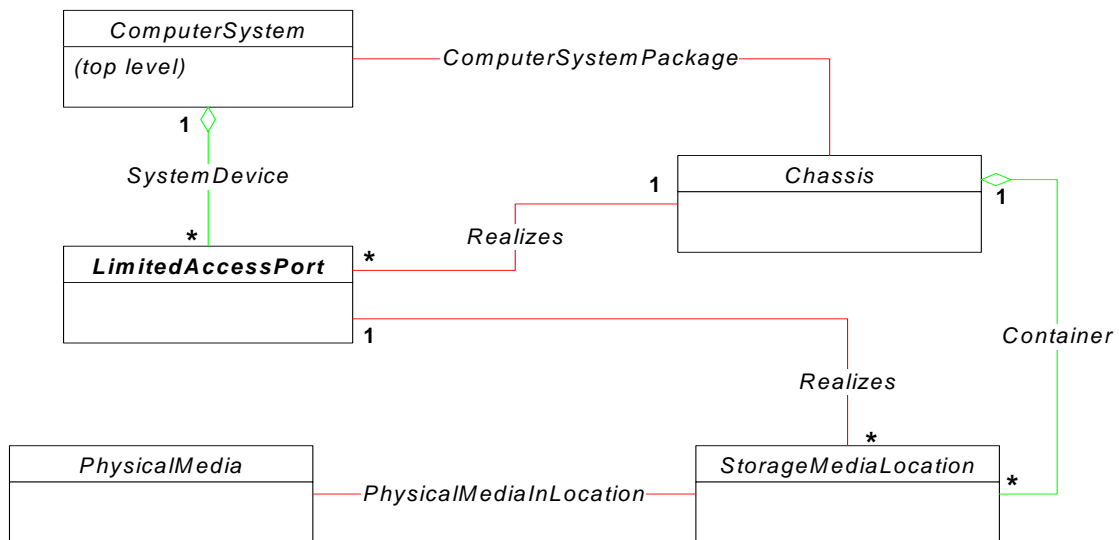


Figure 14: Tape Libraries with no Magazines in LimitedAccessPorts



9.2 Health and Fault Management Considerations

Not defined in this standard.

9.3 Cascading Considerations

Not defined in this standard.

9.4 Supported Subprofiles and Packages

None.

9.5 Methods of the Profile

None.

9.5.1 Client Considerations and Recipes

None

9.6 Registered Name and Version

Storage Library Limited Access Port Elements version 1.2.0

9.7 CIM Elements

Table 38: CIM Elements for Storage Library Limited Access Port Elements

Element Name	Requirement	Description
CIM_LimitedAccessPort (9.7.1)	Mandatory	LimitedAccessPorts represent hardware that transports physical media into or out of a Storage Library. They are identified as 'limited' since these ports do not provide access to ALL the PhysicalMedia or StorageMediaLocations in a Library, but only to a subset.
CIM_Magazine (9.7.2)	Mandatory	
CIM_Realizes (9.7.3)	Mandatory	The relationship between a LimitedAccessPort and the StorageMediaLocations, Magazines or Chassis to which it has access.
CIM_SystemDevice (9.7.4)	Mandatory	The relationship between a LimitedAccessPort and its hosting top-level ComputerSystem which represents the Storage Library.
CIM_Container (9.7.5)	Mandatory	The containment relationship of Magazines within a Chassis or StorageMediaLocations within a Magazine.
SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_LimitedAccessPort	Mandatory	Creation of an instance of LimitedAccessPort
SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_LimitedAccessPort	Mandatory	Deletion of an instance of LimitedAccessPort
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_LimitedAccessPort AND SourceInstance.OperationalStatus <> PreviousInstance.OperationalStatus	Mandatory	Deprecated WQL - Change in OperationalStatus of a LimitedAccessPort
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_LimitedAccessPort AND SourceInstance.CIM_LimitedAccessPort::OperationalStatus <> PreviousInstance.CIM_LimitedAccessPort::OperationalStatus	Optional	Experimental CQL - Change in OperationalStatus of a LimitedAccessPort

9.7.1 CIM_LimitedAccessPort

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 39 describes class CIM_LimitedAccessPort.

Table 39: SMI Referenced Properties/Methods for CIM_LimitedAccessPort

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	
CreationClassName		Mandatory	
SystemName		Mandatory	
DeviceID		Mandatory	
Extended		Mandatory	When true, the port's StorageMediaLocations are accessible to a human operator. When false, the StorageMediaLocations are accessible to a PickerElement.
ElementName		Mandatory	User-friendly name
OperationalStatus		Mandatory	Status of the LimitedAccessPort.
StatusDescriptions		Optional	Additional information related to the values in OperationalStatus.

9.7.2 CIM_Magazine

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 40 describes class CIM_Magazine.

Table 40: SMI Referenced Properties/Methods for CIM_Magazine

Properties	Flags	Requirement	Description & Notes
CreationClassName		Mandatory	
Tag		Mandatory	
LocationType		Mandatory	"Magazine"
LocationCoordinates		Mandatory	
MediaTypesSupported		Mandatory	

Table 40: SMI Referenced Properties/Methods for CIM_Magazine

Properties	Flags	Requirement	Description & Notes
MediaCapacity		Mandatory	The maximum number of PhysicalMedia that this StorageMediaLocation can hold.
PhysicalLabels		Optional	
LabelStates		Optional	
LabelFormats		Optional	

9.7.3 CIM_Realizes

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 41 describes class CIM_Realizes.

Table 41: SMI Referenced Properties/Methods for CIM_Realizes

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	
Dependent		Mandatory	

9.7.4 CIM_SystemDevice

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 42 describes class CIM_SystemDevice.

Table 42: SMI Referenced Properties/Methods for CIM_SystemDevice

Properties	Flags	Requirement	Description & Notes
GroupComponent		Mandatory	
GroupComponent		Mandatory	

9.7.5 CIM_Container

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 43 describes class CIM_Container.

Table 43: SMI Referenced Properties/Methods for CIM_Container

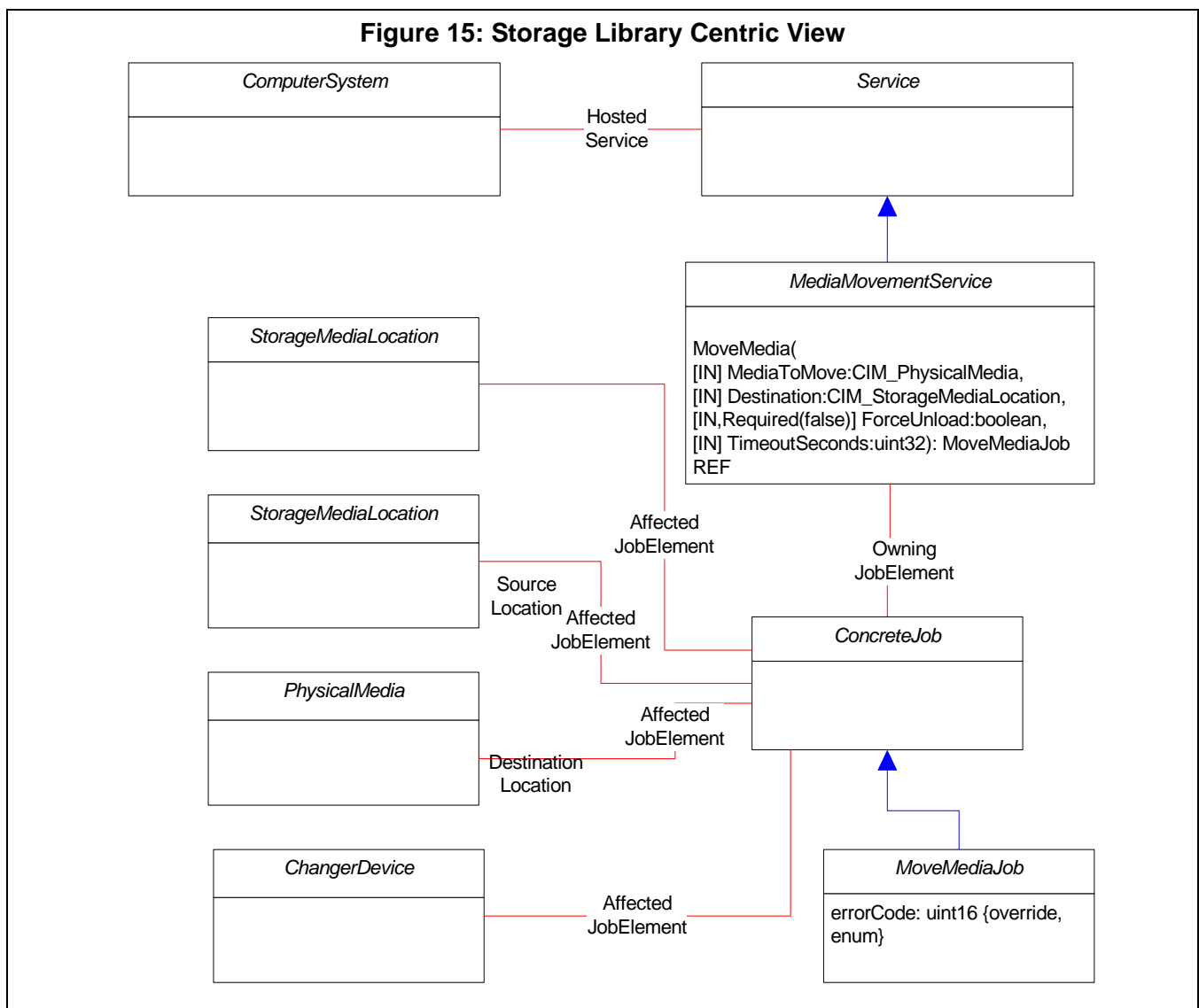
Properties	Flags	Requirement	Description & Notes
GroupComponent		Mandatory	
PartComponent		Mandatory	

STABLE

EXPERIMENTAL
Clause 10: Media Movement Subprofile
10.1 Description

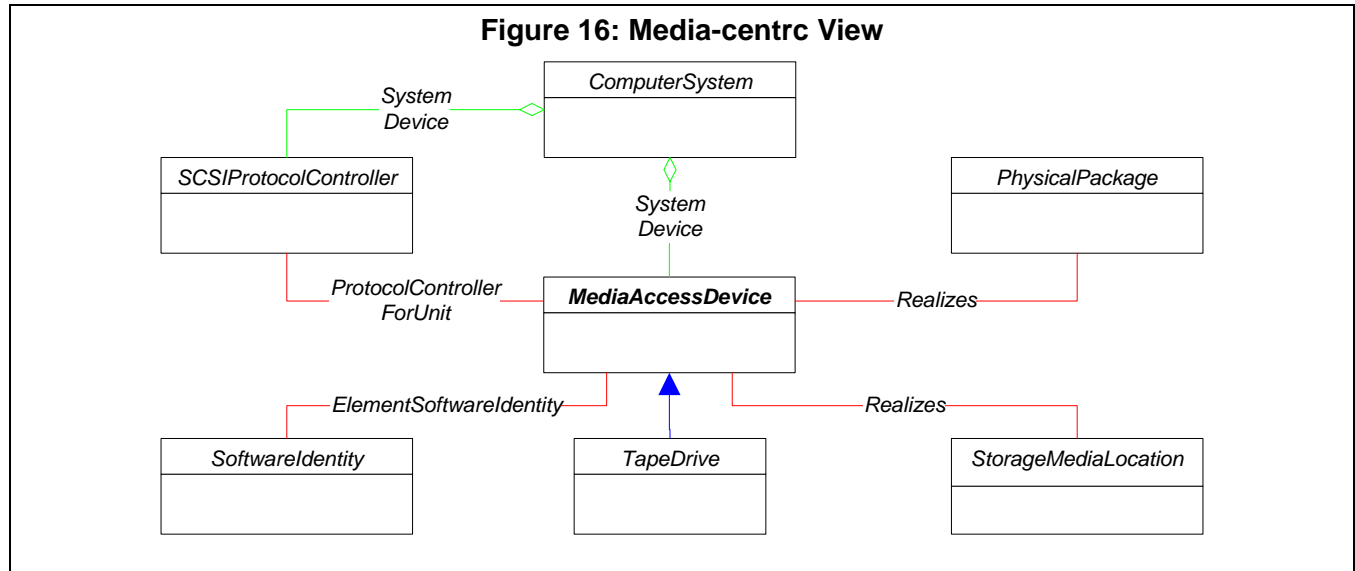
The Media Movement Subprofile defines a method to physically move a *PhysicalMedia* element from its current *StorageMediaLocation* to another *StorageMediaLocation* within the library with which the media is compatible. Such a method is convenient for purposes including library maintenance, self test, and demonstration. The method is implemented by a *HostedService* associated with the *ComputerSystem* which models the storage library. The method supports asynchronous operation according to the Job Control Subprofile.

Figure 15 illustrates the subprofile from the library perspective.



When the move media operation is performed, the storage library shall physically move the medium, and then update the storage library's CIM object model. In particular, the *StorageMediaInLocation* association between the *PhysicalMedia* instance and the source *StorageMediaLocation* instance shall be removed and a new association

made between the PhysicalMedia instance and the destination StorageMediaLocation. This is illustrated in Figure 16.



10.2 Health and Fault Management Considerations

10.2.1 NULL Instance Handling

If a non-null instance of ConcreteJob is returned by the MoveMedia method, the implementation shall report errors which occur during the execution of the job through the ConcreteJob.GetError() method. See *Storage Management Technical Specification, Part 2 Common Profiles* Clause 10: Media Movement Subprofile for details.

10.2.2 8.1 Media Movement Subprofile Standard Messages

The standard messages specific to this profile are listed Table 44.

Table 44: Media Movement Standard Messages

Message ID	Message Name
1	Source Media not Found
2	Destination Location Full
3	Invalid Source Media
4	Invalid Destination Location
5	Media not Compatible with Destination
6	Reservation Conflict
7	Busy
8	Hardware Error
9	Internal Model Error
10	Command Sequence Error

10.3 Cascading Considerations

Not defined in this standard.

10.4 Supported Subprofiles and Packages

None.

10.5 Methods of the Profile

10.5.1 Moving a piece of PhysicalMedia

```
uint32 MoveMedia(
    [OUT, Description("Reference to the job (may be null if job completed.)")]
    CIM_ConcreteJob REF MoveMediaJob,
    [IN, Description( "The piece of media to be moved" ) ]
    CIM_PhysicalMedia REF MediaToMove,
    [IN, Description( "The destination location" ) ]
    CIM_StorageMediaLocation REF Destination,
    [IN, Required(false),
     Description( "Optional parameter instructing the storage library to "
                 "first unload the media if it is loaded in a MediaAccessDevice." ) ]
    boolean ForceUnload,
    [IN, Required(false),
     Description( "The timeout time in seconds" ) ]
    unit32 Timeout )
```

Error returns are:

```
{ "Job Completed with No Error", "Not Supported", "Unknown", "Timeout",
  "Failed", "Invalid Parameter", "In Use", "DMTF Reserved",
  "Method Parameters Checked - Job Started", "Busy", "Method Reserved",
  "Vendor Specific" }
```

The MoveMedia method takes as input references to the media to be moved, the destination location, and a timeout value. The method attempts to initiate a process on the Storage Library which will perform the media movement. If the process is successfully initiated, the MoveMedia returns a ConcreteJob object and an integer return code indicating the status of the job creation. If a non-null instance of ConcreteJob is returned, the instance shall be associated with an instance of MethodResult as specified by the Job Control Subprofile. See *Storage Management Technical Specification, Part 2 Common Profiles* Clause 30: Job Control Subprofile for details of job creation and execution.

10.5.1.1 Timeout parameter

The optional Timeout parameter allows the MediaMovementService process or a sub-process to handle job timeout rather than delegating the responsibility to the SMI client. If the Timeout parameter is omitted (set to "null"), the method shall use the library's default behavior, which may be vendor or library specific.

10.5.1.2 ForceUnload parameter

When set to "true", the optional ForceUnload parameter instructs the Storage Library to first unload the PhysicalMedia if it is loaded in a MediaAccessDevice. If the ForceUnload parameter is set to "false" and the PhysicalMedia is loaded in a MediaAccessDevice, the job shall fail and the ConcreteJob's GetError() method shall return an instance of

Error indicating “Media Loaded in Access Device”, an error message specific to the Media Movement Subprofile. If the ForceUnload parameter is omitted (set to “null”), the method shall use the library’s default behavior, which may be vendor or library specific.

10.6 Client Considerations and Recipes

10.6.1 Concurrent library access by SMI clients and other applications.

The MoveMedia method introduces an alternate path to modify the configuration of the storage library, possibly interfering with the operation of other applications using the library concurrently. The MoveMedia method shall be used with caution in situations where applications other than the SMI client are moving media in the storage library.

10.6.2 Use of the ForceUnload parameter

Forcing a MediaAccessDevice to unload media while in use by other applications may cause data loss.

10.6.3 Job Lifecycle Indications

SMI Servers implementing the Job Control profile are required to support a set of indications which indicate transitions in the operational status of the job. In particular, an indication shall be provided when a job stops, either successfully or with an error condition. The server may also generate indications for change in job status or percent complete. See 30.8, “CIM Elements” in *Storage Management Technical Specification, Part 2 Common Profiles* for indication subscription details.

10.7 Registered Name and Version

Storage Library Media Movement version 1.1.0

10.8 CIM Elements

Table 45: CIM Elements for Storage Library Media Movement

Element Name	Requirement	Description
CIM_HostedService (10.8.1)	Mandatory	The relationship between the top-level ComputerSystem representing the Storage Library and the MediaMovementService
SNIA_MediaMovementService (10.8.2)	Mandatory	

10.8.1 CIM_HostedService

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 46 describes class CIM_HostedService.

Table 46: SMI Referenced Properties/Methods for CIM_HostedService

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	
Dependent		Mandatory	

10.8.2 SNIA_MediaMovementService

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 47 describes class SNIA_MediaMovementService.

Table 47: SMI Referenced Properties/Methods for SNIA_MediaMovementService

Properties	Flags	Requirement	Description & Notes
SystemCreationClass Name		Mandatory	
CreationClassName		Mandatory	
SystemName		Mandatory	
Name		Mandatory	
MoveMedia()		Mandatory	

EXPERIMENTAL
