



Storage Management Technical Specification, Part 8 Media Libraries

Version 1.7.0, Revision 5

Abstract: This SNIA Technical Position defines an interface between WBEM-capable clients and servers for the secure, extensible, and interoperable management of networked storage.

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

SNIA Technical Position

8 March, 2016

REVISION HISTORY

Revision 1

Date

8 Sept 2014

SCRs Incorporated and other changes

None

Comments

Editorial notes and DRAFT material are displayed.

Revision 2

Date

18 December 2014

SCRs Incorporated and other changes

None

Comments

Editorial notes and DRAFT material are displayed.

USAGE text was revised to address code.

(now included in the front matter for all SNIA specifications)

Revision 3

Date

20 May 2015

SCRs Incorporated and other changes

Removed Media Movement Subprofile, Partitioned Tape Library Profile, Virtual Tape Library Profile, Virtual Tape Library Copy Profile, Library Views Profile, and VTL Capacity, and Reporting Profile per TSG voice vote--profiles that were Experimental and using SNIA_ classes.

References

- Added DMTF DSP1054 v1.2.2, Indications Profile (and changed version to 1.2.2 throughout book)

All recipes and associated text were removed.

Comments

Editorial notes and DRAFT material were hidden.

Revision 4

Date

9 September 2015

SCRs Incorporated and other changes

Multiple profiles

- Instances of *subprofile* were changed to *profile*. (TSG meeting voice vote)
- Profile versions and related text were updated. (TSG meeting voice vote)
- *CIM/XML* was changed to *CIM-XML* (Response to ballot comments)

References

- Removed reference to DMTF DSP0200, CIM Operations over HTTP

Storage Library profile

- Simplified the use case for discovery of data path technology (CORE-SMIS-SCR-00084)

Annex A (informative) SMI-S Information Model

- DMTF's CIM schema version changed to 2.45.0. (TSG meeting voice vote)

Comments

- Editorial notes and DRAFT material were hidden.

Revision 5

Date

9 October 2015

SCRs Incorporated and other changes

Multiple profiles: Addressed SMI-S 1.7.0 Revision 4 TSG ballot comments that were strictly editorial and were approved by voice vote of the TSG.

Comments

- Editorial notes were hidden.

Suggestion for changes or modifications to this document should be sent to the SNIA Storage Management Initiative Technical Steering Group (SMI-TSG) at <http://www.snia.org/feedback/>

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, and,
- 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.

All code fragments, scripts, data tables, and sample code in this SNIA document are made available under the following license:

BSD 3-Clause Software License

Copyright (c) 2016, The Storage Networking Industry Association.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of The Storage Networking Industry Association (SNIA) nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

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 © 2003-2016 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 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.

INTENDED AUDIENCE

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

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 1.7.0. 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

Maturity Level

In addition to informative and normative content, this specification includes guidance about the maturity of emerging material that has completed a rigorous design review but has limited implementation in commercial products. This material is clearly delineated as described in the following sections. The typographical convention is intended to provide a sense of the maturity of the affected material, without altering its normative content. By recognizing the relative maturity of different sections of the standard, an implementer should be able to make more informed decisions about the adoption and deployment of different portions of the standard in a commercial product.

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. 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. 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. 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.

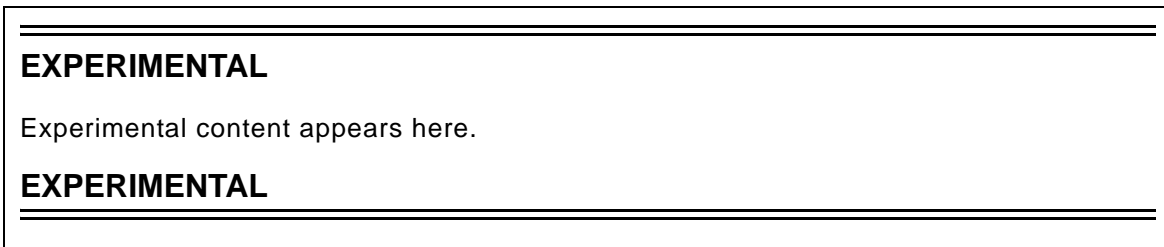


Figure 1 - Experimental Maturity Level Tag

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.

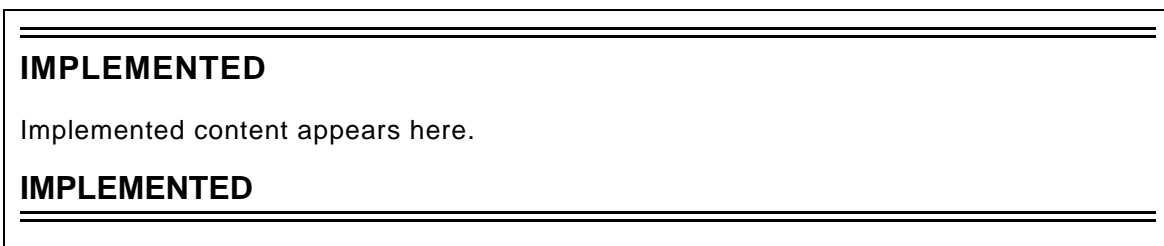


Figure 2 - Implemented Maturity Level Tag

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.



Figure 3 - Stable Maturity Level Tag

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.



Figure 4 - Deprecated Tag

Contents

Revision History	2
List of Figures	13
List of Tables	15
Foreword	17
1 Scope	19
2 Normative References	21
2.1 General	21
2.2 Approved references	21
2.3 References under development	21
3 Terms, Definitions, Symbols, Abbreviations, and Conventions	23
3.1 General	23
3.2 Terms and Definitions	23
4 Storage Library Profile	25
4.1 Synopsis	25
4.2 Description	25
4.3 Health and Fault Management Considerations	39
4.4 Methods of the Profile	39
4.5 Use Cases	39
4.6 CIM Elements	41
5 Element Counting Profile	57
5.1 Synopsis	57
5.2 Description	57
5.3 Health and Fault Management Considerations	57
5.4 Methods of the Profile	57
5.5 Use Cases	59
5.6 CIM Elements	61
6 Storage Library Capacity Profile	63
6.1 Synopsis	63
6.2 Description	63
6.3 Health and Fault Management Considerations	63
6.4 Methods of the Profile	63
6.5 Use Cases	64
6.6 CIM Elements	64
7 Limited Access Port Elements Profile	67
7.1 Synopsis	67
7.2 Description	67
7.3 Health and Fault Management Considerations	68
7.4 Methods of the Profile	68
7.5 Use Cases	68
7.6 CIM Elements	68
Annex A (informative) SMI-S Information Model	77

LIST OF FIGURES

Figure 1 - Experimental Maturity Level Tag	8
Figure 2 - Implemented Maturity Level Tag	8
Figure 3 - Stable Maturity Level Tag	9
Figure 4 - Deprecated Tag	9
Figure 5 - Storage Library-centric Instance Diagram	27
Figure 6 - MediaAccessDevice-centric Instance Diagram	28
Figure 7 - ChangerDevice-centric Instance Diagram	28
Figure 8 - Physical View Instance Diagram	29
Figure 9 - StorageMediaLocation Instance Diagram	29
Figure 10 - Instance Diagram	57
Figure 11 - Storage Library Capacity Instance Diagram	63
Figure 12 - Tape Libraries with Magazines in LimitedAccessPorts.....	67
Figure 13 - Tape Libraries with no Magazines in LimitedAccessPorts.....	68

LIST OF TABLES

Table 1 - Supported Profiles for Storage Library	25
Table 2 - CIM Elements for Storage Library	41
Table 3 - SMI Referenced Properties/Methods for CIM_ChangerDevice	48
Table 4 - SMI Referenced Properties/Methods for CIM_Chassis	49
Table 5 - SMI Referenced Properties/Methods for CIM_ComputerSystem	49
Table 6 - SMI Referenced Properties/Methods for CIM_ComputerSystemPackage	50
Table 7 - SMI Referenced Properties/Methods for CIM_ElementCapabilities	50
Table 8 - SMI Referenced Properties/Methods for CIM_ElementSoftwareIdentity	50
Table 9 - SMI Referenced Properties/Methods for CIM_MediaAccessDevice	51
Table 10 - SMI Referenced Properties/Methods for CIM_PackagedComponent	51
Table 11 - SMI Referenced Properties/Methods for CIM_PhysicalMedia	52
Table 12 - SMI Referenced Properties/Methods for CIM_PhysicalMediaInLocation	52
Table 13 - SMI Referenced Properties/Methods for CIM_ProtocolControllerForUnit	52
Table 14 - SMI Referenced Properties/Methods for CIM_Realizes	53
Table 15 - SMI Referenced Properties/Methods for CIM_SCSIProtocolController	53
Table 16 - SMI Referenced Properties/Methods for CIM_SoftwareIdentity	54
Table 17 - SMI Referenced Properties/Methods for CIM_StorageLibraryCapabilities	54
Table 18 - SMI Referenced Properties/Methods for CIM_StorageMediaLocation	55
Table 19 - SMI Referenced Properties/Methods for CIM_SystemDevice (System to Changer Device)	55
Table 20 - SMI Referenced Properties/Methods for CIM_SystemDevice (System to MediaAccessDevice)	55
Table 21 - SMI Referenced Properties/Methods for CIM_SystemDevice (System to SCSIProtocolController)	56
Table 22 - CIM Elements for Storage Library Element Counting	61
Table 23 - SMI Referenced Properties/Methods for CIM_ConfigurationReportingService	61
Table 24 - SMI Referenced Properties/Methods for CIM_HostedService	61
Table 25 - CIM Elements for Storage Library Capacity	64
Table 26 - SMI Referenced Properties/Methods for CIM_ConfigurationCapacity	64
Table 27 - SMI Referenced Properties/Methods for CIM_ElementCapacity	65
Table 28 - CIM Elements for Storage Library Limited Access Port Elements	68
Table 29 - SMI Referenced Properties/Methods for CIM_Container	69
Table 30 - SMI Referenced Properties/Methods for CIM_LimitedAccessPort	69
Table 31 - SMI Referenced Properties/Methods for CIM_Magazine	70
Table 32 - SMI Referenced Properties/Methods for CIM_Realizes	70
Table 33 - SMI Referenced Properties/Methods for CIM_SystemDevice	71

FOREWORD

Storage Library Profile and related component profiles defined in this part 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 part also standardizes library specific life-cycle and alert indications that are delivered to a client asynchronously, once a client subscribes to these indications.

This part covers Part 7 (Media Libraries) of the SMI-S standard listed below. While Part 1 describes SMI-S concepts and terms, some of the profiles referenced in the 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 Overview, 1.7.0 Rev 5*
- *Storage Management Technical Specification, Part 2 Common Architecture, 1.7.0 Rev 5*
- *Storage Management Technical Specification, Part 3 Common Profiles, 1.7.0 Rev 5*
- *Storage Management Technical Specification, Part 4 Block Devices, 1.7.0 Rev 5*
- *Storage Management Technical Specification, Part 5 Filesystems, 1.7.0 Rev 5*
- *Storage Management Technical Specification, Part 6 Fabric, 1.7.0 Rev 5*
- *Storage Management Technical Specification, Part 7 Host Elements, 1.7.0 Rev 5*
- *Storage Management Technical Specification, Part 8 Media Libraries, 1.7.0 Rev 5*

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, 4360 ArrowsWest Drive, Colorado Springs, Colorado 80907, U.S.A.

1 Scope

This version of the 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.

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

DMTF DSP1054 v1.2.2, Indications Profile

http://www.dmtf.org/sites/default/files/standards/documents/DSP1054_1.2.2.pdf

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

2.3 References under development

Storage Management Technical Specification, Part 2 Common Architecture, 1.7.0 Rev 5

Storage Management Technical Specification, Part 3 Common Profiles, 1.7.0 Rev 5

Storage Management Technical Specification, Part 4 Block Devices, 1.7.0 Rev 5

Normative References

3 Terms, Definitions, Symbols, Abbreviations, and Conventions

3.1 General

For the purposes of this document, the definitions, symbols, abbreviations, and conventions given in *Storage Management Technical Specification, Part 2 Common Architecture, 1.7.0 Rev 5* and the following apply.

3.2 Terms and 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

Note 1 to entry: In tape libraries, it is the tape drive.

3.2.3

storage media location

one of various locations within a media library where the physical media can be placed

Note 1 to entry: 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

Note 1 to entry: A library also contains a limited number of media access devices for reading and writing to the media.

Note 2 to entry: A changer device within the library moves the media between a stored location and drive or between two locations.

Note 3 to entry: 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.

Note 4 to entry: 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

Note 1 to entry: A limited access port is also known as an I/O port, import export port, mailslot, etc.

3.2.6

library capacity (storage library capacity)

the capacity of a storage media library, measured in terms of the number of physical media it can hold

3.2.7

magazine

a container that holds multiple physical media

Note 1 to entry: Some storage media libraries have magazines that fit into the physical slot instead of single media.

STABLE

4 Storage Library Profile

4.1 Synopsis

Profile Name: Storage Library (Autonomous Profile)

Version: 1.6.0

Organization: SNIA

Central Class: ChangerDevice

Scoping Class: ComputerSystem

Supported Profiles: Table 1 describes the supported profiles for Storage Library.

Table 1 - Supported Profiles for Storage Library

Profile Name	Organization	Version	Requirement	Description
Access Points	SNIA	1.3.0	Optional	
Location	SNIA	1.4.0	Optional	
Software	SNIA	1.4.0	Optional	
Storage Library Limited Access Port Elements	SNIA	1.2.0	Optional	
Storage Library Capacity	SNIA	1.1.0	Optional	Experimental.
Storage Library Element Counting	SNIA	1.1.0	Optional	Experimental.
Physical Package	SNIA	1.5.0	Mandatory	
Launch In Context	DMTF	1.0.0	Optional	Experimental. See DSP1102, version 1.0.0
Indications	DMTF	1.2.2	Mandatory	See DSP1054, version 1.2.2
FC Target Ports	SNIA	1.7.0	Support for at least one is mandatory.	
SAS Target Ports	SNIA	1.4.0		
iSCSI Target Ports	SNIA	1.6.0		

4.2 Description

The schema for a storage library provides the classes and associations necessary to represent various forms of removable media libraries. This profile 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.

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

Storage Library Profile

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 4.2.7 "Durable Names and Correlatable IDs of the Profile". All other objects do not require durable names and have instances within a single name space.

4.2.1 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.

4.2.2 System Level View

Figure 5: "Storage Library-centric Instance Diagram" 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.

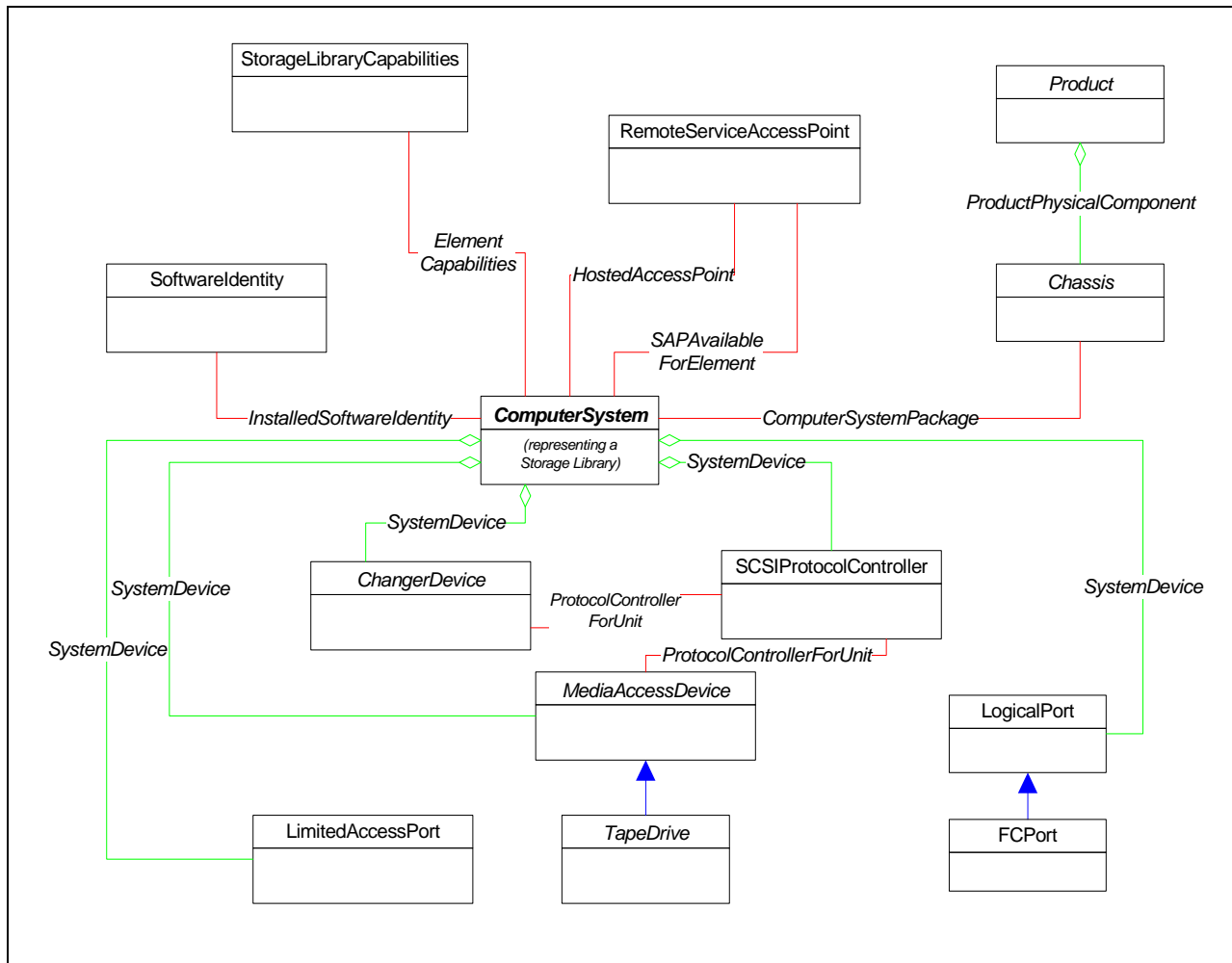


Figure 5 - Storage Library-centric Instance Diagram

EXPERIMENTAL

The capabilities of the Storage Library implementation are identified in an instance of CIM_StorageLibraryCapabilities, which is associated to the top level Storage Library ComputerSystem via ElementCapabilities. This includes information on the capacity optimization techniques, views, capabilities and timings supported by the Storage Library implementation.

EXPERIMENTAL

4.2.3 MediaAccessDevice-centric View

Figure 6: "MediaAccessDevice-centric Instance Diagram" 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.

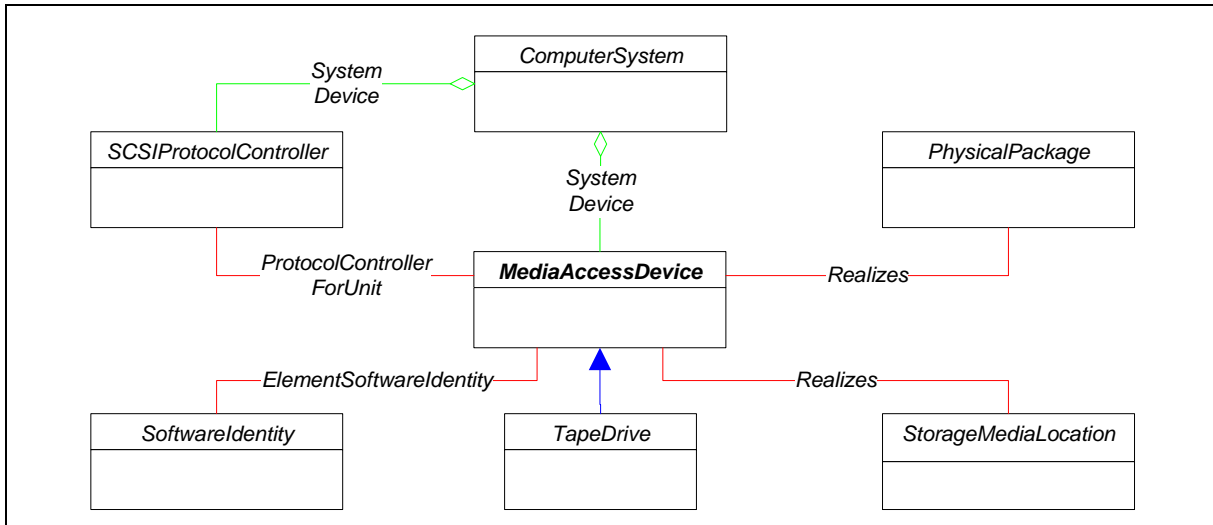


Figure 6 - MediaAccessDevice-centric Instance Diagram

4.2.4 ChangerDevice-centric View

Figure 7 shows the required classes related to ChangerDevice.

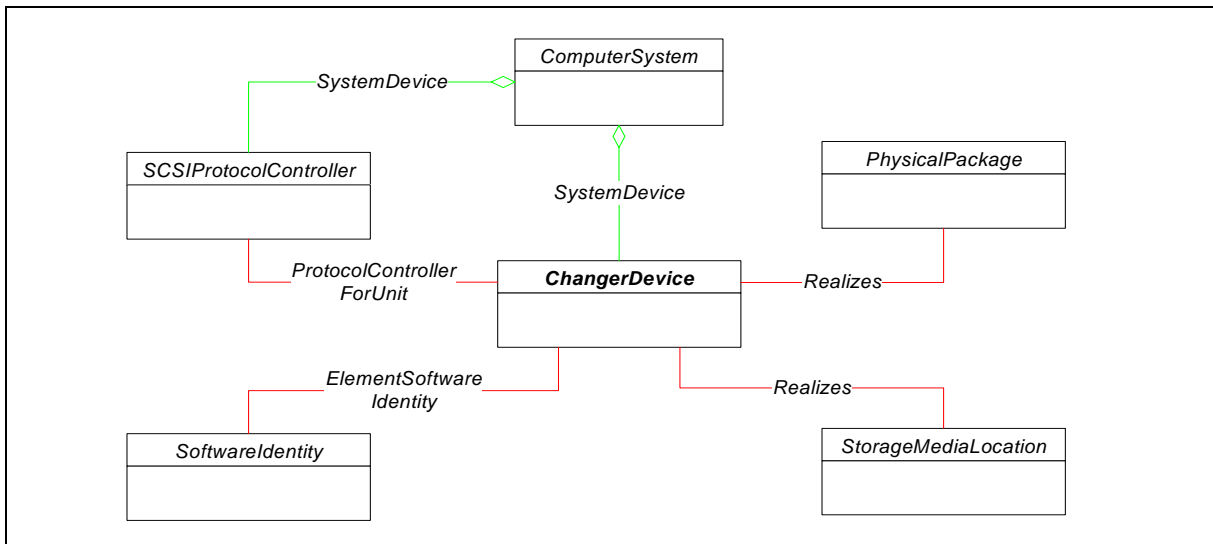


Figure 7 - ChangerDevice-centric Instance Diagram

4.2.5 Physical View

Figure 8: "Physical View Instance Diagram" 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:

- a) Instantiate multiple Magazines associated to Chassis via Container, then instantiate StorageMediaLocations that are contained (again via Container) within each Magazine;

- b) 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.

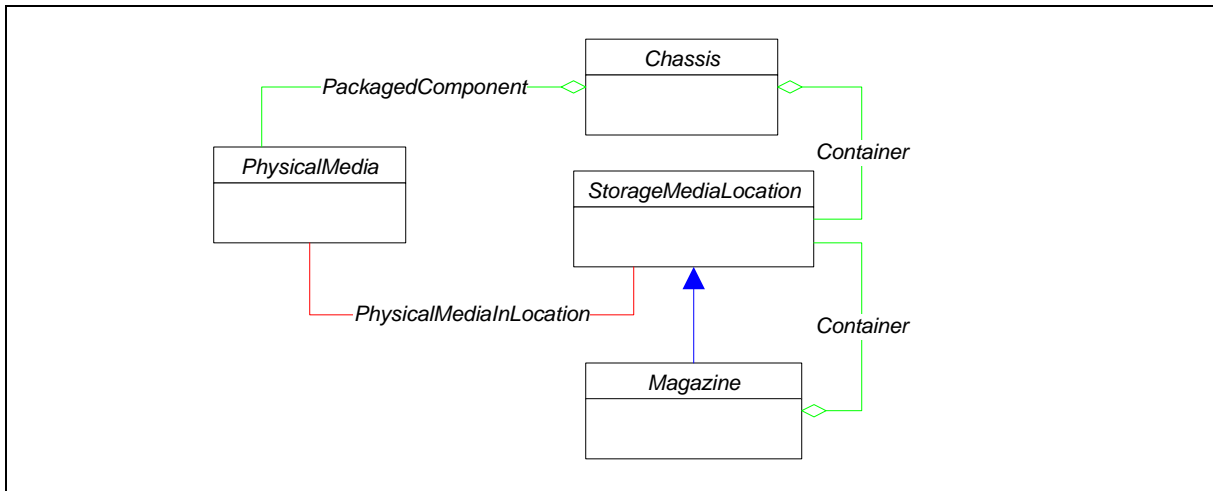


Figure 8 - Physical View Instance Diagram

4.2.6 StorageMediaLocation Instance Diagram

Figure 9: "StorageMediaLocation Instance Diagram" 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: "StorageMediaLocation Instance Diagram"). All tapes, irrespective of the location, are associated with the chassis using PackagedComponent.

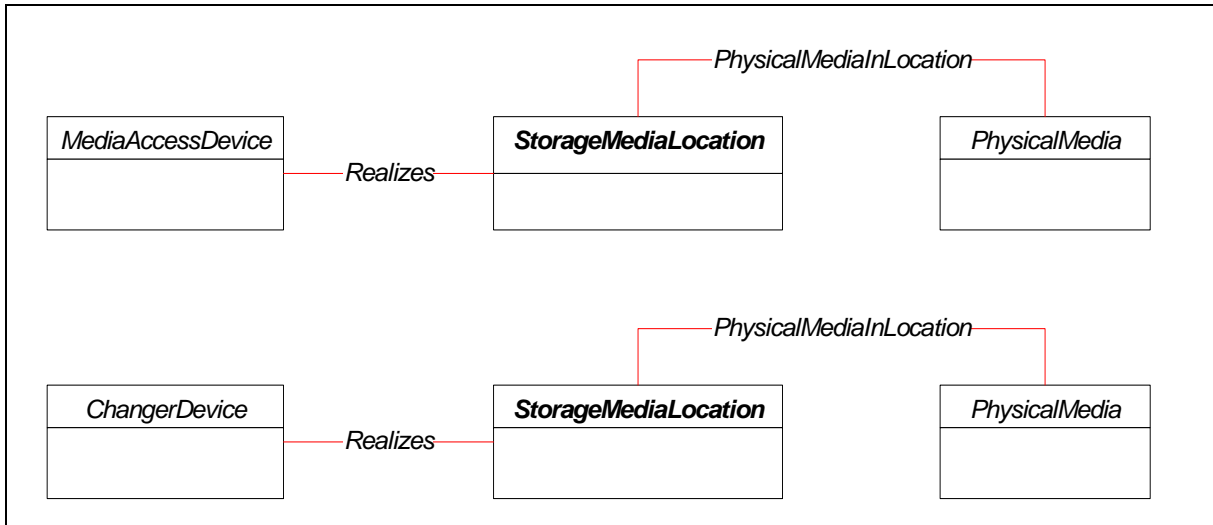


Figure 9 - StorageMediaLocation Instance Diagram

4.2.7 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 *Storage Management Technical Specification, Part 2 Common Architecture, 1.7.0 Rev 5, 7.5 "Guidelines for Storage System Names"*. 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 2 Common Architecture, 1.7.0 Rev 5 7.6 "Standard Formats for Correlatable Names"* for additional information.

4.2.8 Media Library Indications

4.2.8.1 Read Warning Alert

The drive is having severe trouble reading.

Probable Cause: The drive is having problems reading data. No data has been lost, but there has been a reduction in the performance.

4.2.8.2 Write Warning Alert

The drive is having severe trouble writing.

Probable Cause: Worn out Media

Recommended Actions: 1. Discard the worn out media.2. Use a new cleaning media.

4.2.8.3 Hard Error Alert

The drive had a hard read or write error.

Probable Cause: Bad Media or Drive. The operation has stopped because an error has occurred while reading or writing data that the drive cannot correct.

4.2.8.4 Media Alert

Media can no longer be written/read, or performance is severely degraded.

Probable Cause: Bad Media

Recommended Actions: 1. Copy any data you require from this media.2. Do not use this media again.3. Restart the operation with a different media.

4.2.8.5 Read Failure Alert

The drive can no longer read data from the storage media.

Probable Cause: Worn out media

Recommended Actions: 1. Replace media.2. Call the drive supplier help line.

4.2.8.6 Write Failure Alert

The drive can no longer write data to the media.

Probable Cause: The media is from a faulty batch or the drive is faulty:

Recommended Actions: 1. Use known-good media to test the drive. 2. If the problem persists, call the media drive supplier

4.2.8.7 Media Life Alert

The media has exceeded its specified life.

Probable Cause: The media has reached the end of its calculated useful life:

Recommended Actions: 1. Copy any data you need to another media.2. Discard the old media.

4.2.8.8 Not Data Grade Alert

The cartridge is not data-grade. Any data you write to the media is at risk. Replace the cartridge with a data- grade media.

Probable Cause: The cartridge is not data-grade. Any data you write to the media is at risk.

Recommended Actions: Replace the cartridge with a data-grade media.

4.2.8.9 Write Protect Alert

Write command is attempted to a write protected media.

Probable Cause: Replace with writable media

Recommended Actions: You are trying to write to a write protected cartridge. Remove the write protection or use another media.

4.2.8.10 No Removal Alert

Manual or software unload attempted when prevent media removal is on.

Probable Cause: Wait until drive is not in-use.

Recommended Actions: You cannot eject the cartridge because the drive is in use. Wait until the operation is complete before ejecting the cartridge.

4.2.8.11 Cleaning Media Alert

Cleaning media loaded into drive

Probable Cause: The media in the drive is a cleaning cartridge.

Recommended Actions: Replace this media with writeable media

4.2.8.12 Unsupported Format Alert

Attempted load of unsupported media format (e.g., DDS2 in DDS1 drive).

Probable Cause: You have tried to load a cartridge of a type that is not supported by this drive.

Recommended Actions: Insert media of a type supported by this drive

4.2.8.13 Recoverable Snapped Tape Alert

Tape snapped/cut in the drive where media can be de-mounted.

Probable Cause: The operation has failed because the tape in the drive has snapped:

Recommended Actions: 1. Discard the old tape.2. Restart the operation with a different tape.

4.2.8.14 Unrecoverable Snapped Tape Alert

Tape snapped/cut in the drive where media cannot be de-mounted.

Probable Cause: The operation has failed because the tape in the drive has snapped:

Recommended Actions: 1. Do not attempt to extract the tape cartridge.2. Call the tape drive supplier help line.

4.2.8.15 Memory Chip In Cartridge Failure Alert

Memory chip failed in cartridge.

Probable Cause: The memory in the media has failed, which reduces performance.

Recommended Actions: Do not use the cartridge for further write operations.

4.2.8.16 Forced Eject Alert

Manual or forced eject while drive actively writing or reading.

Probable Cause: The operation has failed because the media was manually de-mounted while the drive was actively writing or reading.

4.2.8.17 Read Only Format Alert

Media loaded that is read-only format.

Probable Cause: You have loaded a cartridge of a type that is read-only in this drive. The cartridge will appear as write protected.

4.2.8.18 Directory Corrupted On Load Alert

Drive powered down while loaded, or permanent error prevented the directory being updated.

Probable Cause: The directory on the cartridge has been corrupted. File search performance will be degraded.

Recommended Actions: The directory can be rebuilt by reading all the data on the cartridge.

4.2.8.19 Nearing Media Life Alert

Media may have exceeded its specified number of passes.

Probable Cause: The storage media is nearing the end of its calculated life.

Recommended Actions: 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.

4.2.8.20 Clean Now Alert

The drive thinks it has a head clog or needs cleaning.

Probable Cause: The drive needs cleaning:

Recommended Actions: 1. If the operation has stopped, eject the storage media and clean the drive.2. If the operation has not stopped, wait for it to finish and then clean the drive. Check the drive users manual for device specific cleaning.

4.2.8.21 Clean Periodic Alert

The drive is ready for a periodic cleaning.

Probable Cause: The drive is due for routine cleaning:

Recommended Actions: 1. Wait for the current operation to finish. 2. Then use a cleaning cartridge. Check the drive users manual for device specific cleaning instructions.

4.2.8.22 Expired Cleaning Media Alert

The cleaning media has expired.

Probable Cause: The last cleaning cartridge used in the drive has worn out:

Recommended Actions: 1. Discard the worn out cleaning cartridge.2. Wait for the current operation to finish.3. Then use a new cleaning cartridge.

4.2.8.23 Invalid Cleaning Media Alert

Invalid cleaning media type used.

Probable Cause: The last cleaning cartridge used in the drive was an invalid type:

Recommended Actions: 1. Do not use this cleaning cartridge in this drive.2. Wait for the current operation to finish.3. Then use a valid cleaning cartridge.

4.2.8.24 Retention Requested Alert

The drive is having severe trouble reading or writing, which will be resolved by a retention cycle.

Probable Cause: The drive has requested a retention operation.

4.2.8.25 Dual-Port Interface Error Alert

Failure of one interface port in a dual-port configuration (i.e., Fibre Channel)

Probable Cause: A redundant interface port on the drive has failed.

4.2.8.26 Drive Maintenance Alert

The drive requires preventive maintenance (not cleaning).

Probable Cause: Preventive maintenance of the drive is required.

Recommended Actions: Check the drive users manual for device specific preventive maintenance tasks or call the drive supplier help line.

4.2.8.27 Hardware A Alert

The drive has a hardware fault that requires reset to recover.

Probable Cause: The drive has a hardware fault

Recommended Actions: 1. Eject the media or magazine.2. Reset the drive.3. Restart the operation.

4.2.8.28 Hardware B Alert

The drive has a hardware fault that is not read/write related or requires a power cycle to recover.

Probable Cause: The drive has a hardware fault

Recommended Actions: 1. Turn the drive off and then on again. 2. Restart the operation. 3. If the problem persists, call the drive supplier help line.

4.2.8.29 Interface Alert

The drive has identified an interface fault.

Probable Cause: Bad cable or drive interface.

Recommended Actions: 1. Check the cables and cable connections. 2. Restart the operation.

4.2.8.30 Eject Media Alert

Error recovery action: Media Ejected

Recommended Actions: 1. Eject the media or magazine 2. Insert the media or magazine again. 3. Restart the operation.

4.2.8.31 Download Failure Alert

Firmware download failed.

Probable Cause: The firmware download has failed because you have tried to use the incorrect firmware for this drive.

Recommended Actions: Obtain the correct firmware and try again.

4.2.8.32 Loader Hardware A Alert

Loader mechanism is having trouble communicating with the drive.

Probable Cause: The changer mechanism is having difficulty communicating with the drive:

Recommended Actions: 1. Turn the autoloader off then on. 2. Restart the operation. 3. If a problem persists, call the drive supplier help line.

4.2.8.33 Loader Stray Media Alert

Stray media left in loader after previous error recovery.

Probable Cause: A media has been left in the autoloader by a previous hardware fault:

Recommended Actions: 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.

4.2.8.34 Loader Hardware B Alert

Loader mechanism has a hardware fault.

Probable Cause: There is a problem with the autoloader mechanism.

4.2.8.35 Loader Door Alert

Changer door open.

Probable Cause: The operation has failed because the autoloader door is open:

Recommended Actions: 1. Clear any obstructions from the autoloader door. 2. Eject the magazine and then insert it again. 3. If the fault does not clear, turn the autoloader off and then on again. 4. If the problem persists, call the drive supplier help line.

4.2.8.36 Loader Hardware C Alert

The loader mechanism has a hardware fault that is not mechanically related.

Probable Cause: The autoloader has a hardware fault:

Recommended Actions: 1. Turn the autoloader off and then on again. 2. Restart the operation. 3. If the problem persists, call the drive supplier help line. Check the autoloader users manual for device specific instructions on turning the device power on and off.

4.2.8.37 Loader Magazine Alert

Loader magazine not present.

Probable Cause: The autoloader cannot operate without the magazine:

Recommended Actions: 1. Insert the magazine into the autoloader. 2. Restart the operation.

4.2.8.38 Loader Predictive Failure Alert

Predictive failure of loader mechanism hardware

Recommended Actions: A hardware failure of the changer mechanism is predicted. Call the drive supplier help line.

4.2.8.39 Load Statistics Alert

Drive or library powered down with media loaded.

Probable Cause: Media statistics have been lost at some time in the past.

4.2.8.40 Media Directory Invalid at Unload Alert

Error preventing the media directory being updated on unload.

Probable Cause: The directory on the media just unloaded has been corrupted.

Recommended Actions: The directory can be rebuilt by reading all the data.

4.2.8.41 Media System area Write Failure Alert

Write errors while writing the system area on unload.

Probable Cause: The media just unloaded could not write its system area successfully:

Recommended Actions: 1. Copy data to another cartridge. 2. Discard the old cartridge.

4.2.8.42 Media System Area Read Failure Alert

Read errors while reading the system area on load.

Probable Cause: The media system area could not be read successfully at load time:

Recommended Actions: Copy data to another cartridge.

4.2.8.43 No Start of Data Alert

Media damaged, bulk erased, or incorrect format.

Probable Cause: The start of data could not be found on the media:

Recommended Actions: 1. Check that you are using the correct format media. 2. Discard the media or return the media to your supplier.

4.2.8.44 Loading Failure Alert

The drive is unable to load the media

Probable Cause: The operation has failed because the media cannot be loaded and threaded.

Recommended Actions: 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.

4.2.8.45 Library Hardware A Alert

Changer mechanism is having trouble communicating with the internal drive

Probable Cause: The library mechanism is having difficulty communicating with the drive:

Recommended Actions: 1. Turn the library off then on. 2. Restart the operation. 3. If the problem persists, call the library supplier help line.

4.2.8.46 Library Hardware B Alert

Changer mechanism has a hardware fault

Recommended Actions: There is a problem with the library mechanism. If problem persists, call the library supplier help line.

4.2.8.47 Library Hardware C Alert

The changer mechanism has a hardware fault that requires a reset to recover.

Probable Cause: The library has a hardware fault

Recommended Actions: 1. Reset the library. 2. Restart the operation. Check the library users manual for device specific instructions on resetting the device.

4.2.8.48 Library Hardware D Alert

The changer mechanism has a hardware fault that is not mechanically related or requires a power cycle to recover.

Probable Cause: The library has a hardware fault:

Recommended Actions: 1. Turn the library off then on again. 2. Restart the operation. 3. If the problem persists, call the library supplier help line. Check the library users manual for device specific instructions on turning the device power on and off.

4.2.8.49 Library Diagnostic Required Alert

The changer mechanism may have a hardware fault which would be identified by extended diagnostics.

Probable Cause: The library mechanism may have a hardware fault.

Recommended Actions: Run extended diagnostics to verify and diagnose the problem. Check the library users manual for device specific instructions on running extended diagnostic tests.

4.2.8.50 Library Interface Alert

The library has identified an interface fault

Probable Cause: Bad cable

Recommended Actions: 1. Check the cables and connections. 2. Restart the operation.

4.2.8.51 Failure Prediction Alert

Predictive failure of library hardware

Recommended Actions: A hardware failure of the library is predicted. Call the library supplier help line.

4.2.8.52 Library Maintenance Alert

Library preventative maintenance required.

Probable Cause: Operator action

Recommended Actions: Preventive maintenance of the library is required. Check the library users manual for device specific preventative maintenance tasks, or call your library supplier help line.

4.2.8.53 Library Humidity Limits

Library humidity limits exceeded

Probable Cause: Library humidity range is outside the operational conditions

4.2.8.54 Library Voltage Limits Alert

Library voltage limits exceeded

Probable Cause: Potential problem with a power supply.

4.2.8.55 Library Stray Media Alert

Stray cartridge left in library after previous error recovery

Probable Cause: Cartridge left in picker or drive

Recommended Actions: 1. Insert an empty magazine to clear the fault. 2. If the fault does not clear, turn the library off and then on again. 3. If the problem persists, call the library supplier help line.

4.2.8.56 Library Pick Retry Alert

Operation to pick a cartridge from a slot had to perform an excessive number of retries before succeeding

Probable Cause: There is a potential problem with the drive ejecting cartridges or with the library mechanism picking a cartridge from a slot.

Recommended Actions: 1. Run diagnostics to determine the health of the Library. 2. If the problem persists, call the library supplier help line.

4.2.8.57 Library Place Retry Alert

Operation to place a cartridge in a slot had to perform an excessive number of retries before succeeding

Probable Cause: Worn cartridge or bad storage slot/ magazine

Recommended Actions: 1. No action needs to be taken at this time. 2. If the problem persists, call the library supplier help line.

4.2.8.58 Library Load Retry Alert

Operation to load a cartridge in a drive had to perform an excessive number of retries before succeeding

Probable Cause: Worn cartridge or picker

Recommended Actions: Run diagnostics to determine the health of the library.

4.2.8.59 Library Door Alert

Library door open is preventing the library from functioning

Probable Cause: The library has failed because the door is open:

Recommended Actions: 1. Clear any obstructions from the library door. 2. Close the library door. 3. If the problem persists, call the library supplier help line.

4.2.8.60 Library Mailslot Alert

Mechanical problem with import/export mailslot

Probable Cause: There is a mechanical problem with the library media mailslot.

Recommended Actions: Check for wedged storage media in import/export mailslot

4.2.8.61 Library Magazine Alert

Library magazine not present

Probable Cause: Administrator has removed the library's magazine.

Recommended Actions: 1. Insert the magazine into the library. 2. Restart the operation.

4.2.8.62 Library Security Alert

Library door opened then closed during operation

Probable Cause: Administrator is trying to remove or insert a storage media

4.2.8.63 Library Security Mode Alert

Library security mode changed

Probable Cause: Administrator changed security mode

Recommended Actions: 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.

4.2.8.64 Library Offline Alert

Library manually turned offline

Probable Cause: The library has been manually turned offline and is unavailable for use.

4.2.8.65 Library Drive Offline Alert

Library turned internal drive offline.

Probable Cause: Drive failure

Recommended Actions: A drive inside the library has been taken offline. This is for information purposes only. No action is required.

4.2.8.66 Library Scan Retry Alert

Operation to scan the bar code on a cartridge had to perform an excessive number of retries before succeeding

Probable Cause: There is a potential problem with the bar code label or the scanner hardware in the library mechanism.

Recommended Actions: 1. No action needs to be taken at this time. 2. If the problem persists, call the library supplier help line.

4.2.8.67 Library Inventory Alert

Inconsistent media inventory

Probable Cause: Media label has changed or bad Bar code scanner subsystem problem.

Recommended Actions: 1. Redo the library inventory to correct inconsistency. 2. Restart the operation. Check the applications users manual or the hardware users manual for specific instructions on redoing the library inventory.

4.2.8.68 Library Illegal Operation Alert

Illegal operation detected

Probable Cause: A library operation has been attempted that is invalid at this time.

4.2.8.69 Pass Through Mechanism Failure Alert

Error occurred in pass-through mechanism during self test or while attempting to transfer a cartridge between library modules

Probable Cause: A failure has occurred in the cartridge pass- through mechanism between two library modules.

4.2.8.70 Cartridge in Pass-through Mechanism Alert

Cartridge left in the pass-through mechanism between two library modules

Recommended Actions: 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.

4.2.8.71 Unreadable barcode Labels Alert

Unable to read a bar code label on a cartridge during library inventory/scan

Probable Cause: Bad Bar Code Labels or Scanner

Recommended Actions: The library was unable to read the bar code on a cartridge.

4.3 Health and Fault Management Considerations

Not defined in this standard.

4.4 Methods of the Profile

Not defined in this standard.

4.5 Use Cases

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.

4.5.1 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 4.2.7 "Durable Names and Correlatable IDs of the Profile". Specifically, NameFormat shall be set to "VendorModelSerial" and the Name shall be of the form Vendor+Product+Serial

4.5.2 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 Storage Library Capacity Profile, minimum and maximum slot capacities for a Storage Library are modeled in the ConfigurationCapacity, which is described in 6 "Storage Library Capacity Profile". Since this use case relies on an optional part of the profile, it may not be supported by each agent implementation.

4.5.3 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.

4.5.4 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 a HostedService associated with the ComputerSystem that models the storage library.

4.5.5 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 Storage Library Capacity Profile 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 Storage Library Capacity Profile is not implemented.

4.5.6 Determine Drive Data Path Technology

Clients can discover the data path protocol of each drive within a storage library by following SystemDevice associations to CIM_LogicalPort instances, then examining instances of the subclass (for example, CIM_FCPort) and their properties. See *Storage Management Technical Specification, Part 3 Common Profiles, 1.7.0 Rev 5 6 FC Target Ports Profile* for more information on Fibre Channel connectivity.

4.5.7 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.

4.5.8 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) component profile (see 7 Limited Access Port Elements Profile).

4.5.9 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 Element Counting Profile to quickly count PhysicalMedia and StorageMediaLocation instances. Note that this component profile is optional and experimental and may not be supported by some providers.

4.6 CIM Elements

Table 2 describes the CIM elements for Storage Library.

Table 2 - CIM Elements for Storage Library

Element Name	Requirement	Description
4.6.1 CIM_ChangerDevice	Mandatory	
4.6.2 CIM_Chassis	Mandatory	
4.6.3 CIM_ComputerSystem	Mandatory	'Top level' system that represents the whole Storage Library. Associated to RegisteredProfile.
4.6.4 CIM_ComputerSystemPackage	Mandatory	
4.6.5 CIM_ElementCapabilities	Optional	Class to implement the association between the top-level ComputerSystem representing a Storage Library and it's StorageLibraryCapabilities.
4.6.6 CIM_ElementSoftwareIdentity	Mandatory	
4.6.7 CIM_MediaAccessDevice	Mandatory	
4.6.8 CIM_PackagedComponent	Mandatory	
4.6.9 CIM_PhysicalMedia	Mandatory	
4.6.10 CIM_PhysicalMediaInLocation	Mandatory	

Storage Library Profile

Table 2 - CIM Elements for Storage Library

Element Name	Requirement	Description
4.6.11 CIM_ProtocolControllerForUnit	Mandatory	
4.6.12 CIM_Realizes	Conditional	Conditional requirement: Support for Inter-Library Port profile.
4.6.13 CIM_SCSIProtocolController	Mandatory	
4.6.14 CIM_SoftwareIdentity	Mandatory	
4.6.15 CIM_StorageLibraryCapabilities	Optional	Experimental. Describes the capabilities of the Storage Library represented by the top level ComputerSystem this is associated with.
4.6.16 CIM_StorageMediaLocation	Mandatory	
4.6.17 CIM_SystemDevice (System to Changer Device)	Mandatory	This association links Changer to the scoping system.
4.6.18 CIM_SystemDevice (System to MediaAccessDevice)	Mandatory	This association links MediaAccessDevice To the scoping system.
4.6.19 CIM_SystemDevice (System to SCSIProtocolController)	Mandatory	This association links SCSIProtocolController To the scoping system.
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.
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. 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. Deprecated WQL -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.OperationalStatus <> SourceInstance.OperationalStatus	Mandatory	Deprecated. Deprecated WQL -Change in OperationalStatus for a Changer Device.
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ComputerSystem AND PreviousInstance.CIM_ComputerSystem::O perationalStatus <> SourceInstance.CIM_ComputerSystem::Ope rationalStatus	Optional	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	CQL -Change in OperationalStatus for a media access device.
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ChangerDevice AND PreviousInstance.CIM_ChangerDevice::Ope rationalStatus <> SourceInstance.CIM_ChangerDevice::Opera tionalStatus	Optional	CQL -Change in OperationalStatus for a Changer Device.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML1'	Optional	Experimental. The drive is having severe trouble reading.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML2'	Optional	Experimental. The drive is having severe trouble writing.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML3'	Optional	Experimental. The drive had a hard read or write error.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML4'	Optional	Experimental. Media can no longer be written/read, or performance is severely degraded.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML5'	Optional	Experimental. The drive can no longer read data from the storage media.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML6'	Optional	Experimental. The drive can no longer write data to the media.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML7'	Optional	Experimental. The media has exceeded its specified life.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML8'	Optional	Experimental. The cartridge is not data-grade. Any data you write to the media is at risk. Replace the cartridge with a data-grade media.

Storage Library Profile

Table 2 - CIM Elements for Storage Library

Element Name	Requirement	Description
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML9'	Optional	Experimental. Write command is attempted to a write protected media.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML10'	Optional	Experimental. Manual or software unload attempted when prevent media removal is on.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML11'	Optional	Experimental. Cleaning media loaded into drive.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML12'	Optional	Experimental. Attempted load of unsupported media format (e.g., DDS2 in DDS1 drive).
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML13'	Optional	Experimental. Tape snapped/cut in the drive where media can be de-mounted.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML14'	Optional	Experimental. Tape snapped/cut in the drive where media cannot be de-mounted.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML15'	Optional	Experimental. Memory chip failed in cartridge.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML16'	Optional	Experimental. Manual or forced eject while drive actively writing or reading.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML17'	Optional	Experimental. Media loaded that is read-only format.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML18'	Optional	Experimental. Drive powered down while loaded, or permanent error prevented the directory being updated.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML19'	Optional	Experimental. Media may have exceeded its specified number of passes.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML20'	Optional	Experimental. The drive thinks it has a head clog or needs cleaning.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML21'	Optional	Experimental. The drive is ready for a periodic cleaning.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML22'	Optional	Experimental. The cleaning media has expired.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML23'	Optional	Experimental. Invalid cleaning media type used.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML24'	Optional	Experimental. The drive is having severe trouble reading or writing, which will be resolved by a retention cycle.

Table 2 - CIM Elements for Storage Library

Element Name	Requirement	Description
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML25'	Optional	Experimental. Failure of one interface port in a dual-port configuration (i.e., Fibre Channel).
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='Core13'	Optional	Experimental. Fan failure inside drive mechanism or drive enclosure.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='Core14'	Optional	Experimental. Power supply unit failure inside the drive enclosure or rack subsystem.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='Core15'	Optional	Experimental. Power consumption of the drive is outside specified range.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML26'	Optional	Experimental. The drive requires preventive maintenance (not cleaning).
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML27'	Optional	Experimental. The drive has a hardware fault that requires reset to recover.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML28'	Optional	Experimental. The drive has a hardware fault that is not read/write related or requires a power cycle to recover.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML29'	Optional	Experimental. The drive has identified an interface fault.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML30'	Optional	Experimental. Error recovery action: Media Ejected.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML31'	Optional	Experimental. Firmware download failed.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='Core2'	Optional	Experimental. Drive humidity or temperature limits exceeded.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='Core17'	Optional	Experimental. Drive voltage limits exceeded.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='Core18'	Optional	Experimental. Predictive failure of drive hardware.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='Core19'	Optional	Experimental. The drive may have a hardware fault that may be identified by extended diagnostics.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML32'	Optional	Experimental. Loader mechanism is having trouble communicating with the drive.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML33'	Optional	Experimental. Stray media left in loader after previous error recovery.

Storage Library Profile

Table 2 - CIM Elements for Storage Library

Element Name	Requirement	Description
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML34'	Optional	Experimental. Loader mechanism has a hardware fault.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML35'	Optional	Experimental. Changer door open.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML36'	Optional	Experimental. The loader mechanism has a hardware fault that is not mechanically related.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML37'	Optional	Experimental. Loader magazine not present.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML38'	Optional	Experimental. Predictive failure of loader mechanism hardware.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML39'	Optional	Experimental. Drive or library powered down with media loaded.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML40'	Optional	Experimental. Error preventing the media directory being updated on unload.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML41'	Optional	Experimental. Write errors while writing the system area on unload.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML42'	Optional	Experimental. Read errors while reading the system area on load.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML43'	Optional	Experimental. Media damaged, bulk erased, or incorrect format.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML44'	Optional	Experimental. The drive is unable to load the media.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML45'	Optional	Experimental. Changer mechanism is having trouble communicating with the internal drive.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML46'	Optional	Experimental. Changer mechanism has a hardware fault.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML47'	Optional	Experimental. The changer mechanism has a hardware fault that requires a reset to recover.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML48'	Optional	Experimental. The changer mechanism has a hardware fault that is not mechanically related or requires a power cycle to recover.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML49'	Optional	Experimental. The changer mechanism may have a hardware fault which would be identified by extended diagnostics.

Table 2 - CIM Elements for Storage Library

Element Name	Requirement	Description
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML50'	Optional	Experimental. The library has identified an interface fault.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML51'	Optional	Experimental. Predictive failure of library hardware.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML52'	Optional	Experimental. Library preventative maintenance required.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML53'	Optional	Experimental. Library humidity limits exceeded.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='Core2'	Optional	Experimental. Library temperature limits exceeded.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML54'	Optional	Experimental. Library voltage limits exceeded.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML55'	Optional	Experimental. Stray cartridge left in library after previous error recovery.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML56'	Optional	Experimental. Operation to pick a cartridge from a slot had to perform an excessive number of retries before succeeding.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML57'	Optional	Experimental. Operation to place a cartridge in a slot had to perform an excessive number of retries before succeeding.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML58'	Optional	Experimental. Operation to load a cartridge in a drive had to perform an excessive number of retries before succeeding.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML59'	Optional	Experimental. Library door open is preventing the library from functioning.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML60'	Optional	Experimental. Mechanical problem with import/export mailslot.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML61'	Optional	Experimental. Library magazine not present.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML62'	Optional	Experimental. Library door opened then closed during operation.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML63'	Optional	Experimental. Library security mode changed.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML64'	Optional	Experimental. Library manually turned offline.

Storage Library Profile

Table 2 - CIM Elements for Storage Library

Element Name	Requirement	Description
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML65'	Optional	Experimental. Library turned internal drive offline.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML66'	Optional	Experimental. Operation to scan the bar code on a cartridge had to perform an excessive number of retries before succeeding.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML67'	Optional	Experimental. Inconsistent media inventory.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML68'	Optional	Experimental. Illegal operation detected.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='Core1'	Optional	Experimental. Failure of one interface port in a dual-port configuration.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML69'	Optional	Experimental. Error occurred in pass-through mechanism during self test or while attempting to transfer a cartridge between library modules.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML70'	Optional	Experimental. Cartridge left in the pass-through mechanism between two library modules.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML71'	Optional	Experimental. Unable to read a bar code label on a cartridge during library inventory/scan,

4.6.1 CIM_ChangerDevice

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 3 describes class CIM_ChangerDevice.

Table 3 - 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.
StatusDescriptions		Optional	Additional information related to the values in OperationalStatus.

4.6.2 CIM_Chassis

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 4 describes class CIM_Chassis.

Table 4 - 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	

4.6.3 CIM_ComputerSystem

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Shall be associated to RegisteredProfile using ElementConformsToProfile association. The RegisteredProfile instance shall have RegisteredName set to 'Storage Library', RegisteredOrganization set to 'SNIA', and RegisteredVersion set to '1.6.0'.

Table 5 describes class CIM_ComputerSystem.

Table 5 - SMI Referenced Properties/Methods for CIM_ComputerSystem

Properties	Flags	Requirement	Description & Notes
CreationClassName		Mandatory	
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.

Table 5 - SMI Referenced Properties/Methods for CIM_ComputerSystem

Properties	Flags	Requirement	Description & Notes
PrimaryOwnerContact	M	Optional	Contact details for storage library owner.
PrimaryOwnerName	M	Optional	Owner of the storage library.

4.6.4 CIM_ComputerSystemPackage

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 6 describes class CIM_ComputerSystemPackage.

Table 6 - SMI Referenced Properties/Methods for CIM_ComputerSystemPackage

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	
Dependent		Mandatory	

4.6.5 CIM_ElementCapabilities

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Optional

Table 7 describes class CIM_ElementCapabilities.

Table 7 - SMI Referenced Properties/Methods for CIM_ElementCapabilities

Properties	Flags	Requirement	Description & Notes
Capabilities		Mandatory	
ManagedElement		Mandatory	

4.6.6 CIM_ElementSoftwareIdentity

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 8 describes class CIM_ElementSoftwareIdentity.

Table 8 - SMI Referenced Properties/Methods for CIM_ElementSoftwareIdentity

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	
Dependent		Mandatory	Reference to Media Access Device or Changer Device.

4.6.7 CIM_MediaAccessDevice

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 9 describes class CIM_MediaAccessDevice.

Table 9 - SMI Referenced Properties/Methods for CIM_MediaAccessDevice

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		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	

4.6.8 CIM_PackagedComponent

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 10 describes class CIM_PackagedComponent.

Table 10 - SMI Referenced Properties/Methods for CIM_PackagedComponent

Properties	Flags	Requirement	Description & Notes
GroupComponent		Mandatory	
PartComponent		Mandatory	

4.6.9 CIM_PhysicalMedia

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 11 describes class CIM_PhysicalMedia.

Table 11 - 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	

4.6.10 CIM_PhysicalMediaInLocation

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 12 describes class CIM_PhysicalMediaInLocation.

Table 12 - SMI Referenced Properties/Methods for CIM_PhysicalMediaInLocation

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	
Dependent		Mandatory	

4.6.11 CIM_ProtocolControllerForUnit

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 13 describes class CIM_ProtocolControllerForUnit.

Table 13 - SMI Referenced Properties/Methods for CIM_ProtocolControllerForUnit

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

4.6.12 CIM_Realizes

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Support for Inter-Library Port profile.

Table 14 describes class CIM_Realizes.

Table 14 - SMI Referenced Properties/Methods for CIM_Realizes

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	
Dependent		Mandatory	

4.6.13 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

Requirement: Mandatory

Table 15 describes class CIM_SCSIProtocolController.

Table 15 - 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	

4.6.14 CIM_SoftwareIdentity

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Storage Library Profile

Table 16 describes class CIM_SoftwareIdentity.

Table 16 - SMI Referenced Properties/Methods for CIM_SoftwareIdentity

Properties	Flags	Requirement	Description & Notes
InstanceID		Mandatory	
VersionString		Mandatory	The software or 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	

EXPERIMENTAL

4.6.15 CIM_StorageLibraryCapabilities

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Optional

Table 17 describes class CIM_StorageLibraryCapabilities.

Table 17 - 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.
SupportedCapacityOptimizations		Mandatory	Experimental. This array of strings lists the capacity optimization techniques that are supported by the implementation. Valid string values are "none" "SNIA:Thin Provisioning" "SNIA:Data Compression" "SNIA:Data Deduplication".
SupportedViews		Mandatory	Experimental. This array of strings lists the view classes that are supported by the implementation. Valid string values are "none" "SNIA:MaskingMappingExposedDeviceView" "SNIA:SystemMediaLocationView".
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.

EXPERIMENTAL

4.6.16 CIM_StorageMediaLocation

Created By: Static
 Modified By: Static
 Deleted By: Static
 Requirement: Mandatory

Table 18 describes class CIM_StorageMediaLocation.

Table 18 - SMI Referenced Properties/Methods for CIM_StorageMediaLocation

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

4.6.17 CIM_SystemDevice (System to Changer Device)

Created By: Static
 Modified By: Static
 Deleted By: Static
 Requirement: Mandatory

Table 19 describes class CIM_SystemDevice (System to Changer Device).

Table 19 - SMI Referenced Properties/Methods for CIM_SystemDevice (System to Changer Device)

Properties	Flags	Requirement	Description & Notes
PartComponent		Mandatory	
GroupComponent		Mandatory	

4.6.18 CIM_SystemDevice (System to MediaAccessDevice)

Created By: Static
 Modified By: Static
 Deleted By: Static
 Requirement: Mandatory

Table 20 describes class CIM_SystemDevice (System to MediaAccessDevice).

Table 20 - SMI Referenced Properties/Methods for CIM_SystemDevice (System to MediaAccessDevice)

Properties	Flags	Requirement	Description & Notes
PartComponent		Mandatory	
GroupComponent		Mandatory	

4.6.19 CIM_SystemDevice (System to SCSIProtocolController)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 21 describes class CIM_SystemDevice (System to SCSIProtocolController).

Table 21 - SMI Referenced Properties/Methods for CIM_SystemDevice (System to SCSIProtocolController)

Properties	Flags	Requirement	Description & Notes
PartComponent		Mandatory	
GroupComponent		Mandatory	

STABLE

EXPERIMENTAL

5 Element Counting Profile

5.1 Synopsis

Profile Name: Storage Element Counting (Component Profile)

Version: 1.1.0

Organization: SNIA

Central Class: ConfigurationReportingService

Scoping Class: ComputerSystem (autonomous Storage Library)

Supported Profiles: Not defined in this standard.

5.2 Description

The Element Counting Profile 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.

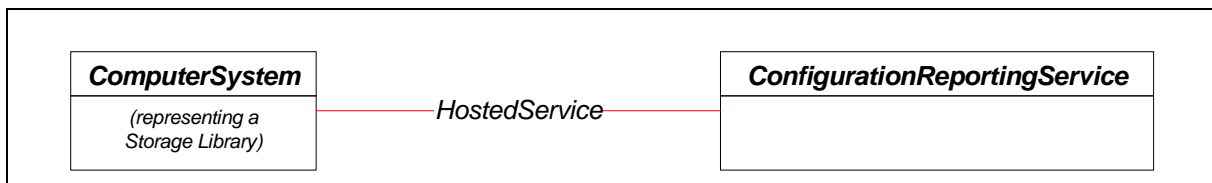


Figure 10 - Instance Diagram

5.2.1 Discovery

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

5.3 Health and Fault Management Considerations

Not defined in this standard.

5.4 Methods of the Profile

5.4.1 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 5.5 Use Cases 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 component profile, 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.

5.4.2 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 5.5 Use Cases for an overview and example).

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

[IN] uint16 InquiryType: see details in 5.4.1 GetClassTypes. “Supported” or “Installed” are valid enumerated values.

[IN] boolean Recursive: see details in 5.4.1 GetClassTypes. Generally, a value of “false” is expected.

[IN] CIM_ManagedElement REF Target: see details in 5.4.1 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 5.4.1 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.

5.4.3 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 5.5 Use Cases 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 5.4.1 GetClassTypes. “Supported” or “Installed” are valid enumerated values.

[IN] boolean Recursive: see details in 5.4.1 GetClassTypes. Generally, a value of “false” is expected.

[IN] CIM_ManagedElement REF Target: in 5.4.1 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 5.4.1 GetClassTypes. The class type to be counted.

[IN] uint16 UnitType: see details in 5.4.1 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. “Unknown” indicates that the result cannot be determined for the given parameter combination at this time.

5.5 Use Cases

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.

5.6 CIM Elements

Table 22 describes the CIM elements for Storage Library Element Counting.

Table 22 - CIM Elements for Storage Library Element Counting

Element Name	Requirement	Description
5.6.1 CIM_ConfigurationReportingService	Mandatory	
5.6.2 CIM_HostedService	Mandatory	

5.6.1 CIM_ConfigurationReportingService

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 23 describes class CIM_ConfigurationReportingService.

Table 23 - SMI Referenced Properties/Methods for CIM_ConfigurationReportingService

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

5.6.2 CIM_HostedService

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: 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

Element Counting Profile

EXPERIMENTAL

6 Storage Library Capacity Profile

6.1 Synopsis

Profile Name: Storage Library Capacity (Component Profile)

Version: 1.1.0

Organization: SNIA

Central Class: ConfigurationCapacity

Scoping Class: ComputerSystem (autonomous Storage Library)

Supported Profiles: Not defined in this standard.

6.2 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 11 illustrates the use of ConfigurationCapacity and ElementCapacity in conjunction with the basic storage library profile.

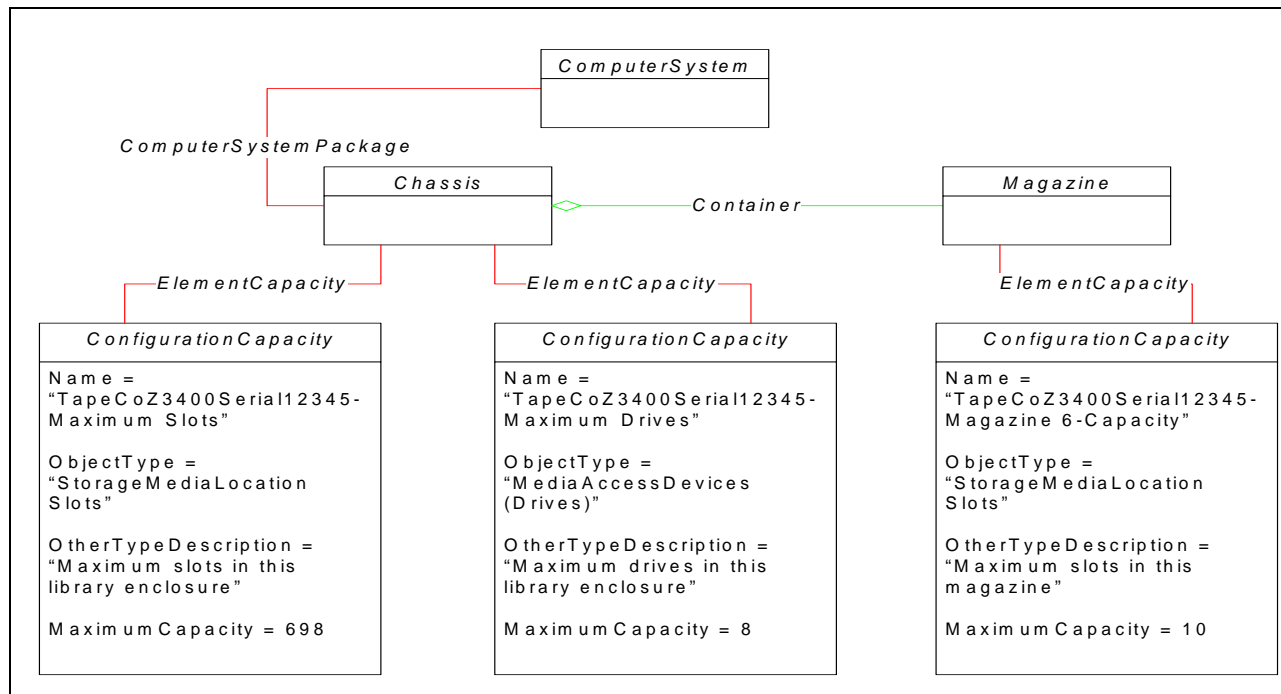


Figure 11 - Storage Library Capacity Instance Diagram

6.3 Health and Fault Management Considerations

Not defined in this standard.

6.4 Methods of the Profile

Not defined in this standard.

6.5 Use Cases

Not defined in this standard.

6.6 CIM Elements

Table 25 describes the CIM elements for Storage Library Capacity.

Table 25 - CIM Elements for Storage Library Capacity

Element Name	Requirement	Description
6.6.1 CIM_ConfigurationCapacity	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.
6.6.2 CIM_ElementCapacity	Mandatory	

6.6.1 CIM_ConfigurationCapacity

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 26 describes class CIM_ConfigurationCapacity.

Table 26 - 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	

6.6.2 CIM_ElementCapacity

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 27 describes class CIM_ElementCapacity.

Table 27 - SMI Referenced Properties/Methods for CIM_ElementCapacity

Properties	Flags	Requirement	Description & Notes
Element		Mandatory	
Capacity		Mandatory	

EXPERIMENTAL

Storage Library Capacity Profile

STABLE

7 Limited Access Port Elements Profile

7.1 Synopsis

Profile Name: Storage Library Limited Access Port Elements (Component Profile)

Version: 1.2.0

Organization: SNIA

Central Class: LimitedAccessPort

Scoping Class: ComputerSystem (autonomous Storage Library)

Supported Profiles: Not defined in this standard.

7.2 Description

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

7.2.1 Instance Diagram

Figure 12 and Figure 13 show the relationship between LimitedAccessPorts and other portions of the Storage Library Profile.

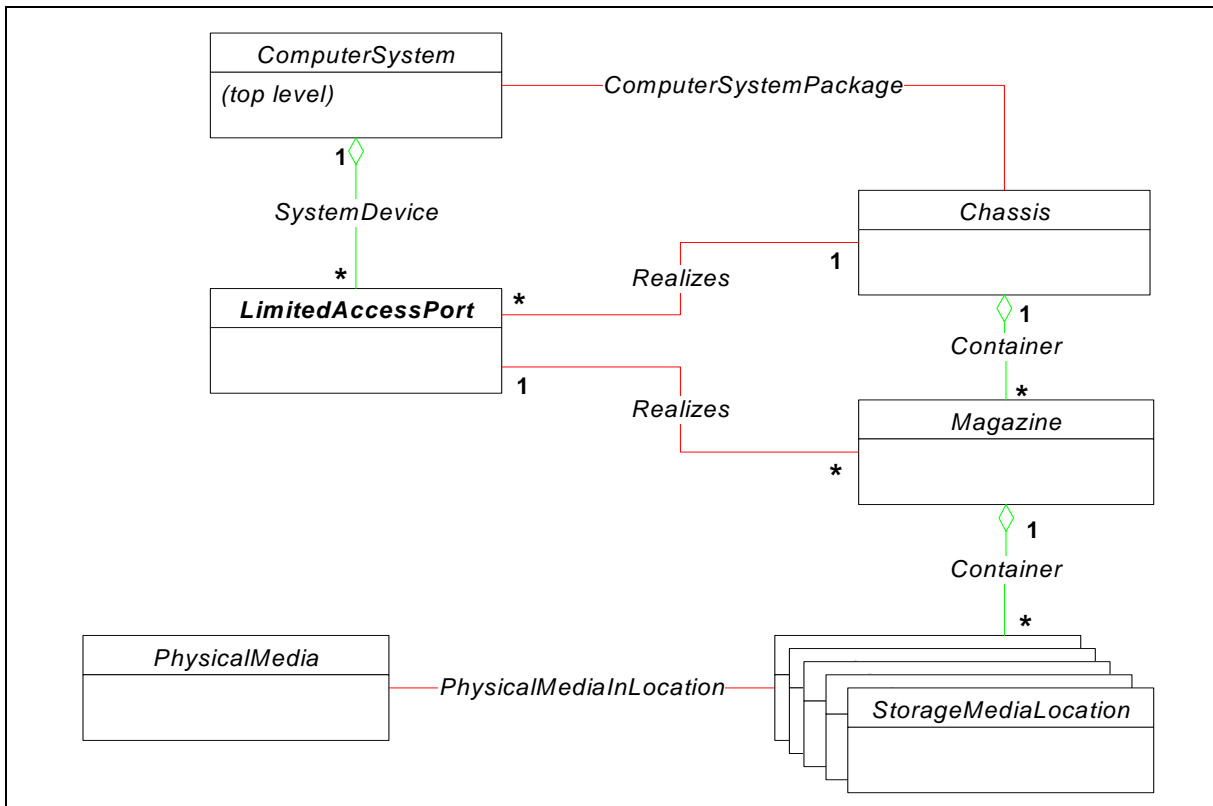


Figure 12 - Tape Libraries with Magazines in LimitedAccessPorts

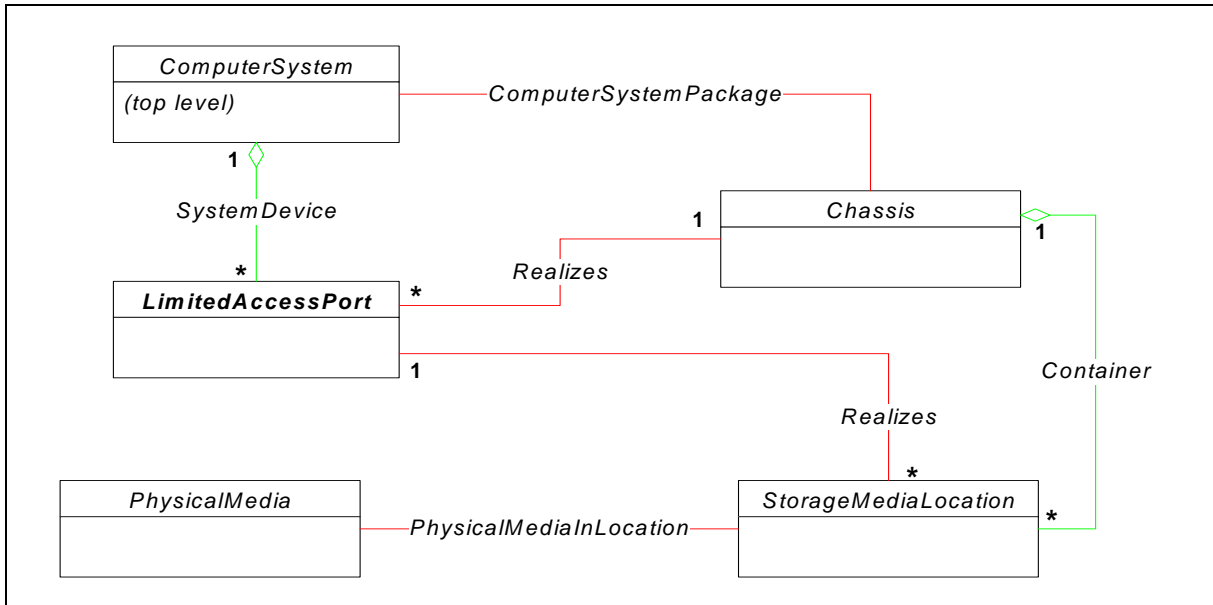


Figure 13 - Tape Libraries with no Magazines in LimitedAccessPorts

7.3 Health and Fault Management Considerations

Not defined in this standard.

7.4 Methods of the Profile

Not defined in this standard.

7.5 Use Cases

Not defined in this standard.

7.6 CIM Elements

Table 28 describes the CIM elements for Storage Library Limited Access Port Elements.

Table 28 - CIM Elements for Storage Library Limited Access Port Elements

Element Name	Requirement	Description
7.6.1 CIM_Container	Mandatory	The containment relationship of Magazines within a Chassis or StorageMediaLocations within a Magazine.
7.6.2 CIM_LimitedAccessPort	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.
7.6.3 CIM_Magazine	Mandatory	
7.6.4 CIM_Realizes	Mandatory	The relationship between a LimitedAccessPort and the StorageMediaLocations, Magazines or Chassis to which it has access.
7.6.5 CIM_SystemDevice	Mandatory	The relationship between a LimitedAccessPort and its hosting top-level ComputerSystem which represents the Storage Library.

Table 28 - CIM Elements for Storage Library Limited Access Port Elements

Element Name	Requirement	Description
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	Mandatory	CQL -Change in OperationalStatus of a LimitedAccessPort.

7.6.1 CIM_Container

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 29 describes class CIM_Container.

Table 29 - SMI Referenced Properties/Methods for CIM_Container

Properties	Flags	Requirement	Description & Notes
GroupComponent		Mandatory	
PartComponent		Mandatory	

7.6.2 CIM_LimitedAccessPort

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 30 describes class CIM_LimitedAccessPort.

Table 30 - SMI Referenced Properties/Methods for CIM_LimitedAccessPort

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	
CreationClassName		Mandatory	
SystemName		Mandatory	
DeviceID		Mandatory	

Table 30 - SMI Referenced Properties/Methods for CIM_LimitedAccessPort

Properties	Flags	Requirement	Description & Notes
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.

7.6.3 CIM_Magazine

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 31 describes class CIM_Magazine.

Table 31 - SMI Referenced Properties/Methods for CIM_Magazine

Properties	Flags	Requirement	Description & Notes
CreationClassName		Mandatory	
Tag		Mandatory	
LocationType		Mandatory	"Magazine".
LocationCoordinates		Mandatory	
MediaTypesSupported		Mandatory	
MediaCapacity		Mandatory	The maximum number of PhysicalMedia that this StorageMediaLocation can hold.
PhysicalLabels		Optional	
LabelStates		Optional	
LabelFormats		Optional	

7.6.4 CIM_Realizes

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 32 describes class CIM_Realizes.

Table 32 - SMI Referenced Properties/Methods for CIM_Realizes

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	
Dependent		Mandatory	

7.6.5 CIM_SystemDevice

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 33 describes class CIM_SystemDevice.

Table 33 - SMI Referenced Properties/Methods for CIM_SystemDevice

Properties	Flags	Requirement	Description & Notes
GroupComponent		Mandatory	
PartComponent		Mandatory	

STABLE

Annex A (informative) SMI-S Information Model

This standard is based on DMTF's CIM schema, version 2.45.0. The DMTF schema is available in the machine-readable Managed Object Format (MOF) format. DMTF MOFs are simultaneously released both as an "Experimental" and a "Final" version of the schema. This provides developers with early access to experimental parts of the models. Both versions are available at

<http://www.dmtf.org/standards/cim>

Content marked as "Experimental" or "Implemented" may be based on DMTF's Experimental MOFs.

