Storage Management Technical Specification, Part 7 Media Libraries

Version 1.6.0, Revision 4

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

10 February, 2012

Revision History

Revision 1

Date

16 July 2010

SCRs Incorporated and other changes

Virtual Tape Library Profile (SML-SMIS-SCR00026)
- Added CIM_SystemPackaging association between Virtual Tape Library ComputerSystem [3, 34] and its associated virtual StorageMediaLocation instances

Library Views profile (SML-SMIS-SCR00024)
- Changed the CIM-VERSION to 2.27
- Promoted all SNIA_ classes to CIM_ classes
- Added an Inheritance diagram to the profile
- Did a lot of editing of diagrams and text for the promotion to CIM_ classes

Comments

Editorial notes and DRAFT material are displayed.

Revision 2

Date

7 October 2010

SCRs Incorporated and other changes

None.

Comments

Editorial notes and DRAFT material are displayed.

Revision 3

Date

10 March 2011

SCRs Incorporated and other changes

Storage Library (SML-SMIS-SCR00028)
- Added SupportedCapacityOptimizations and SupportedViews to StorageLibraryCapabilities
- Marked StorageLibraryCapabilities as Experimental (since the mof is Experimental)
- Fixed a number of problems with the Supported Profile list

Virtual Tape Library (SML-SMIS-SCR00028)
- Added ImplementationCapabilities to the profile (for Capacity Optimizations) as Experimental
- Promoted SystemPackaging from the VTL ComputerSystem to the StorageMediaLocation to Experimental
- Fixed a number of problems with the Supported Profile list

Library Views Profile (SML-SMIS-SCR00024)
- Changed the CIM-VERSION to 2.28
- Backed out the promotion of CIM_MediaLocationView and CIM_SystemMediaLocationView as CIM_ classes (now SNIA_ classes)
- Modified the Inheritance diagram to the profile
- Did some editing of diagrams and text for the roll-back of the promotion to CIM_ classes

VTL Capacity and Reporting (SMIS-160-Draft-SCR00006)
- Added this as a new Profile
- Promoted the Profile to Experimental

SMI-S Information Model (TSG-SMIS-SCR00294)
- Added this Annex

Comments

Editorial notes are displayed.
DRAFT material was hidden.

Revision 4

Date

28 September 2011

SCRs Incorporated and other changes

VTL Capacity and Reporting (SMIS-160-Draft-SCR00006)
- Updated the profile filling out descriptions in the CIM Element Tables

Front matter
- Paragraph added to Maturity Level explanation, as requested by ISO editor for SMI-S 1.1.1.

Comments

Editorial notes and DRAFT material are 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/.

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.

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/.

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.

CHANGES TO THE SPECIFICATION

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

- Major Revision: A major revision of the specification represents a substantial change to the underlying scope or architecture of the SMI-S API. A major revision results in an increase in the version number of the version identifier (e.g., from version 1.x.x to version 2.x.x). There is no assurance of interoperability or backward compatibility between releases with different version numbers.

- Minor Revision: A minor revision of the specification represents a technical change to existing content or an adjustment to the scope of the SMI-S API. A minor revision results in an increase in the release number of the specification’s identifier (e.g., from x.1.x to x.2.x). Minor revisions with the same version number preserve interoperability and backward compatibility.

- Update: An update to the specification is limited to minor corrections or clarifications of existing specification content. An update will result in an increase in the third component of the release identifier (e.g., from x.x.1 to x.x.2). Updates with the same version and minor release levels preserve interoperability and backward compatibility.

TYPOGRAPHICAL CONVENTIONS

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.

EXPERIMENTAL

Experimental content appears here.

EXPERIMENTAL

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.

IMPLEMENTED

Implemented content appears here.

IMPLEMENTED

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.

STABLE

Stable content appears here.

STABLE

Figure 3 - Stable Maturity Level Tag

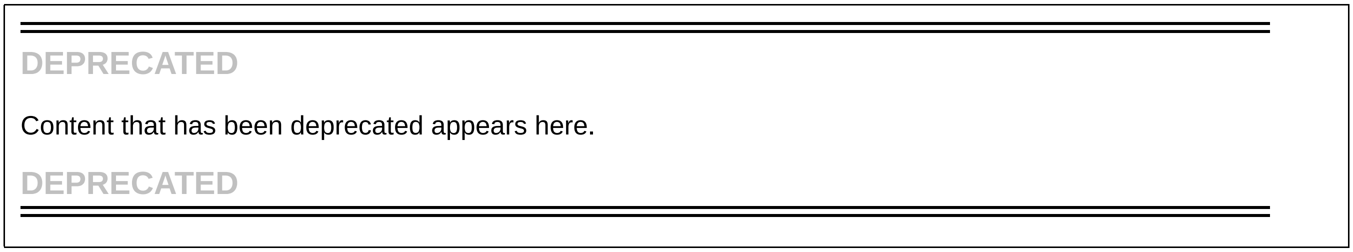
Finalized Maturity Level

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.



DEPRECATED

Content that has been deprecated appears here.

DEPRECATED

Figure 4 - Deprecated Tag

USAGE

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

1) Any text, diagram, chart, table or definition reproduced shall be reproduced in its entirety with no alteration.

2) Any document, printed or electronic, in which material from this document (or any portion hereof) is reproduced shall acknowledge the SNIA copyright on that material, and shall credit the SNIA for granting permission for its reuse.

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

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

Contents

List of Tables

List of Figures

Foreword

Storage Library Profile and related subprofiles 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 and subprofiles referenced in Storage Library profile are specified in Part 2 Common Profiles book.

Parts of this Standard

This standard is subdivided in the following parts:

- *Storage Management Technical Specification, Overview, 1.6.0 Rev 4*
- *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4*
- *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4*
- *Storage Management Technical Specification, Part 3 Block Devices, 1.6.0 Rev 4*
- *Storage Management Technical Specification, Part 4 Filesystems, 1.6.0 Rev 4*
- *Storage Management Technical Specification, Part 5 Fabric, 1.6.0 Rev 4*
- *Storage Management Technical Specification, Part 6 Host Elements, 1.6.0 Rev 4*
- *Storage Management Technical Specification, Part 7 Media Libraries, 1.6.0 Rev 4*

Acknowledgments

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

Organization Represented	*Name of Representative*
Brocade Communications Systems	John Crandall
EMC Corporation	Tony Fiorentino
	Mike Hadavi
Hitachi Data Systems	Steve Quinn
IBM	Jun Wei Zhang
Individual Contributor	Mike Walker
Individual Contributor	Paul von Behren
WBEM Solutions, Inc	Jim Davis

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, 425 Market Street, Suite 1020, San Francisco, CA 94105, U.S.A.

Clause 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.

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

Clause 2: Normative References

2.1 General

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

2.2 Approved references

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

2.3 References under development

Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4

Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4

Storage Management Technical Specification, Part 3 Block Devices, 1.6.0 Rev 4

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

2.4 Other references

DMTF DSP0214:2004 CIM Operations over HTTP

Clause 3: 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 1 Common Architecture, 1.6.0 Rev 4* and the following apply.

3.2 Definitions

3.2.1 Changer Device

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

3.2.2 Media Access Device

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

3.2.3 Storage Media Location

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

3.2.4 Storage Media Library

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

3.2.5 Limited Access Port

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

3.2.6 Library Capacity

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

3.2.7 Magazine

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

STABLE

Clause 4: Storage Library Profile

4.1 Description

The schema for a storage library provides the classes and associations necessary to represent various forms of removable media libraries. This profile 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 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 4.1.7 "Durable Names and Correlatable IDs of the Profile". All other objects do not require durable names and have instances within a single name space.

4.1.1 Instance Diagrams

The following instance diagrams represent five related views of the Storage Library Profile:

a) System Level

b) MediaAccessDevice and its physical and logical relationships

c) ChangerDevice and its connections to SoftwareIdentity, ProtocolController, and StorageMediaLocation

d) StorageMediaLocation and its relationship to PhysicalMedia and other physical classes

e) StorageMediaLocation and its required Realizes relationships.

4.1.2 System Level View

Figure 5: "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.

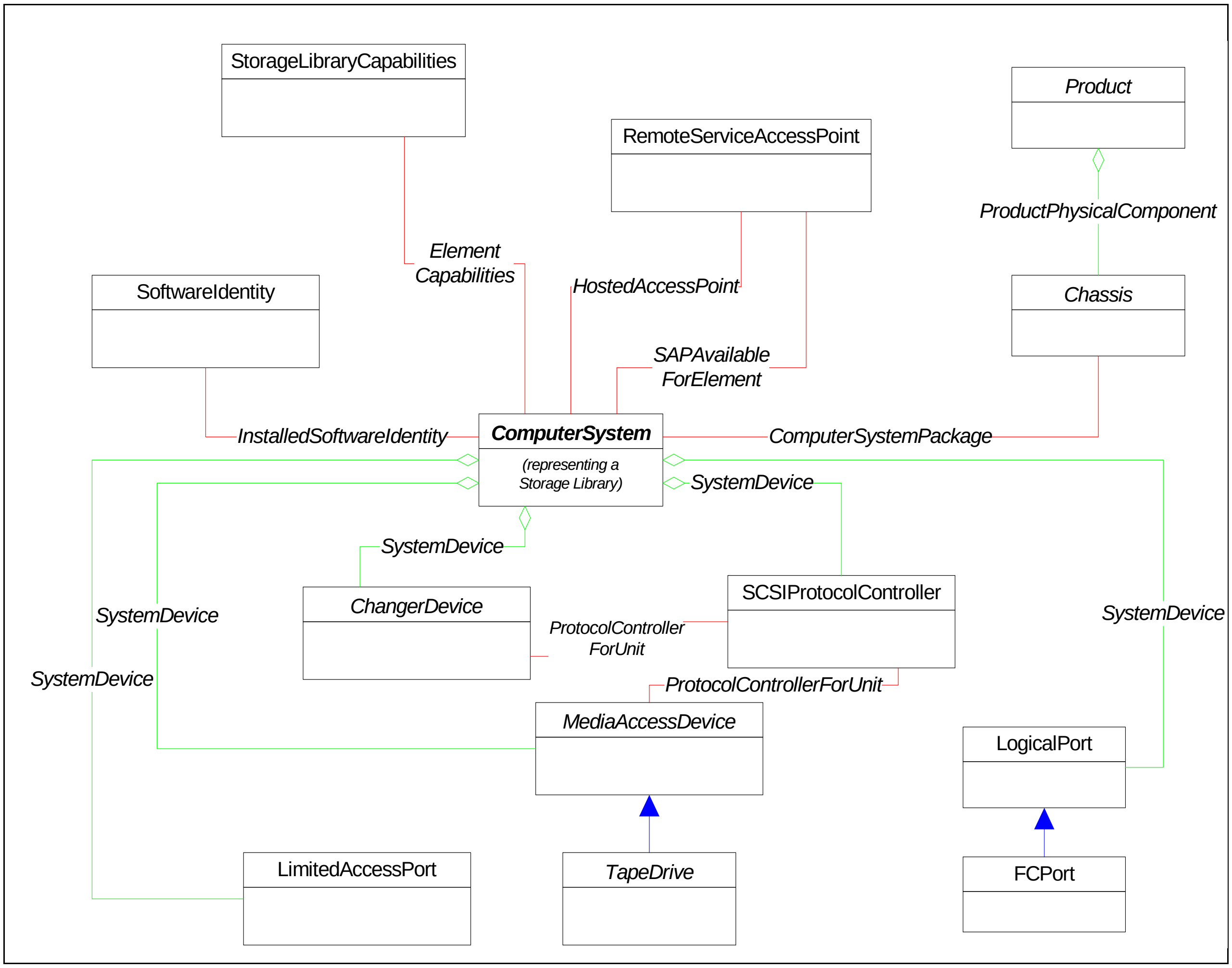


Figure 5 - Storage Library-centric Instance Diagram

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

4.1.3 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.

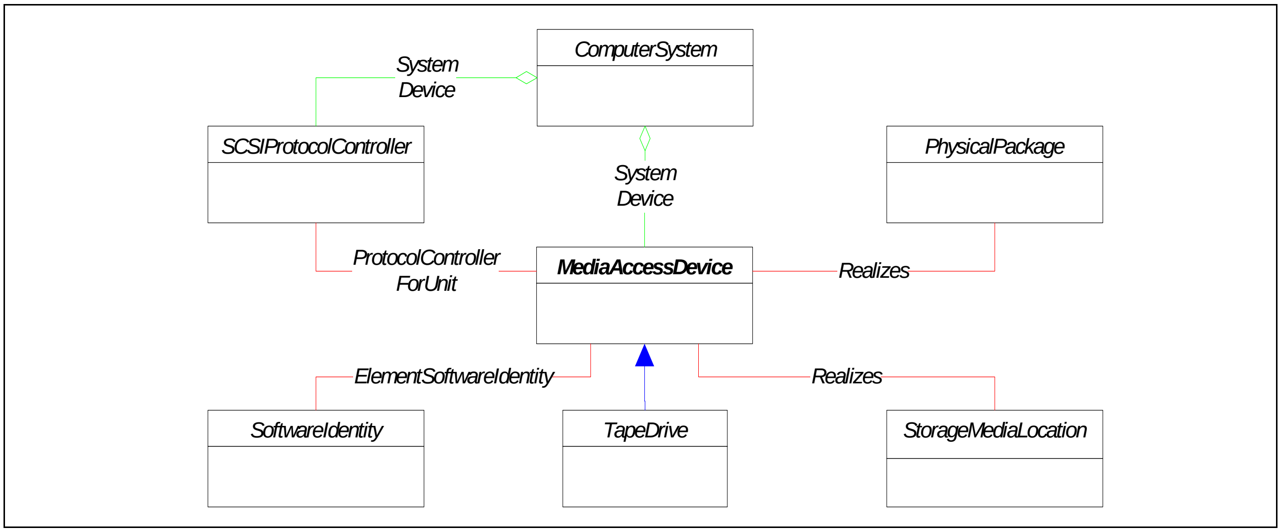


Figure 6 - MediaAccessDevice-centric Instance Diagram

4.1.4 ChangerDevice-centric View

Figure 7: "ChangerDevice-centric Instance Diagram" shows the required classes related to ChangerDevice.

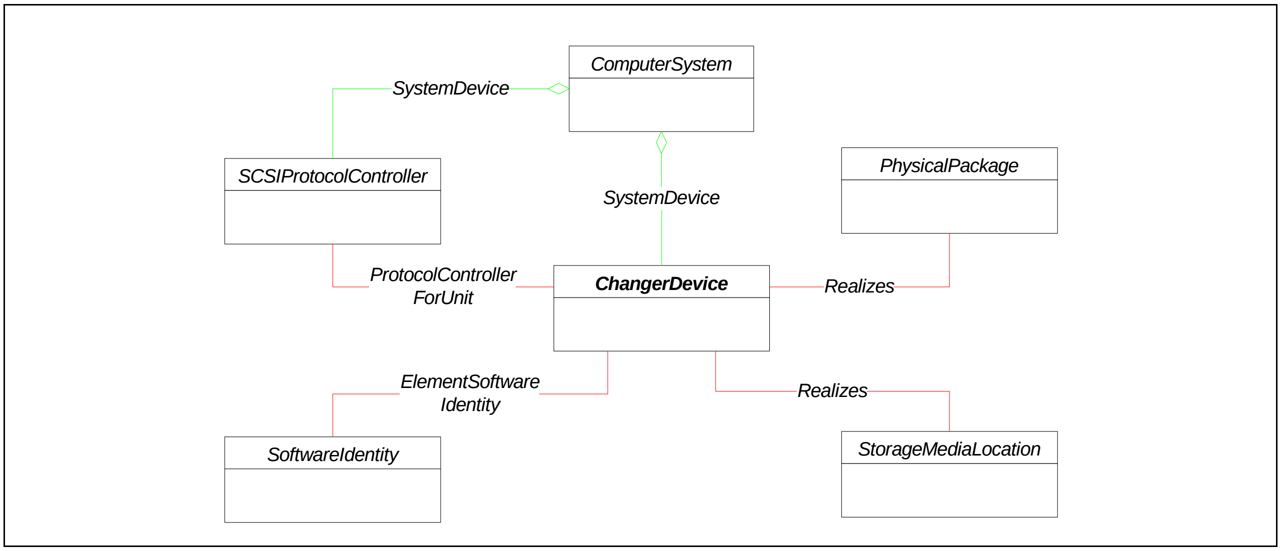


Figure 7 - ChangerDevice-centric Instance Diagram

4.1.5 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.

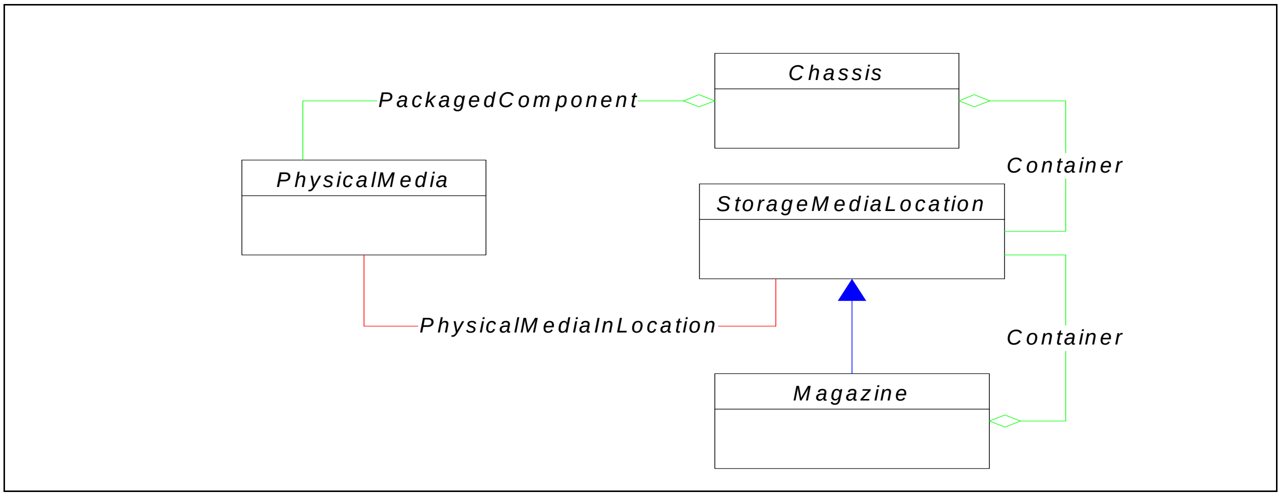


Figure 8 - Physical View Instance Diagram

4.1.6 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.

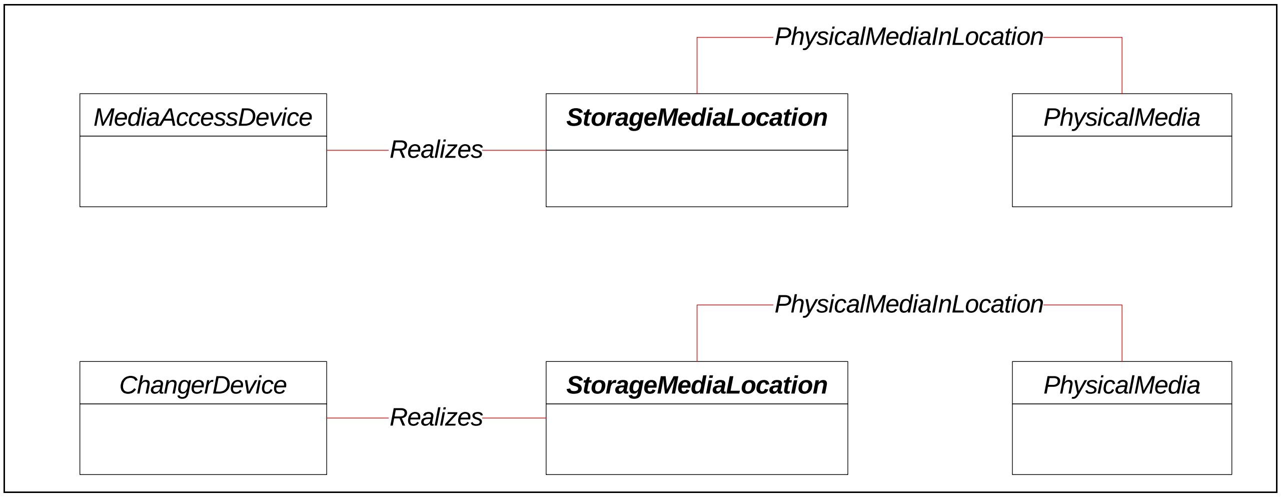


Figure 9 - StorageMediaLocation Instance Diagram

4.1.7 Durable Names and Correlatable IDs of the Profile

Different implementations use different approaches to uniquely identify the SCSI units pertinent to Storage Media Libraries (i.e., Changer Devices and Media Access Devices). The agent should utilize the same Durable Name techniques described for volumes in *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4*, 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 1 Common Architecture, 1.6.0 Rev 4* 7.6 "Standard Formats for Correlatable Names" for additional information.

4.1.8 Media Library Indications

4.1.8.1 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.

4.1.8.2 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.

4.1.8.3 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.

4.1.8.4 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.

4.1.8.5 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.

4.1.8.6 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

4.1.8.7 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.

4.1.8.8 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.

4.1.8.9 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.

4.1.8.10 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.

4.1.8.11 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

4.1.8.12 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

4.1.8.13 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.

4.1.8.14 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.

4.1.8.15 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.

4.1.8.16 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.

4.1.8.17 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.

4.1.8.18 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.

4.1.8.19 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.

4.1.8.20 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.

4.1.8.21 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.

4.1.8.22 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.

4.1.8.23 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.

4.1.8.24 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.

4.1.8.25 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.

4.1.8.26 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.

4.1.8.27 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.

4.1.8.28 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.

4.1.8.29 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.

4.1.8.30 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.

4.1.8.31 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.

4.1.8.32 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.

4.1.8.33 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.

4.1.8.34 Loader Hardware B Alert

Loader mechanism has a hardware fault.

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

4.1.8.35 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.

4.1.8.36 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.

4.1.8.37 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.

4.1.8.38 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.

4.1.8.39 Load Statistics Alert

Drive or library powered down with media loaded.

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

4.1.8.40 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.

4.1.8.41 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.

4.1.8.42 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.

4.1.8.43 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.

4.1.8.44 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.

4.1.8.45 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.

4.1.8.46 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.

4.1.8.47 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.

4.1.8.48 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.

4.1.8.49 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.

4.1.8.50 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.

4.1.8.51 Failure Prediction Alert

Predictive failure of library hardware

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

4.1.8.52 Library Maintenance Alert

Library preventative maintenance required.

Probable Cause:

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.

4.1.8.53 Library Humidity Limits

Library humidity limits exceeded

Probable Cause: Library humidity range is outside the operational conditions

4.1.8.54 Library Voltage Limits Alert

Library voltage limits exceeded

Probable Cause: Potential problem with a power supply.

4.1.8.55 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.

4.1.8.56 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.

4.1.8.57 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.

4.1.8.58 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.

4.1.8.59 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.

4.1.8.60 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

4.1.8.61 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.

4.1.8.62 Library Security Alert

Library door opened then closed during operation

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

4.1.8.63 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.

4.1.8.64 Library Offline Alert

Library manually turned offline

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

4.1.8.65 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.

4.1.8.66 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.

4.1.8.67 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.

4.1.8.68 Library Illegal Operation Alert

Illegal operation detected

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

4.1.8.69 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.

4.1.8.70 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.

4.1.8.71 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.

4.2 Health and Fault Management Considerations

None

4.3 Cascading Considerations

None

4.4 Supported Subprofiles and Packages

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 Media Movement	SNIA	1.1.0	Optional	Experimental.
Storage Library Capacity	SNIA	1.1.0	Optional	Experimental.

Table 1 - Supported Profiles for Storage Library

Profile Name	Organization	Version	Requirement	Description
Storage Library Element Counting	SNIA	1.1.0	Optional	Experimental.
Storage Library InterLibraryPort Connection	SNIA	1.1.0	Optional	Deprecated.
Physical Package	SNIA	1.5.0	Mandatory	
Launch In Context	DMTF	1.0.0	Optional	Experimental. See DSP1102, version 1.0.0
Library Views	SNIA	1.6.0	Optional	Experimental.
Indication	SNIA	1.5.0	Support for at least one is mandatory.	Deprecated. See the SNIA Indications Profile
Indications	SNIA	1.6.0		Experimental.
Indications	DMTF	1.2.0		Experimental. See DSP1054, version 1.2.0
FC Target Ports	SNIA	1.4.0	Support for at least one is mandatory.	
SPI Target Ports	SNIA	1.4.0		

4.5 Methods of this Profile

None

4.6 Client Considerations and Recipes

4.6.1 Recipe Overview

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

4.6.2 Discover a Storage Media Library

Discovery of Storage Media Libraries is achieved by looking up instances of ComputerSystem which are subclassed from System and have a corresponding Name and NameFormat property as described above under 4.1.7 "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

4.6.3 Determine Library Physical Media Capacity

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

In implementations that choose to include the Library Capacity Subprofile, minimum and maximum slot capacities for a Storage Library are modeled in the ConfigurationCapacity, which is described in Clause 7: "Library Capacity

Subprofile". Since this use case relies on an optional part of the profile, it may not be supported by each agent implementation.

4.6.4 Determine Physical Media Inventory

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

4.6.5 Discover Storage Library Control Type

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

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

If a library does not have a ProtocolController instance associated via ProtocolControllerForUnit to the ChangerDevice then the client should conclude that an alternate mechanism for controlling the library is required. This mechanism may vary, but should be represented by an instance of a HostedService associated with the ComputerSystem that models the storage library.

4.6.6 Determine Library Drive Capacity

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

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

4.6.7 Determine Drive Data Path Technology

Clients can discover the data path protocol of each drive within a storage library by enumerating MediaAccessDevice instances, then following the ProtocolControllerForUnit association linking a MediaAccessDevivce with a ProtocolController. Properties within Contoller can then be queried for more information. If the MediaAccessDevice has a fibre channel interface, an FCPort instance is linked to its ProtocolController by a ProtocolControllerForPort association. See *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* Clause 8: FC Target Ports Profile for more information on fibre channel connectivity.

4.6.8 Find asset Information

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

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

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

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

Note that some smaller libraries do not have the type of import/export element modeled by LimitedAccessPort. As a result, LimitedAccessPort elements are included in an (optional) subprofile (see Clause 8: Limited Access Port Elements Subprofile).

4.6.10 Counting assets in large storage libraries

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

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

4.7 Registered Name and Version

Storage Library version 1.6.0 (Autonomous Profile)

4.8 CIM Elements

Table 2 describes the CIM elements for Storage Library.

Table 2 - CIM Elements for Storage Library

Element Name	Requirement	Description
4.8.1 CIM_ChangerDevice	Mandatory	
4.8.2 CIM_Chassis	Mandatory	
4.8.3 CIM_ComputerSystem	Mandatory	'Top level' system that represents the whole Storage Library. Associated to RegisteredProfile.
4.8.4 CIM_ComputerSystemPackage	Mandatory	
4.8.5 CIM_ElementCapabilities	Optional	Class to implement the association between the top-level ComputerSystem representing a Storage Library and it's StorageLibraryCapabilities.

Table 2 - CIM Elements for Storage Library

Element Name	Requirement	Description
4.8.6 CIM_ElementSoftwareIdentity	Mandatory	
4.8.7 CIM_MediaAccessDevice	Mandatory	
4.8.8 CIM_PackagedComponent	Mandatory	
4.8.9 CIM_PhysicalMedia	Mandatory	
4.8.10 CIM_PhysicalMediaInLocation	Mandatory	
4.8.11 CIM_ProtocolControllerForUnit	Mandatory	
4.8.12 CIM_Realizes	Conditional	Conditional requirement: Support for Inter-Library Port profile.
4.8.13 CIM_SCSIProtocolController	Mandatory	
4.8.14 CIM_SoftwareIdentity	Mandatory	
4.8.15 CIM_StorageLibraryCapabilities	Optional	Experimental. Describes the capabilities of the Storage Library represented by the top level ComputerSystem this is associated with.
4.8.16 CIM_StorageMediaLocation	Mandatory	
4.8.17 CIM_SystemDevice (System to Changer Device)	Mandatory	This association links Changer to the scoping system.
4.8.18 CIM_SystemDevice (System to MediaAccessDevice)	Mandatory	This association links MediaAccessDevice To the scoping system.
4.8.19 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.

Table 2 - CIM Elements for Storage Library

Element Name	Requirement	Description
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.
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::OperationalStatus <> SourceInstance.CIM_ComputerSystem::OperationalStatus	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::OperationalStatus <> SourceInstance.CIM_ChangerDevice::OperationalStatus	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. See 4.1.8.1 Read Warning Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.1 Message: Read Warning.

Table 2 - CIM Elements for Storage Library

Element Name	Requirement	Description
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML2'	Optional	Experimental. The drive is having severe trouble writing. See 4.1.8.2 Write Warning Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.2 Message: Write Warning.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML3'	Optional	Experimental. The drive had a hard read or write error. See 4.1.8.3 Hard Error Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.3 Message: Hard 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. See 4.1.8.4 Media Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.4 Message: Media.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML5'	Optional	Experimental. The drive can no longer read data from the storage media. See 4.1.8.5 Read Failure Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.5 Message: Read Failure.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML6'	Optional	Experimental. The drive can no longer write data to the media. See 4.1.8.6 Write Failure Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.6 Message: Write Failure.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML7'	Optional	Experimental. The media has exceeded its specified life. See 4.1.8.7 Media Life Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.7 Message: Media 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. See 4.1.8.8 Not Data Grade Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.8 Message: Not Data Grade.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML9'	Optional	Experimental. Write command is attempted to a write protected media. See 4.1.8.9 Write Protect Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.9 Message: Write Protect.

Table 2 - CIM Elements for Storage Library

Element Name	Requirement	Description
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML10'	Optional	Experimental. Manual or software unload attempted when prevent media removal is on. See 4.1.8.10 No Removal Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.10 Message: No Removal.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML11'	Optional	Experimental. Cleaning media loaded into drive. See 4.1.8.11 Cleaning Media Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.11 Message: Cleaning Media.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML12'	Optional	Experimental. Attempted load of unsupported media format (e.g., DDS2 in DDS1 drive). See 4.1.8.12 Unsupported Format Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.12 Message: Unsupported Format.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML13'	Optional	Experimental. Tape snapped/cut in the drive where media can be de-mounted. See 4.1.8.13 Recoverable Snapped Tape Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.13 Message: Recoverable Snapped Tape.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML14'	Optional	Experimental. Tape snapped/cut in the drive where media cannot be de-mounted. See 4.1.8.14 Unrecoverable Snapped Tape Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.14 Message: Unrecoverable Snapped Tape.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML15'	Optional	Experimental. Memory chip failed in cartridge. See 4.1.8.15 Memory Chip In Cartridge Failure Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.15 Message: Memory Chip In Cartridge Failure.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML16'	Optional	Experimental. Manual or forced eject while drive actively writing or reading. See 4.1.8.16 Forced Eject Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.16 Message: Forced Eject.

Table 2 - CIM Elements for Storage Library

Element Name	Requirement	Description
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML17'	Optional	Experimental. Media loaded that is read-only format. See 4.1.8.17 Read Only Format Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.17 Message: 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. See 4.1.8.18 Directory Corrupted On Load Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.18 Message: Directory Corrupted On Load.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML19'	Optional	Experimental. Media may have exceeded its specified number of passes. See 4.1.8.19 Nearing Media Life Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.19 Message: Nearing Media Life.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML20'	Optional	Experimental. The drive thinks it has a head clog or needs cleaning. See 4.1.8.20 Clean Now Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.20 Message: Clean Now.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML21'	Optional	Experimental. The drive is ready for a periodic cleaning. See 4.1.8.21 Clean Periodic Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.21 Message: Clean Periodic.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML22'	Optional	Experimental. The cleaning media has expired. See 4.1.8.22 Expired Cleaning Media Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.22 Message: Expired Cleaning Media.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML23'	Optional	Experimental. Invalid cleaning media type used. See 4.1.8.23 Invalid Cleaning Media Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.23 Message: Invalid Cleaning Media.

Table 2 - CIM Elements for Storage Library

Element Name	Requirement	Description
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. See 4.1.8.24 Retention Requested Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.24 Message: Retention Requested.
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). See 4.1.8.25 Dual-Port Interface Error Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.25 Message: Dual-Port Interface Error.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='Core13'	Optional	Experimental. Fan failure inside drive mechanism or drive enclosure. See *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* .
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='Core14'	Optional	Experimental. Power supply unit failure inside the drive enclosure or rack subsystem. See *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* .
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='Core15'	Optional	Experimental. Power consumption of the drive is outside specified range. See *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* .
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML26'	Optional	Experimental. The drive requires preventive maintenance (not cleaning). See 4.1.8.26 Drive Maintenance Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.26 Message: Drive Maintenance.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML27'	Optional	Experimental. The drive has a hardware fault that requires reset to recover. See 4.1.8.27 Hardware A Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.27 Message: Hardware A.
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. See 4.1.8.28 Hardware B Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.28 Message: Hardware B.

Table 2 - CIM Elements for Storage Library

Element Name	Requirement	Description
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML29'	Optional	Experimental. The drive has identified an interface fault. See 4.1.8.29 Interface Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.29 Message: Interface.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML30'	Optional	Experimental. Error recovery action: Media Ejected. See 4.1.8.30 Eject Media Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.30 Message: Eject Media.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML31'	Optional	Experimental. Firmware download failed. See 4.1.8.31 Download Failure Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.31 Message: Download Failure.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='Core2'	Optional	Experimental. Drive humidity or temperature limits exceeded. See *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* .
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='Core17'	Optional	Experimental. Drive voltage limits exceeded. See *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* .
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='Core18'	Optional	Experimental. Predictive failure of drive hardware. See *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* .
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. See *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* .
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML32'	Optional	Experimental. Loader mechanism is having trouble communicating with the drive. See 4.1.8.32 Loader Hardware A Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.32 Message: Loader Hardware A.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML33'	Optional	Experimental. Stray media left in loader after previous error recovery. See 4.1.8.33 Loader Stray Media Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.33 Message: Loader Stray Media.

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. See 4.1.8.34 Loader Hardware B Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.34 Message: Loader Hardware B.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML35'	Optional	Experimental. Changer door open. See 4.1.8.35 Loader Door Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.35 Message: Loader Door.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML36'	Optional	Experimental. The loader mechanism has a hardware fault that is not mechanically related. See 4.1.8.36 Loader Hardware C Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.36 Message: Loader Hardware C.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML37'	Optional	Experimental. Loader magazine not present. See 4.1.8.37 Loader Magazine Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.37 Message: Loader Magazine.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML38'	Optional	Experimental. Predictive failure of loader mechanism hardware. See 4.1.8.38 Loader Predictive Failure Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.38 Message: Loader Predictive Failure.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML39'	Optional	Experimental. Drive or library powered down with media loaded. See Probable Cause: The autoloader cannot operate without the magazine: and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.39 Message: Load Statistics.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML40'	Optional	Experimental. Error preventing the media directory being updated on unload. See 4.1.8.40 Media Directory Invalid at Unload Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.40 Message: Media Directory Invalid at Unload.

Table 2 - CIM Elements for Storage Library

Element Name	Requirement	Description
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML41'	Optional	Experimental. Write errors while writing the system area on unload. See 4.1.8.41 Media System area Write Failure Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.41 Message: Media System area Write Failure.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML42'	Optional	Experimental. Read errors while reading the system area on load. See 4.1.8.42 Media System Area Read Failure Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.42 Message: Media System Area Read Failure.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML43'	Optional	Experimental. Media damaged, bulk erased, or incorrect format. See 4.1.8.43 No Start of Data Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.43 Message: No Start of Data.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML44'	Optional	Experimental. The drive is unable to load the media. See 4.1.8.44 Loading Failure Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.44 Message: Loading Failure.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML45'	Optional	Experimental. Changer mechanism is having trouble communicating with the internal drive. See 4.1.8.45 Library Hardware A Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.45 Message: Library Hardware A.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML46'	Optional	Experimental. Changer mechanism has a hardware fault. See 4.1.8.46 Library Hardware B Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.46 Message: Library Hardware B.
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. See 4.1.8.47 Library Hardware C Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.47 Message: Library Hardware C.

Table 2 - CIM Elements for Storage Library

Element Name	Requirement	Description
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. See 4.1.8.48 Library Hardware D Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.48 Message: Library Hardware D.
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. See 4.1.8.49 Library Diagnostic Required Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.49 Message: Library Diagnostic Required.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML50'	Optional	Experimental. The library has identified an interface fault. See 4.1.8.50 Library Interface Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.50 Message: Library Interface.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML51'	Optional	Experimental. Predictive failure of library hardware. See 4.1.8.51 Failure Prediction Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.51 Message: Failure Prediction.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML52'	Optional	Experimental. Library preventative maintenance required. See 4.1.8.52 Library Maintenance Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.52 Message: Library Maintenance.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML53'	Optional	Experimental. Library humidity limits exceeded. See 4.1.8.53 Library Humidity Limits and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.53 Message: Library Humidity Limits.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='Core2'	Optional	Experimental. Library temperature limits exceeded. See *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* .

Table 2 - CIM Elements for Storage Library

Element Name	Requirement	Description
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML54'	Optional	Experimental. Library voltage limits exceeded. See 4.1.8.54 Library Voltage Limits Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.54 Message: Library Voltage Limits.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML55'	Optional	Experimental. Stray cartridge left in library after previous error recovery. See 4.1.8.55 Library Stray Media Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.55 Message: Library Stray Media.
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. See 4.1.8.56 Library Pick Retry Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.56 Message: Library Pick Retry.
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. See 4.1.8.57 Library Place Retry Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.57 Message: Library Place Retry.
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. See 4.1.8.58 Library Load Retry Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.58 Message: Library Load Retry.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML59'	Optional	Experimental. Library door open is preventing the library from functioning. See 4.1.8.59 Library Door Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.59 Message: Library Door.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML60'	Optional	Experimental. Mechanical problem with import/export mailslot. See 4.1.8.60 Library Mailslot Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.60 Message: Library Mailslot.

Table 2 - CIM Elements for Storage Library

Element Name	Requirement	Description
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML61'	Optional	Experimental. Library magazine not present. See 4.1.8.61 Library Magazine Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.61 Message: Library Magazine.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML62'	Optional	Experimental. Library door opened then closed during operation. See 4.1.8.62 Library Security Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.62 Message: Library Security.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML63'	Optional	Experimental. Library security mode changed. See 4.1.8.63 Library Security Mode Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.63 Message: Library Security Mode.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML64'	Optional	Experimental. Library manually turned offline. See 4.1.8.64 Library Offline Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.64 Message: Library Offline.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML65'	Optional	Experimental. Library turned internal drive offline. See 4.1.8.65 Library Drive Offline Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.65 Message: Library 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. See 4.1.8.66 Library Scan Retry Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.66 Message: Library Scan Retry.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML67'	Optional	Experimental. Inconsistent media inventory. See 4.1.8.67 Library Inventory Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.67 Message: Library Inventory.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML68'	Optional	Experimental. Illegal operation detected. See 4.1.8.68 Library Illegal Operation Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.68 Message: Library Illegal Operation.

Table 2 - CIM Elements for Storage Library

Element Name	Requirement	Description
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='Core1'	Optional	Experimental. Failure of one interface port in a dual-port configuration. See *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* .
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. See 4.1.8.69 Pass Through Mechanism Failure Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.69 Message: Pass Through Mechanism Failure.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='SML70'	Optional	Experimental. Cartridge left in the pass-through mechanism between two library modules. See 4.1.8.70 Cartridge in Pass-through Mechanism Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.70 Message: Cartridge in Pass-through Mechanism.
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. See 4.1.8.71 Unreadable barcode Labels Alert and *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.71 Message: Unreadable barcode Labels.

4.8.1 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	

Table 3 - SMI Referenced Properties/Methods for CIM_ChangerDevice

Properties	Flags	Requirement	Description & Notes
DeviceID		Mandatory	
MediaFlipSupported		Mandatory	
ElementName		Mandatory	
OperationalStatus		Mandatory	Status of the changer device.
StatusDescriptions		Optional	Additional information related to the values in OperationalStatus.

4.8.2 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	

4.8.3 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.
PrimaryOwnerContact	M	Optional	Contact details for storage library owner.
PrimaryOwnerName	M	Optional	Owner of the storage library.

4.8.4 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	

4.8.5 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	

4.8.6 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.

4.8.7 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	

4.8.8 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	

4.8.9 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	

4.8.10 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	

4.8.11 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.

4.8.12 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	

4.8.13 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	

Table 15 - SMI Referenced Properties/Methods for CIM_SCSIProtocolController

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

4.8.14 CIM_SoftwareIdentity

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

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 of firmware version of the device (ChangerDevice, MediaAccessDevice, or a SCSIProtocolController).
Manufacturer		Mandatory	
Classifications		Optional	4 = Application Software, 10 = Firmware.
BuildNumber		Optional	
MajorVersion		Optional	
RevisionNumber		Optional	
MinorVersion		Optional	

4.8.15 CIM_StorageLibraryCapabilities

Experimental.

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.

4.8.16 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	

4.8.17 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	

4.8.18 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 MediaAccess-Device)

Properties	Flags	Requirement	Description & Notes
PartComponent		Mandatory	
GroupComponent		Mandatory	

4.8.19 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 SCSIProtocol-Controller)

Properties	Flags	Requirement	Description & Notes
PartComponent		Mandatory	
GroupComponent		Mandatory	

STABLE

EXPERIMENTAL

Clause 5: Element Counting Subprofile

5.1 Description

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

Figure 10 provides a sample instance diagram.

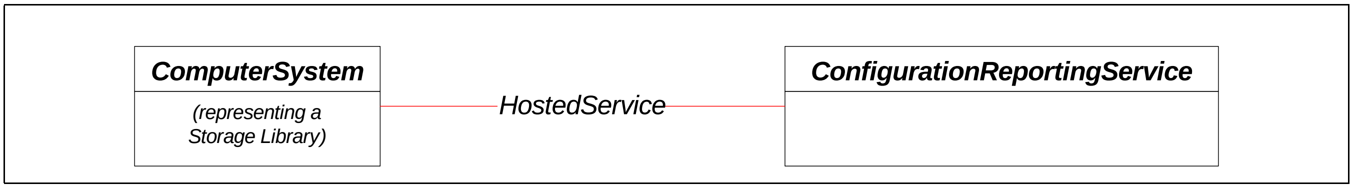

ComputerSystem
(representing a Storage Library)
HostedService
ConfigurationReportingService

Figure 10 - Instance Diagram

5.1.1 Discovery

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

5.2 Health and Fault Management Considerations

Not defined in this standard.

5.3 Cascading Considerations

Not defined in this standard.

5.4 Supported Subprofiles and Packages

Related Profiles for Storage Library Element Counting: Not defined in this standard.

5.5 Methods of the Profile

5.5.1 GetClassTypes

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

The GetClassTypes method uses the following parameters:

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

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

[IN] boolean Recursive = true or false

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

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

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

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

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

5.5.2 GetUnitTypes

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

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

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

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

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

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

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

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

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

5.5.3 ReportCapacity

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

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

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

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

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

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

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

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

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

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

5.6 Client Considerations and Recipes

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

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

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

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

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

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

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

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

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

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

5.7 Registered Name and Version

Storage Library Element Counting version 1.1.0 (Component Profile)

5.8 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
5.8.1 CIM_ConfigurationReportingService	Mandatory	
5.8.2 CIM_HostedService	Mandatory	

5.8.1 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	

5.8.2 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

DEPRECATED

Clause 6: InterLibraryPort Connection Subprofile

Note: The InterLibraryPort Connection Subprofile is scheduled for removal for SMI-S 2.0. The functionality of this profile will not be replaced in SMI-S 2.0. The SNIA would like to hear from anyone that has implemented the InterLibraryPort Connection Subprofile. If your company or organization has implemented this profile and is a member of the SNIA, please contact the SMI-S Medial Library Technical Working Group or indicate your preference to keep this profile in SMI-S 2.0 during member reviews and ballots. If your company or organization has implemented this profile and is not a member of the SNIA, please indicate your preference to keep this profile as part of SMI-S using the SNIA feedback portal: http://www.snia.org/tech_activities/feedback/ .

6.1 Description

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

Figure 11 provides a sample instance diagram.

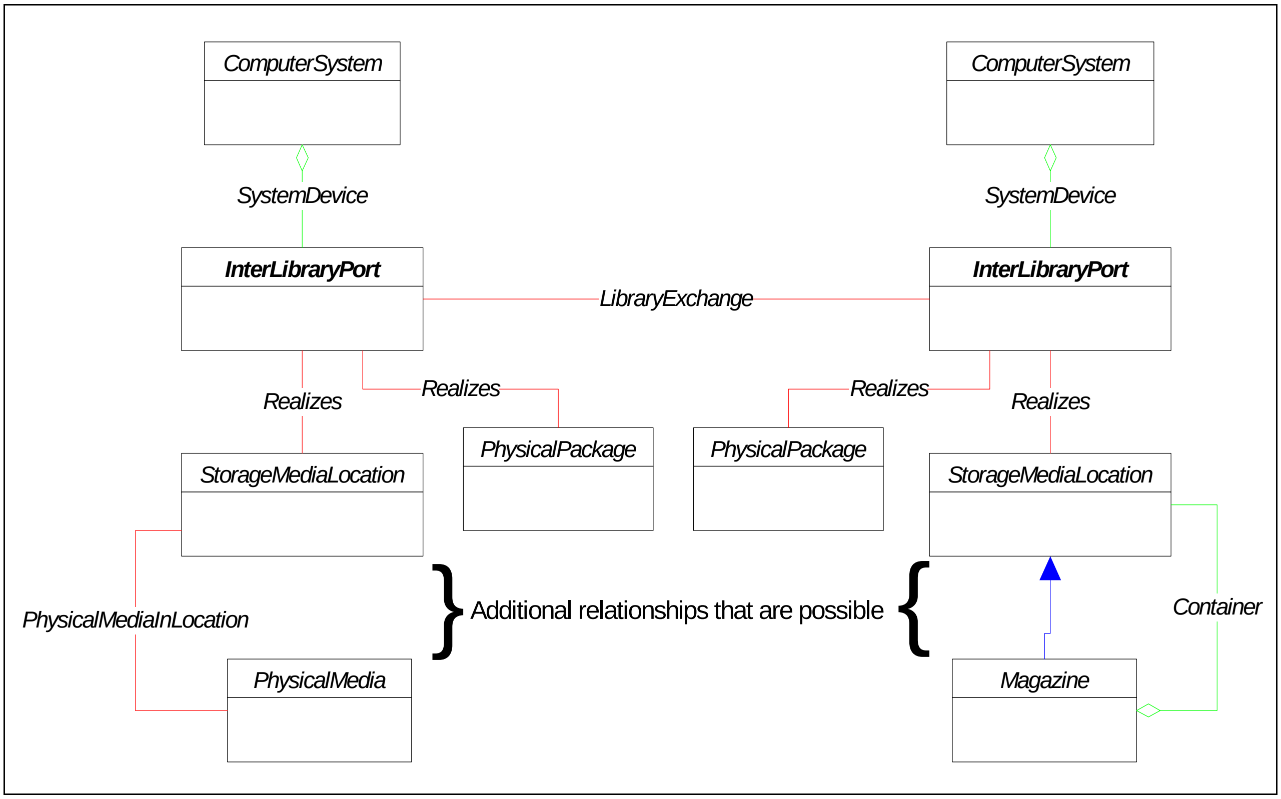


Figure 11 - InterLibraryPort Connection Instance Diagram

Durable Names and Correlatable IDs

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

6.2 Health and Fault Management Considerations

Not defined in this standard.

6.3 Cascading Considerations

Not defined in this standard.

6.4 Supported Subprofiles and Packages

None.

6.5 Methods of the Profile

None.

6.6 Client Considerations and Recipes

None.

6.7 Registered Name and Version

Storage Library InterLibraryPort Connection version 1.1.0 (Component Profile)

6.8 CIM Elements

Table 25 describes the CIM elements for Storage Library InterLibraryPort Connection.

Table 25 - CIM Elements for Storage Library InterLibraryPort Connection

Element Name	Requirement	Description
6.8.1 CIM_InterLibraryPort	Mandatory	InterLibraryPorts represent hardware that transports Physical Media between connected Storage Libraries. The LibraryExchange association identifies the connected Libraries, by identifying the connected InterLibraryPorts.
6.8.2 CIM_LibraryExchange	Mandatory	This relationship identifies that two storage libraries are connected through their InterLibraryPorts.
SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_InterLibraryPort	Mandatory	Creation of an instance of InterLibraryPort.
SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_InterLibraryPort	Mandatory	Deletion of an instance of InterLibraryPort.

Table 25 - CIM Elements for Storage Library InterLibraryPort Connection

Element Name	Requirement	Description
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_InterLibraryPort AND SourceInstance.OperationalStatus <> PreviousInstance.OperationalStatus	Mandatory	Deprecated WQL -Change in OperationalStatus of a InterLibraryPort.
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_InterLibraryPort AND SourceInstance.CIM_InterLibraryPort::OperationalStatus <> PreviousInstance.CIM_InterLibraryPort::OperationalStatus	Mandatory	CQL -Change in OperationalStatus of a InterLibraryPort.

6.8.1 CIM_InterLibraryPort

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 26 describes class CIM_InterLibraryPort.

Table 26 - SMI Referenced Properties/Methods for CIM_InterLibraryPort

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

6.8.2 CIM_LibraryExchange

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

Table 27 describes class CIM_LibraryExchange.

Table 27 - SMI Referenced Properties/Methods for CIM_LibraryExchange

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	
Dependent		Mandatory	

DEPRECATED

EXPERIMENTAL

Clause 7: Library Capacity Subprofile

7.1 Description

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

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

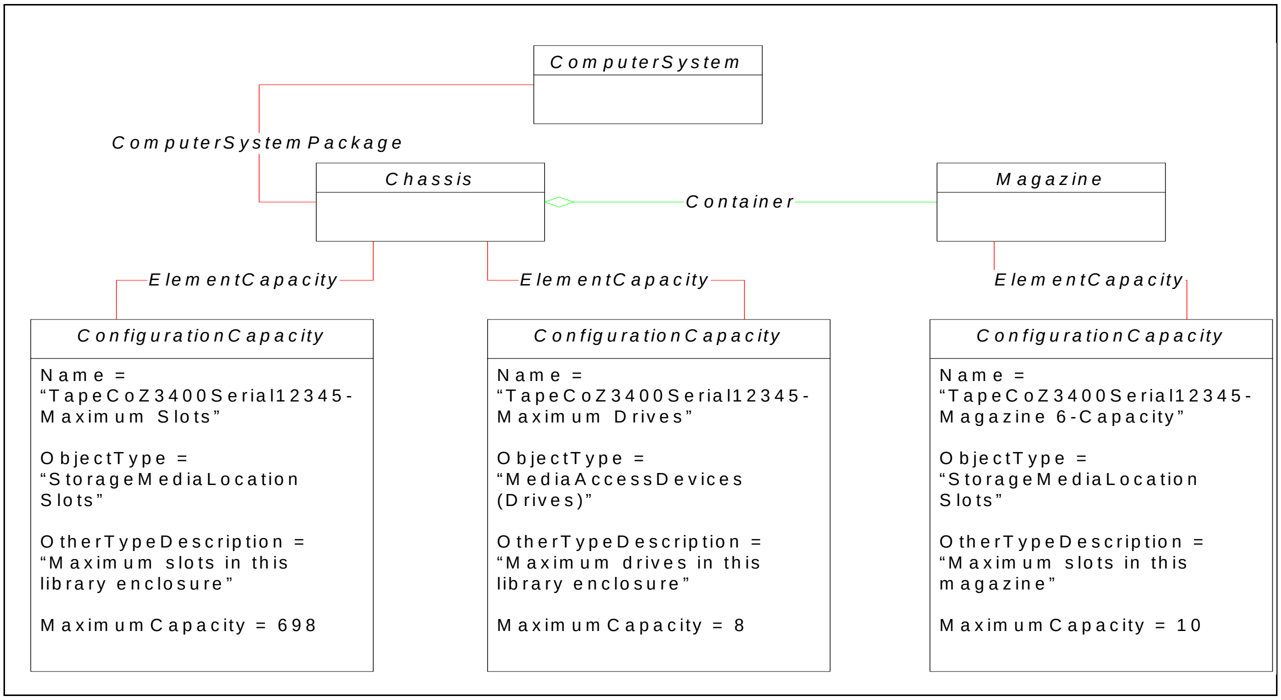

ComputerSystem
ComputerSystemPackage
Chassis
Container
Magazine
ElementCapacity
ElementCapacity
ElementCapacity
ConfigurationCapacity
Name = “TapeCoZ3400Serial12345-Maximum Slots”
ObjectType = “StorageMediaLocation Slots”
OtherTypeDescription = “Maximum slots in this library enclosure”
MaximumCapacity = 698
ConfigurationCapacity
Name = “TapeCoZ3400Serial12345-Maximum Drives”
ObjectType = “MediaAccessDevices (Drives)”
OtherTypeDescription = “Maximum drives in this library enclosure”
MaximumCapacity = 8
ConfigurationCapacity
Name = “TapeCoZ3400Serial12345-Magazine 6-Capacity”
ObjectType = “StorageMediaLocation Slots”
OtherTypeDescription = “Maximum slots in this magazine”
MaximumCapacity = 10

Figure 12 - Library Capacity Instance Diagram

7.2 Health and Fault Management Considerations

Not defined in this standard.

7.3 Cascading Considerations

Not defined in this standard.

7.4 Supported Subprofiles and Packages

None.

7.5 Client Considerations and Recipes

None.

7.6 Registered Name and Version

Storage Library Capacity version 1.1.0 (Component Profile)

7.7 CIM Elements

Table 28 describes the CIM elements for Storage Library Capacity.

Table 28 - CIM Elements for Storage Library Capacity

Element Name	Requirement	Description
7.7.1 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.
7.7.2 CIM_ElementCapacity	Mandatory	

7.7.1 CIM_ConfigurationCapacity

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

Table 29 describes class CIM_ConfigurationCapacity.

Table 29 - SMI Referenced Properties/Methods for CIM_ConfigurationCapacity

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

7.7.2 CIM_ElementCapacity

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

Table 30 describes class CIM_ElementCapacity.

Table 30 - SMI Referenced Properties/Methods for CIM_ElementCapacity

Properties	Flags	Requirement	Description & Notes
Element		Mandatory	
Capacity		Mandatory	

EXPERIMENTAL

STABLE

Clause 8: Limited Access Port Elements Subprofile

8.1 Description

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

8.1.1 Instance Diagram

Figure 13 and Figure 14 show the relationship between LimitedAccessPorts and other portions of the Storage Library Profile.

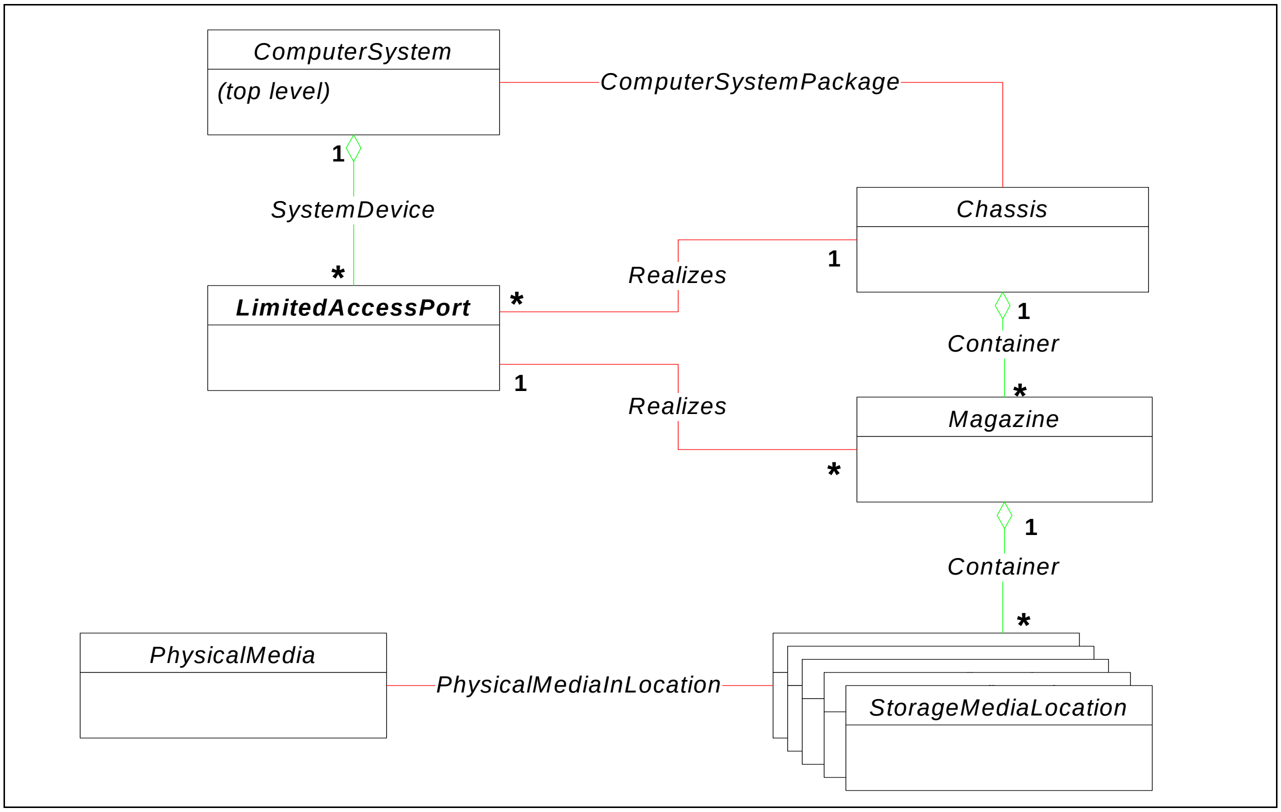


Figure 13 - Tape Libraries with Magazines in LimitedAccessPorts

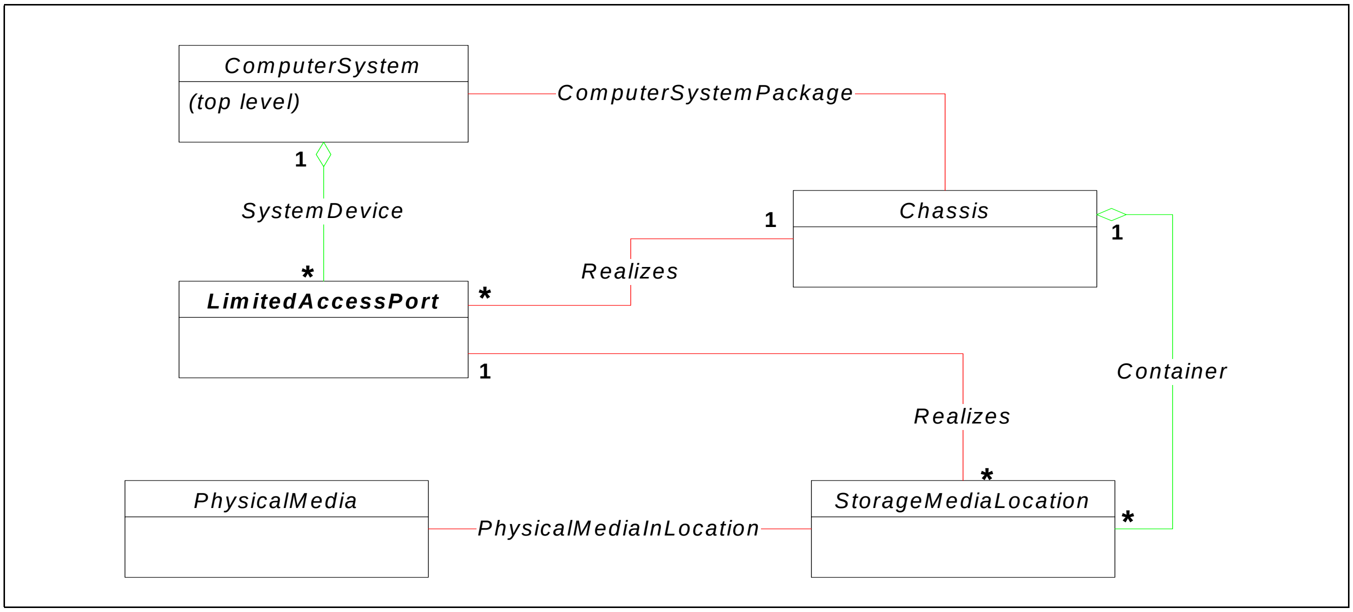


Figure 14 - Tape Libraries with no Magazines in LimitedAccessPorts

8.2 Health and Fault Management Considerations

Not defined in this standard.

8.3 Cascading Considerations

Not defined in this standard.

8.4 Supported Subprofiles and Packages

None.

8.5 Methods of the Profile

None.

8.5.1 Client Considerations and Recipes

None

8.6 Registered Name and Version

Storage Library Limited Access Port Elements version 1.2.0 (Component Profile)

8.7 CIM Elements

Table 31 describes the CIM elements for Storage Library Limited Access Port Elements.

Table 31 - CIM Elements for Storage Library Limited Access Port Elements

Element Name	Requirement	Description
8.7.1 CIM_Container	Mandatory	The containment relationship of Magazines within a Chassis or StorageMediaLocations within a Magazine.
8.7.2 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.
8.7.3 CIM_Magazine	Mandatory	
8.7.4 CIM_Realizes	Mandatory	The relationship between a LimitedAccessPort and the StorageMediaLocations, Magazines or Chassis to which it has access.
8.7.5 CIM_SystemDevice	Mandatory	The relationship between a LimitedAccessPort and its hosting top-level ComputerSystem which represents the Storage Library.
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.

8.7.1 CIM_Container

Created By: Static
Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 32 describes class CIM_Container.

Table 32 - SMI Referenced Properties/Methods for CIM_Container

Properties	Flags	Requirement	Description & Notes
GroupComponent		Mandatory	
PartComponent		Mandatory	

8.7.2 CIM_LimitedAccessPort

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 33 describes class CIM_LimitedAccessPort.

Table 33 - SMI Referenced Properties/Methods for CIM_LimitedAccessPort

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

8.7.3 CIM_Magazine

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 34 describes class CIM_Magazine.

Table 34 - 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	

8.7.4 CIM_Realizes

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

Table 35 describes class CIM_Realizes.

Table 35 - SMI Referenced Properties/Methods for CIM_Realizes

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	
Dependent		Mandatory	

8.7.5 CIM_SystemDevice

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

Table 36 describes class CIM_SystemDevice.

Table 36 - SMI Referenced Properties/Methods for CIM_SystemDevice

Properties	Flags	Requirement	Description & Notes
GroupComponent		Mandatory	
PartComponent		Mandatory	

STABLE

EXPERIMENTAL

Clause 9: Media Movement Subprofile

9.1 Description

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

Figure 15 illustrates the subprofile from the library perspective.

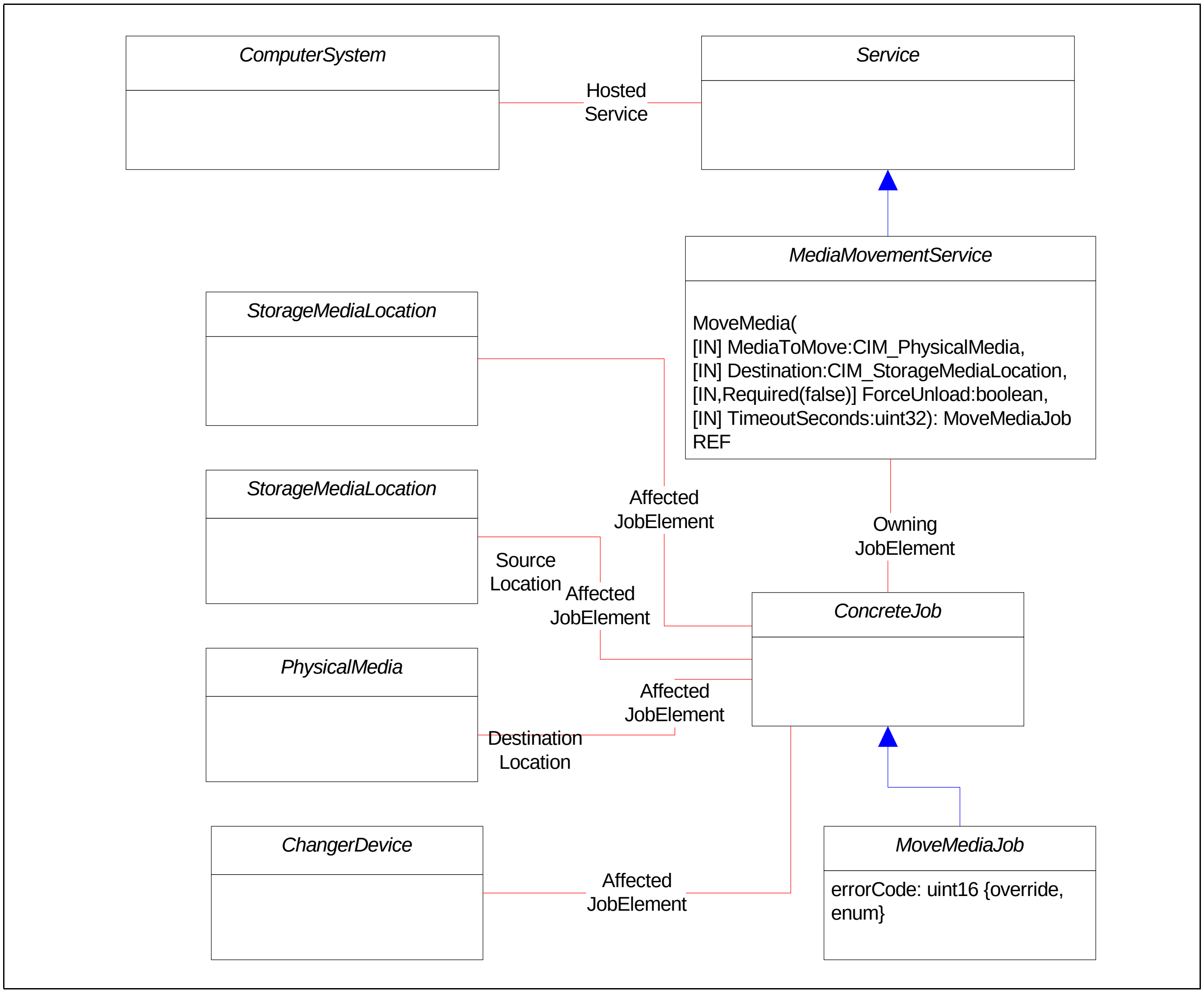


Figure 15 - Storage Library Centric View

When the move media operation is performed, the storage library shall physically move the medium, and then update the storage library's CIM object model. In particular, the StorageMediaInLocation association between the

PhyscialMedia instance and the source StorageMediaLocation instance shall be removed and a new association made between the PhysicalMedia instance and the destination StorageMediaLocation. This is illustrated in Figure 16.

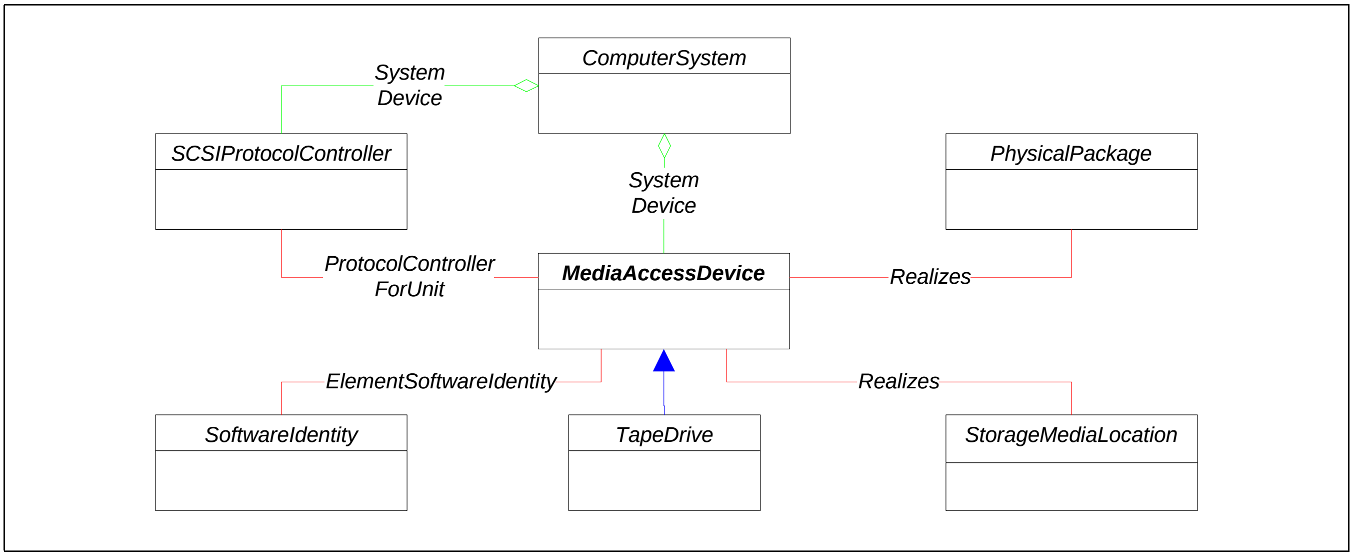


Figure 16 - Media-centrc View

9.2 Health and Fault Management Considerations

9.2.1 NULL Instance Handling

If a non-null instance of ConcreteJob is returned by the MoveMedia method, the implementation shall report errors which occur during the execution of the job through the ConcreteJob.GetError() method. See *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* Clause 9: Media Movement Subprofile for details.

9.2.2 Media Movement Subprofile Standard Messages

The standard messages specific to this profile are listed in Table 37.

Table 37 - Media Movement Standard Messages

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

Table 37 - Media Movement Standard Messages (Continued)

Message ID	Message Name
9	Internal Model Error
10	Command Sequence Error

9.3 Cascading Considerations

Not defined in this standard.

9.4 Supported Subprofiles and Packages

None.

9.5 Methods of the Profile

9.5.1 Moving a piece of PhysicalMedia

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

Error returns are:

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

The MoveMedia method takes as input references to the media to be moved, the destination location, and a timeout value. The method attempts to initiate a process on the Storage Library which will perform the media movement. If the process is successfully initiated, the MoveMedia returns a ConcreteJob object and an integer return code indicating the status of the job creation. If a non-null instance of ConcreteJob is returned, the instance shall be associated with an instance of MethodResult as specified by the Job Control Subprofile. See *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* Clause 26: Job Control Subprofile for details of job creation and execution.

9.5.1.1 Timeout parameter

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

9.5.1.2 ForceUnload parameter

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

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

9.6 Client Considerations and Recipes

9.6.1 Concurrent library access by SMI clients and other applications.

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

9.6.2 Use of the ForceUnload parameter

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

9.6.3 Job Lifecycle Indications

SMI Servers implementing the Job Control Profile are required to support a set of indications which indicate transitions in the operational status of the job. In particular, an indication shall be provided when a job stops, either successfully or with an error condition. The server may also generate indications for change in job status or percent complete. See 26.8 "CIM Elements" in Clause 26: Job Control Subprofile of the *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* for indication subscription details.

9.7 Registered Name and Version

Storage Library Media Movement version 1.1.0 (Component Profile)

9.8 CIM Elements

Table 38 describes the CIM elements for Storage Library Media Movement.

Table 38 - CIM Elements for Storage Library Media Movement

Element Name	Requirement	Description
9.8.1 CIM_HostedService	Mandatory	The relationship between the top-level ComputerSystem representing the Storage Library and the MediaMovementService.
9.8.2 SNIA_MediaMovementService	Mandatory	

9.8.1 CIM_HostedService

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

Table 39 describes class CIM_HostedService.

Table 39 - SMI Referenced Properties/Methods for CIM_HostedService

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	
Dependent		Mandatory	

9.8.2 SNIA_MediaMovementService

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

Table 40 describes class SNIA_MediaMovementService.

Table 40 - SMI Referenced Properties/Methods for SNIA_MediaMovementService

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

EXPERIMENTAL

EXPERIMENTAL

Clause 10: Partitioned Tape Library Profile

10.1 Description

10.1.1 Overview

This profile describes the model for a Partitioned Tape Library (PTL). Partitioning allows an organization to share a physical tape library across multiple clients with disparate departmental needs. Using a single physical infrastructure, it permits individual departments to preserve their own operating environment and security policies. For instance, instead of buying three libraries with 100 slots each and individual service agreements, a single 300-slot library can be purchased cutting down significantly on the total cost of ownership (TCO).

10.1.2 PTL Model

Figure 17 illustrates the major components of the PTL model. The Partitioned Tape Library System component is the central component that is responsible for configuring and managing all the partitions of the library. The ComputerSystem class instance contains the Dedicated property value of “Partitioned Library System” and acts as the top-level ComputerSystem instance for the rest of the components.

Each partition of the library is represented by the left-most column on elements with its ComputerSystem Dedicated property containing a value of “Partition”. These elements are modeled along the same lines as the physical Storage Library Profile.

The middle set of elements represent the unallocated set of resources (tape drives, physical tapes, slots, changer, etc.) and modeled as a partition with the ComputerSystem Dedicated property containing a value of “Unallocated Partition”.

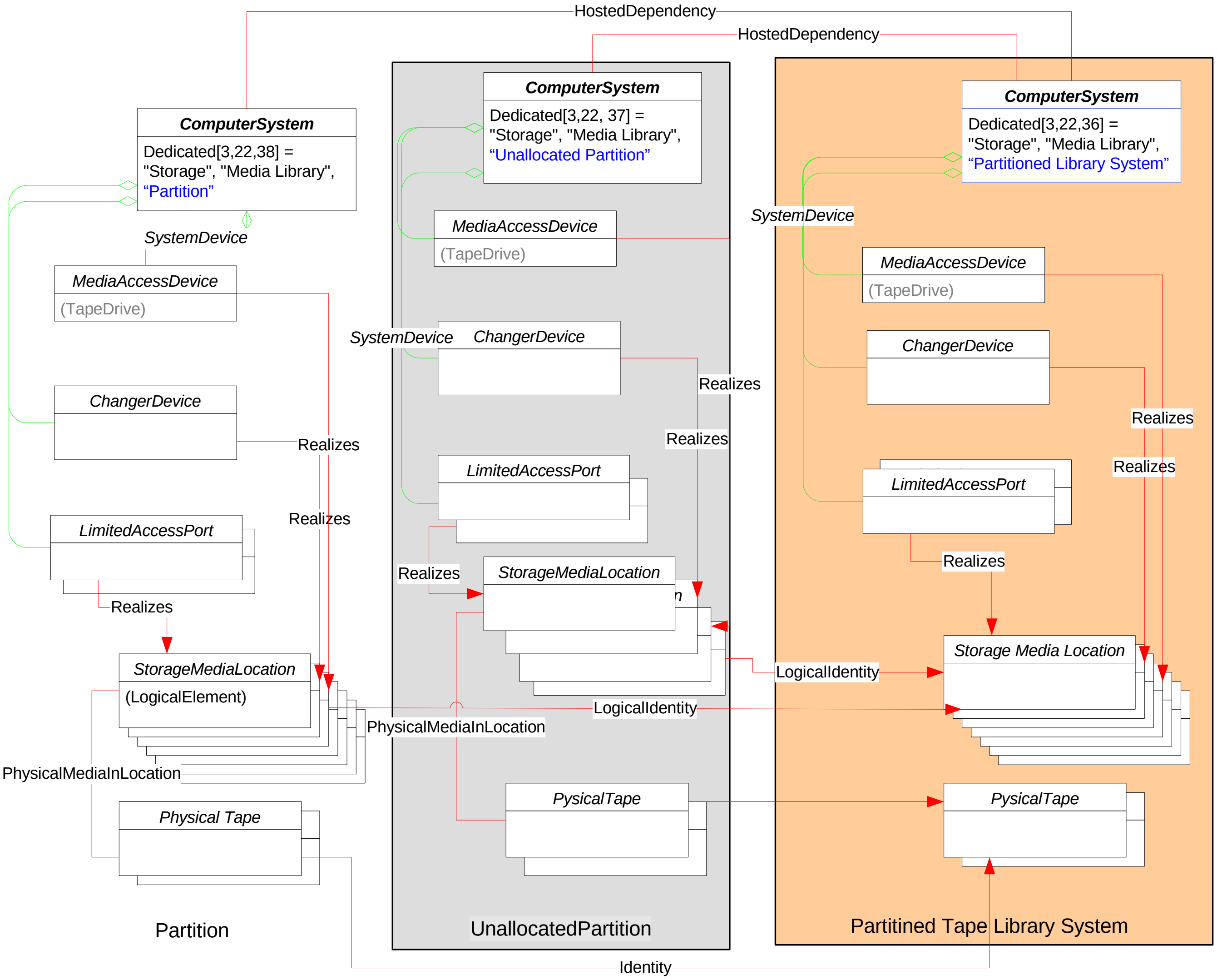


Figure 17 - Partitioned Tape Library System Model

10.1.3 PTL Configuration

Figure 18 shows the model related to the management of partitions in a PTL system.

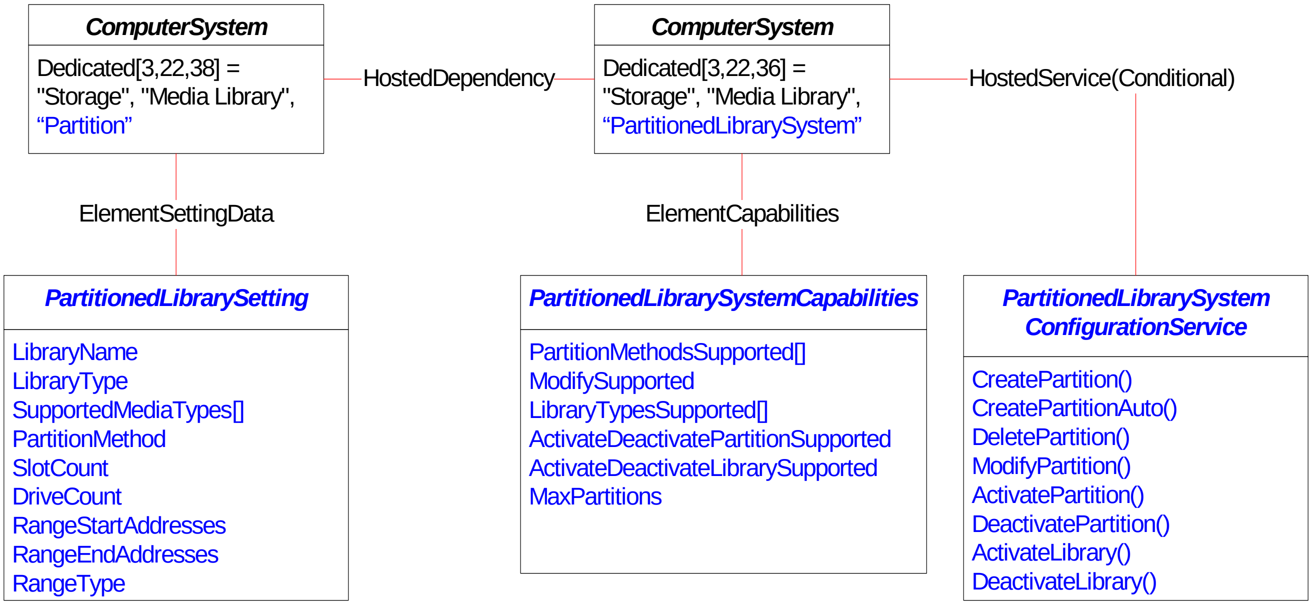


Figure 18 - Partitioned Tape Library Configuration Model

The PartitionedLibrarySystemCapabilities class contains properties that define the capabilities of the PTL system where as the PartitionedLibrarySystemConfigurationService class defines methods to create and manage the partitions in the PTL system.

Depending on the PTL system, there are variations on how you create a partition as listed in the PartitionMethodsSupported property in the capabilities class. The value map for this property defines the possible enumerations.

In **Auto** partitioning, given number of partitions N, the PTL system can create N partitions. How the library chooses to divide the available resources depends on the specific library. One way is to divide the resources equally.

In **Simple** partitioning, a partition is created by providing the number of slots and drives the partition should contain.

In **Slot** partitioning, a set of ranges of Slots (Storage Media Locations) are specified for the partition.

In **VolSer** partitioning, a partition can be created by assigning a set of ranges of cartridge serial numbers that belong to the partition. You also assign drives to the partition by specifying the storage media locations for the drives.

The parameters for all the above variations of the creation method are specified using the PartitionedLibrarySetting class instance.

A PTL system may not support any of the create methods at all in a monitor-only implementation.

10.1.4 PTL Configuration Methods

CreatePartition(PartitionSetting) creates a single partition using one of the variations defined above except for Auto partitioning.

PartitionSetting is a reference to a PartitionedLibrarySetting instance that contain the create parameter values including the PartitionMethod to be used.

CreatePartitionAuto(NumberOfPartitions) creates a number of partitions as given. How the resources are divided among the partitions is implementation-dependent.

ModifyPartition(PartitionSetting) takes a new setting instance as target and modifies the partition definition.

DeletePartition(ThePartition) deletes the referenced partition.

ActivatePartition(ThePartition) Once a partition is created or modified, the referenced partition can be brought online using this method.

DeactivatePartition(ThePartition) Before you delete or modify a partition, you need to bring the partition offline by using this method.

ActivateLibrary() In some libraries, you first create all the partitions and then activate all the partitions at once. This method activates all the partitions. The ActivateDeactivateLibrarySupported property in the capabilities class defines if a particular library has this requirement.

DeactivateLibrary() Similar to the activate, this method deactivates all the partitions so that some or all the partitions can be modified or deleted.

10.2 Health and Fault Management Consideration

Not defined in this standard.

10.3 Cascading Considerations

Not defined in this standard.

10.4 Supported Profiles, Subprofiles, and Packages

Table 41 describes the supported profiles for Partitioned Tape Library.

Table 41 - Supported Profiles for Partitioned Tape Library

Profile Name	Organization	Version	Requirement	Description
Access Points	SNIA	1.3.0	Optional	
Health	SNIA	1.2.0	Mandatory	
Software	SNIA	1.4.0	Optional	
Storage Library	SNIA	1.6.0	Optional	
Indication	SNIA	1.5.0	Mandatory	
Multiple Computer System	SNIA	1.2.0	Optional	
Masking and Mapping	SNIA	1.6.0	Optional	
Storage Library Element Counting	SNIA	1.1.0	Optional	
Storage Library Capacity	SNIA	1.1.0	Optional	

Table 41 - Supported Profiles for Partitioned Tape Library

Profile Name	Organization	Version	Requirement	Description
Storage Library Limited Access Port Elements	SNIA	1.2.0	Optional	
Storage Library Media Movement	SNIA	1.1.0	Optional	
Location	SNIA	1.4.0	Optional	
FC Target Ports	SNIA	1.4.0	Support for at least one is mandatory.	
SAS Target Ports	SNIA	1.4.0		
SPI Target Ports	SNIA	1.4.0		

10.5 Client Considerations and Recipes

Not defined in this profile.

10.6 Registered Name and Version

Partitioned Tape Library version 1.4.0 (Autonomous Profile)

10.7 CIM Elements

Table 42 describes the CIM elements for Partitioned Tape Library.

Table 42 - CIM Elements for Partitioned Tape Library

Element Name	Requirement	Description
10.7.1 CIM_ChangerDevice	Optional	The media changer for a PTL system (this is the physical pool).
10.7.2 CIM_Chassis (PTL System)	Optional	The box for a PTL.
10.7.3 CIM_ComputerSystemPackage (PTL System to Chassis)	Mandatory	This association links Chassis to the scoping system.
10.7.4 CIM_ConcreteIdentity (Slots to Slots)	Mandatory	This association links ports to the slots.
10.7.5 CIM_Container (Chassis to slots)	Mandatory	This association links Slots to the chassis.
10.7.6 CIM_ElementCapabilities	Optional	
10.7.7 CIM_ElementSettingData	Optional	

Table 42 - CIM Elements for Partitioned Tape Library

Element Name	Requirement	Description
10.7.8 CIM_HostedDependency (PTLSystem to Partition)	Mandatory	This association links the PTLSystem ComputerSystem object to the Partition ComputerSystem objects including the unallocated; hence 1ormore.
10.7.9 CIM_HostedDependency (PTLSystem to Unallocated Partition)	Mandatory	This association links the PTLSystem ComputerSystem object to the Unnallocated Partition.
10.7.10 CIM_LimitedAccessPort	Optional	The media export port for a PTL system (this is the physical pool).
10.7.11 CIM_MediaAccessDevice	Optional	If unknown, set to False.
10.7.12 CIM_PhysicalMediaInLocation	Optional	This association links media to the slots.
10.7.13 CIM_PhysicalTape	Mandatory	The media in the PTL Collection.
10.7.14 CIM_Product	Optional	Asset information for the system.
10.7.15 CIM_ProductElementComponent (PTL System)	Optional	
10.7.16 CIM_Realizes (Slots to Changers)	Mandatory	This association links changers to the slots.
10.7.17 CIM_Realizes (Slots to Ports)	Mandatory	This association links ports to the slots.
10.7.18 CIM_Realizes (Slots to TapeDrive)	Mandatory	This association links drives to the slots.
10.7.19 CIM_StorageMediaLocation	Optional	The slots and drive slots in a partition tape library system (the physical pool).
10.7.20 CIM_SystemDevice (PTL System to ChangerDevice)	Mandatory	This association links ChangerDevice to the scoping system.
10.7.21 CIM_SystemDevice (PTL System to LimitedAccessPort)	Mandatory	This association links LimitedAccessDevice to the scoping system.
10.7.22 CIM_SystemDevice (PTL System to MediaAccessDevice)	Mandatory	This association links MediaAccessDevice to the scoping system.
10.7.23 SNIA_ComputerSystem (PTL System)	Mandatory	'Top level' system that represents the entire Virtual Tape Library. Associated to RegisteredProfile.
10.7.24 SNIA_ComputerSystem (Partition)	Optional	'Top level' system that represents a Partition within a Tape Library.
10.7.25 SNIA_ComputerSystem (Unallocated Partition)	Mandatory	'Top level' system that represents the unallocated portion of the physical library .
10.7.26 SNIA_PartitionedLibrarySetting	Optional	Settings used to create the PTL.
10.7.27 SNIA_PartitionedLibrarySystemCapabilities	Optional	Features supported in PTL Service.

Table 42 - CIM Elements for Partitioned Tape Library

Element Name	Requirement	Description
10.7.28 SNIA_PartitionedLibrarySystemConfiguration Service	Optional	Services used to set up the PTLS hardware.
SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA SNIA_ComputerSystem AND ANY SourceInstance.SNIA_ComputerSystem::Dedicated[*] = 38	Mandatory	CQL -Partition was created.
SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_ProtocolControllerForUnit	Mandatory	
SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA SNIA_ComputerSystem AND ANY SourceInstance.SNIA_ComputerSystem::Dedicated[*] = 38	Mandatory	CQL -Partition was deleted.
SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_ProtocolControllerForUnit	Mandatory	
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA SNIA_ComputerSystem AND ANY SourceInstance.SNIA_ComputerSystem::Dedicated[*] = 38 AND SourceIsntance.SNIA_ComputerSystem::OperationalStatus <> PreviousInstance.SNIA_ComputerSystem::OperationalStatus	Mandatory	CQL -Status of a Partition or a PTL System has changed.
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_PhysicalTape	Mandatory	

10.7.1 CIM_ChangerDevice

Created By: Extrinsic
Modified By: Extrinsic
Deleted By: Extrinsic
Requirement: Optional

Table 43 describes class CIM_ChangerDevice.

Table 43 - SMI Referenced Properties/Methods for CIM_ChangerDevice

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

10.7.2 CIM_Chassis (PTL System)

Created By: Extrinsic
Modified By: Extrinsic
Deleted By: Extrinsic
Requirement: Optional

Table 44 describes class CIM_Chassis (PTL System).

Table 44 - SMI Referenced Properties/Methods for CIM_Chassis (PTL System)

Properties	Flags	Requirement	Description & Notes
Tag		Mandatory	
CreationClassName		Mandatory	
PackageType		Mandatory	Shall be 3 (ChassisFrame).
ChassisPackageType		Mandatory	
Manufacturer		Optional	
Model		Optional	
SerialNumber		Optional	
PartNumber		Optional	
SKU		Optional	
VendorCompatibilityStrings		Optional	
ElementName		Optional	

10.7.3 CIM_ComputerSystemPackage (PTL System to Chassis)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 45 describes class CIM_ComputerSystemPackage (PTL System to Chassis).

Table 45 - SMI Referenced Properties/Methods for CIM_ComputerSystemPackage (PTL System to Chassis)

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	
Dependent		Mandatory	

10.7.4 CIM_ConcreteIdentity (Slots to Slots)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 46 describes class CIM_ConcreteIdentity (Slots to Slots).

Table 46 - SMI Referenced Properties/Methods for CIM_ConcreteIdentity (Slots to Slots)

Properties	Flags	Requirement	Description & Notes
SystemElement		Mandatory	
SameElement		Mandatory	

10.7.5 CIM_Container (Chassis to slots)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 47 describes class CIM_Container (Chassis to slots).

Table 47 - SMI Referenced Properties/Methods for CIM_Container (Chassis to slots)

Properties	Flags	Requirement	Description & Notes
GroupComponent		Mandatory	
PartComponent		Mandatory	

10.7.6 CIM_ElementCapabilities

Requirement: Optional

Table 48 describes class CIM_ElementCapabilities.

Table 48 - SMI Referenced Properties/Methods for CIM_ElementCapabilities

Properties	Flags	Requirement	Description & Notes
ManagedElement		Mandatory	
Capabilities		Mandatory	

10.7.7 CIM_ElementSettingData

Requirement: Optional

Table 49 describes class CIM_ElementSettingData.

Table 49 - SMI Referenced Properties/Methods for CIM_ElementSettingData

Properties	Flags	Requirement	Description & Notes
ManagedElement		Mandatory	
SettingData		Mandatory	

10.7.8 CIM_HostedDependency (PTLSystem to Partition)

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

Table 50 describes class CIM_HostedDependency (PTLSystem to Partition).

Table 50 - SMI Referenced Properties/Methods for CIM_HostedDependency (PTLSystem to Partition)

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	PTLSystem ComputerSystem object.
Dependent		Mandatory	Partition ComputerSystem object.

10.7.9 CIM_HostedDependency (PTLSystem to Unallocated Partition)

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

Table 51 describes class CIM_HostedDependency (PTLSystem to Unallocated Partition).

Table 51 - SMI Referenced Properties/Methods for CIM_HostedDependency (PTLSystem to Unallocated Partition)

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	PTLSystem ComputerSystem object.
Dependent		Mandatory	Partition ComputerSystem object.

10.7.10 CIM_LimitedAccessPort

Created By: Extrinsic
Modified By: Extrinsic
Deleted By: Extrinsic
Requirement: Optional

Table 52 describes class CIM_LimitedAccessPort.

Table 52 - SMI Referenced Properties/Methods for CIM_LimitedAccessPort

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

10.7.11 CIM_MediaAccessDevice

Created By: Extrinsic
Modified By: Extrinsic
Deleted By: Extrinsic
Requirement: Optional

Table 53 describes class CIM_MediaAccessDevice.

Table 53 - SMI Referenced Properties/Methods for CIM_MediaAccessDevice

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	
CreationClassName		Mandatory	
SystemName		Mandatory	
DeviceID		Mandatory	
OperationalStatus		Optional	
StatusDescriptions		Optional	Additional information related to the values in OperationalStatus.
NeedsCleaning		Mandatory	
MountCount		Optional	

10.7.12 CIM_PhysicalMediaInLocation

Created By: Static
Modified By: Static
Deleted By: Static
Requirement: Optional

Table 54 describes class CIM_PhysicalMediaInLocation.

Table 54 - SMI Referenced Properties/Methods for CIM_PhysicalMediaInLocation

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	
Dependent		Mandatory	

10.7.13 CIM_PhysicalTape

Created By: Extrinsic
Modified By: Extrinsic
Deleted By: Extrinsic
Requirement: Mandatory

10.7.14 CIM_Product

Requirement: Optional

Table 55 describes class CIM_Product.

Table 55 - SMI Referenced Properties/Methods for CIM_Product

Properties	Flags	Requirement	Description & Notes
ElementName		Mandatory	
Name		Mandatory	
IdentifyingNumber		Mandatory	
Vendor		Mandatory	
Version		Mandatory	

10.7.15 CIM_ProductElementComponent (PTL System)

Requirement: Optional

Table 56 describes class CIM_ProductElementComponent (PTL System).

Table 56 - SMI Referenced Properties/Methods for CIM_ProductElementComponent (PTL System)

Properties	Flags	Requirement	Description & Notes
GroupComponent		Mandatory	
PartComponent		Mandatory	

10.7.16 CIM_Realizes (Slots to Changers)

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

Table 57 describes class CIM_Realizes (Slots to Changers).

Table 57 - SMI Referenced Properties/Methods for CIM_Realizes (Slots to Changers)

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	
Dependent		Mandatory	

10.7.17 CIM_Realizes (Slots to Ports)

Created By: Static

Modified By: Static
Deleted By: Static
Requirement: Mandatory

Table 58 describes class CIM_Realizes (Slots to Ports).

Table 58 - SMI Referenced Properties/Methods for CIM_Realizes (Slots to Ports)

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	
Dependent		Mandatory	

10.7.18 CIM_Realizes (Slots to TapeDrive)

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

Table 59 describes class CIM_Realizes (Slots to TapeDrive).

Table 59 - SMI Referenced Properties/Methods for CIM_Realizes (Slots to TapeDrive)

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	
Dependent		Mandatory	

10.7.19 CIM_StorageMediaLocation

Created By: Extrinsic
Modified By: Extrinsic
Deleted By: Extrinsic
Requirement: Optional

Table 60 describes class CIM_StorageMediaLocation.

Table 60 - SMI Referenced Properties/Methods for CIM_StorageMediaLocation

Properties	Flags	Requirement	Description & Notes
CreationClassName		Mandatory	
Tag		Mandatory	
LocationType		Mandatory	Slot, MediaAccessDevice, or Limited Access Port.
LocationCoordinates		Mandatory	

Table 60 - SMI Referenced Properties/Methods for CIM_StorageMediaLocation

Properties	Flags	Requirement	Description & Notes
MediaTypesSupported		Mandatory	
MediaCapacity		Mandatory	

10.7.20 CIM_SystemDevice (PTL System to ChangerDevice)

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

Table 61 describes class CIM_SystemDevice (PTL System to ChangerDevice).

Table 61 - SMI Referenced Properties/Methods for CIM_SystemDevice (PTL System to ChangerDevice)

Properties	Flags	Requirement	Description & Notes
PartComponent		Mandatory	
GroupComponent		Mandatory	

10.7.21 CIM_SystemDevice (PTL System to LimitedAccessPort)

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

Table 62 describes class CIM_SystemDevice (PTL System to LimitedAccessPort).

Table 62 - SMI Referenced Properties/Methods for CIM_SystemDevice (PTL System to LimitedAccessPort)

Properties	Flags	Requirement	Description & Notes
PartComponent		Mandatory	
GroupComponent		Mandatory	

10.7.22 CIM_SystemDevice (PTL System to MediaAccessDevice)

Created By: Static
Modified By: Static

Deleted By: Static
Requirement: Mandatory

Table 63 describes class CIM_SystemDevice (PTL System to MediaAccessDevice).

Table 63 - SMI Referenced Properties/Methods for CIM_SystemDevice (PTL System to MediaAccessDevice)

Properties	Flags	Requirement	Description & Notes
PartComponent		Mandatory	
GroupComponent		Mandatory	

10.7.23 SNIA_ComputerSystem (PTL System)

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 'Partitioned Tape Library', RegisteredOrganization set to 'SNIA', and RegisteredVersion set to '1.4.0'.

Table 64 describes class SNIA_ComputerSystem (PTL System).

Table 64 - SMI Referenced Properties/Methods for SNIA_ComputerSystem (PTL System)

Properties	Flags	Requirement	Description & Notes
CreationClassName		Mandatory	
Name		Mandatory	Unique identifier for the PTL System. 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 PTL system.
NameFormat		Mandatory	Format for Name property. **HID** is a required format. Others are optional.
OperationalStatus		Mandatory	Overall status of the system.
StatusDescriptions		Optional	Additional information related to the values in OperationalStatus.
ElementName		Mandatory	User friendly name.
PrimaryOwnerContact	M	Optional	Contact details for PTL system owner.

Table 64 - SMI Referenced Properties/Methods for SNIA_ComputerSystem (PTL System)

Properties	Flags	Requirement	Description & Notes
PrimaryOwnerName	M	Optional	Owner of the PTL System.
OtherIdentifyingInfo		Optional	Other data that could be used to identify the PTL system.
IdentifyingDescriptions		Optional	Provides explanations and details for the entries in the OtherIdentifyingInfo property.

10.7.24 SNIA_ComputerSystem (Partition)

Created By: Extrinsic
Modified By: Extrinsic
Deleted By: Extrinsic
Requirement: Optional

Table 65 describes class SNIA_ComputerSystem (Partition).

Table 65 - SMI Referenced Properties/Methods for SNIA_ComputerSystem (Partition)

Properties	Flags	Requirement	Description & Notes
CreationClassName		Mandatory	
Name		Mandatory	
Dedicated		Mandatory	Indicates that this computer system is dedicated to operation as a PTL.
NameFormat		Mandatory	Format for Name property.
OperationalStatus		Mandatory	Overall status of the system.
StatusDescriptions		Optional	Additional information related to the values in OperationalStatus.
ElementName		Mandatory	User friendly name.
PrimaryOwnerContact	M	Optional	Contact details for PTL owner.
PrimaryOwnerName	M	Optional	Owner of the PTL.
OtherIdentifyingInfo		Optional	Other data that could be used to identify the PTL.
IdentifyingDescriptions		Optional	Provides explanations and details for the entries in the OtherIdentifyingInfo property.

10.7.25 SNIA_ComputerSystem (Unallocated Partition)

Created By: Static
Modified By: Static
Deleted By: Static

Requirement: Mandatory

Table 66 describes class SNIA_ComputerSystem (Unallocated Partition).

Table 66 - SMI Referenced Properties/Methods for SNIA_ComputerSystem (Unallocated Partition)

Properties	Flags	Requirement	Description & Notes
CreationClassName		Mandatory	
Name		Mandatory	Unique identifier for the PTL System. 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 PTL system.
NameFormat		Mandatory	Format for Name property. **HID** is a required format. Others are optional.
OperationalStatus		Mandatory	Overall status of the system.
StatusDescriptions		Optional	Additional information related to the values in OperationalStatus.
ElementName		Mandatory	User friendly name.
PrimaryOwnerContact	M	Optional	Contact details for PTL system owner.
PrimaryOwnerName	M	Optional	Owner of the PTL System.
OtherIdentifyingInfo		Optional	Other data that could be used to identify the PTL system.
IdentifyingDescriptions		Optional	Provides explanations and details for the entries in the OtherIdentifyingInfo property.

10.7.26 SNIA_PartitionedLibrarySetting

Created By: Static
Modified By: Static
Deleted By: Static
Requirement: Optional

Table 67 describes class SNIA_PartitionedLibrarySetting.

Table 67 - SMI Referenced Properties/Methods for SNIA_PartitionedLibrarySetting

Properties	Flags	Requirement	Description & Notes
InstanceID		Mandatory	
LibraryType		Mandatory	

Table 67 - SMI Referenced Properties/Methods for SNIA_PartitionedLibrarySetting

Properties	Flags	Requirement	Description & Notes
SupportedMediaTypes		Mandatory	
LibraryName		Optional	If present, this shall be the name of the Partition associated with these settings.
SlotCount		Optional	If present, this shall be the number of slots in the Partition associated with these settings.
DriveCount		Optional	If present, this shall be the number of drives in the Partition associated with these settings.
PartitionMethod		Mandatory	
RangeStartAddresses		Optional	If present, this shall be the starting addresses for ranges of StorageMediaLocations or volume serial numbers of tape cartridges that make up partition depending on creation method.
RangeEndAddresses		Optional	If present, this shall be the ending addresses for ranges of StorageMediaLocations or volume serial number of tape cartridges that make up partition depending on creation method.
RangeType		Optional	If present, this shall be the type of the range (StorageMediaLocation addresses or cartridge volume serial numbers) given in RangeStartAddresses and RangeEndAddresses.

10.7.27 SNIA_PartitionedLibrarySystemCapabilities

Created By: Static
Modified By: Static
Deleted By: Static
Requirement: Optional

Table 68 describes class SNIA_PartitionedLibrarySystemCapabilities.

Table 68 - SMI Referenced Properties/Methods for SNIA_PartitionedLibrarySystemCapabilities

Properties	Flags	Requirement	Description & Notes
InstanceID		Mandatory	
PartitionMethodsSupported		Mandatory	
MaxPartitions		Mandatory	
LibraryTypesSupported		Optional	

Table 68 - SMI Referenced Properties/Methods for SNIA_PartitionedLibrarySystemCapabilities

Properties	Flags	Requirement	Description & Notes
ModifySupported		Mandatory	
ActivateDeactivatePartitionSupported		Mandatory	
ActivateDeactivateLibrarySupported		Mandatory	

10.7.28 SNIA_PartitionedLibrarySystemConfigurationService

Created By: Static
Modified By: Static
Deleted By: Static
Requirement: Optional

Table 69 describes class SNIA_PartitionedLibrarySystemConfigurationService.

Table 69 - SMI Referenced Properties/Methods for SNIA_PartitionedLibrarySystemConfigurationService

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	
CreationClassName		Mandatory	
SystemName		Mandatory	
Name		Mandatory	
CreatePartition()		Optional	Creates a partition using one of the supported methods.
DeletePartition()		Optional	Delete an existing partition.
CreatePartitionAuto()		Optional	Creates a partition by equally distributing the resources among the specified numbers.
ModifyPartition()		Optional	Modify an existing partition using the same LibrarySetting.
ActivatePartition()		Optional	Make a partition active.
DeactivatePartition()		Optional	Make a partition inactive.
ActivateLibrary()		Optional	Activate all partitions.
DeactivateLibrary()		Optional	Deactivate all partitions.

EXPERIMENTAL

EXPERIMENTAL

Clause 11: Virtual Tape Library Profile

11.1 Description

11.1.1 Overview

This profile describes the model for a Virtual Library System. The Virtual Library System uses disk and/or tape storage to emulate one or more tape libraries. A Virtual Library System can use local storage (arrays, JBOD, or tape libraries) or connect to external storage. In the case of local storage the Virtual Library System model may optionally include Storage Media Library as a supported profile. Figure 19 shows the basic components of the Virtual Library System.

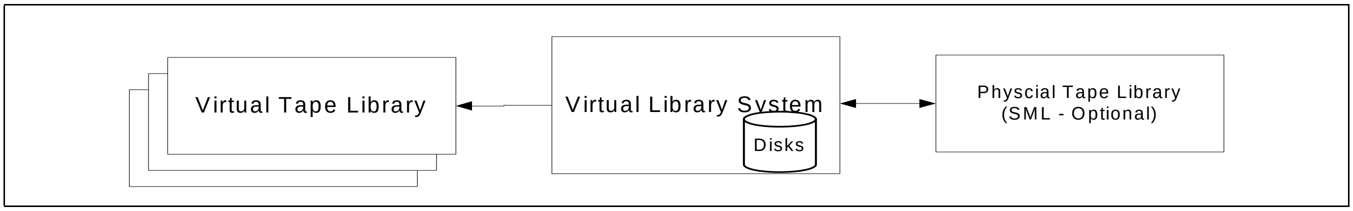


Figure 19 - Block Diagram

11.1.2 Package

The Virtual Tape Library Profile doesn’t stand alone. Figure 20 shows the component profiles that work with the Virtual Tape Library Profile to model a complete Virtual Library System product.

The objects in the center of Figure 20 represent the Virtual Library System Profile. The Virtual Library System uses specialized versions of the Generic Target Port and Generic initiator Port Profile to model the ports.

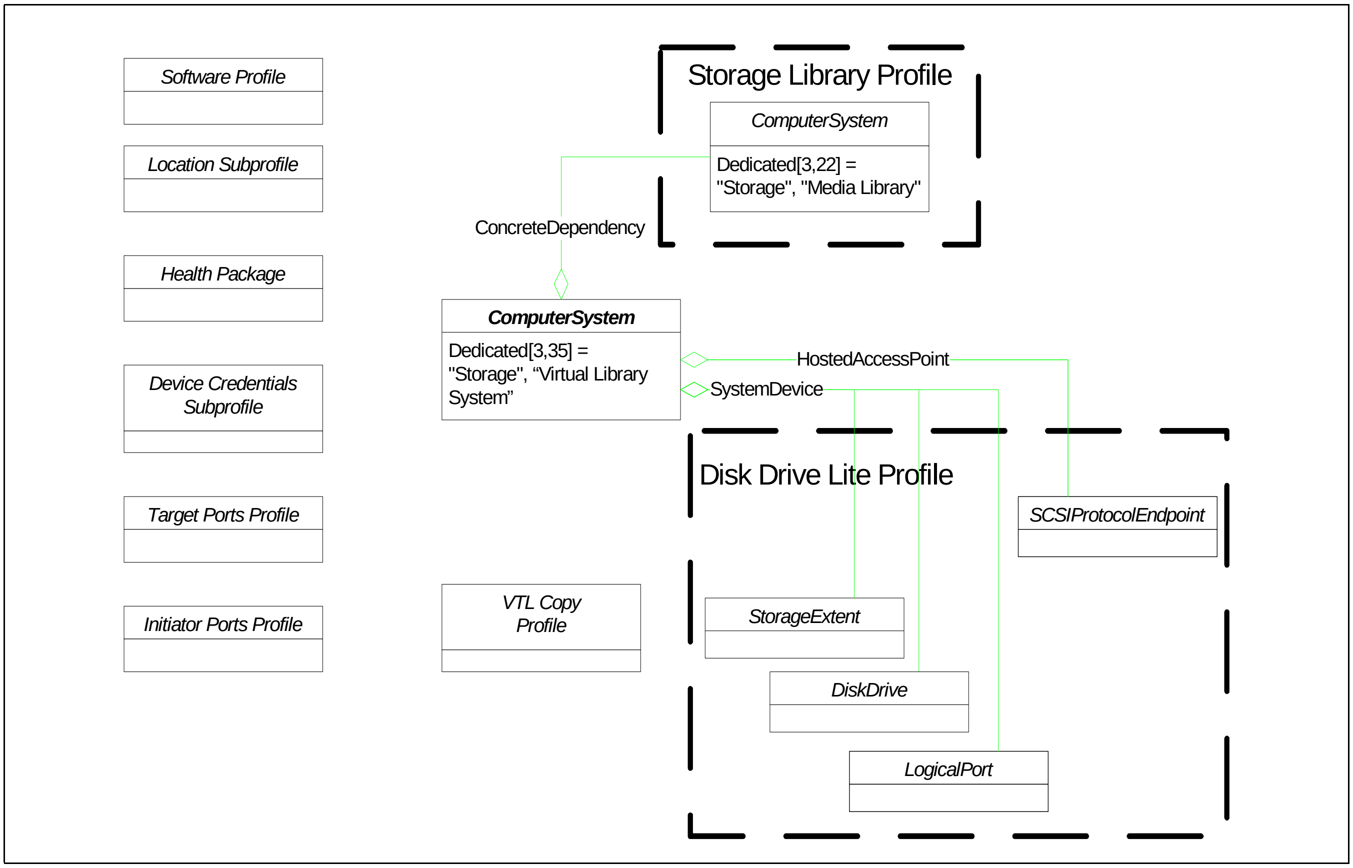


Figure 20 - Virtual Library System Package Diagram

11.1.3 Virtual Library System

11.1.3.1 Overview

Figure 21 shows the basic model of a Virtual Library System. This diagram does not contain all the classes and associations required to implement the profile but gives a picture of the main classes and associations as well as how they interact with major component profiles.

11.1.3.2 Virtual Library System ComputerSystem objects

The top-level system is modeled with **CIM_ComputerSystem;** the value of Dedicated includes 3 (Storage) and 35 (Virtual Library System). It shall be referenced by the CIM_ElementConformstoProfile association from the Profile Registration Profile. This object is also associated by CIM_SystemDevice to logical devices that are part of the Virtual Library System.

Virtual libraries shall have a CIM_ComputerSystem object with the Dedicated property including 3 (Storage) and 34 (Virtual Tape Library). The virtual library CIM_ComputerSystem object is associated by CIM_HostedDependancy. The box on the left of the Figure 21 contains the objects that represent a single Virtual Library. These classes shall be used for each Virtual Tape Library emulated by the system.

Physical libraries in the system shall have CIM_ComputerSystem objects with Dedicated property including values of 3 (Storage) and 22 (Media Library). The CIM_ComputerSystem objects are associated by CIM_ConcreteDependency. The physical library shall be modeled by the Storage Library Profile.

Disks may be modeled using the Disk Drive Lite Profile. StorageExtent instances from Disk Drive Lite shall be associated to this profile’s primordial StoragePool via ConcreteComponent. Storage from an array may also be used. In this case, each array LUN is modeled as a StorageExtent instance associated to this profile’s primordial pool via ConcreteComponent.

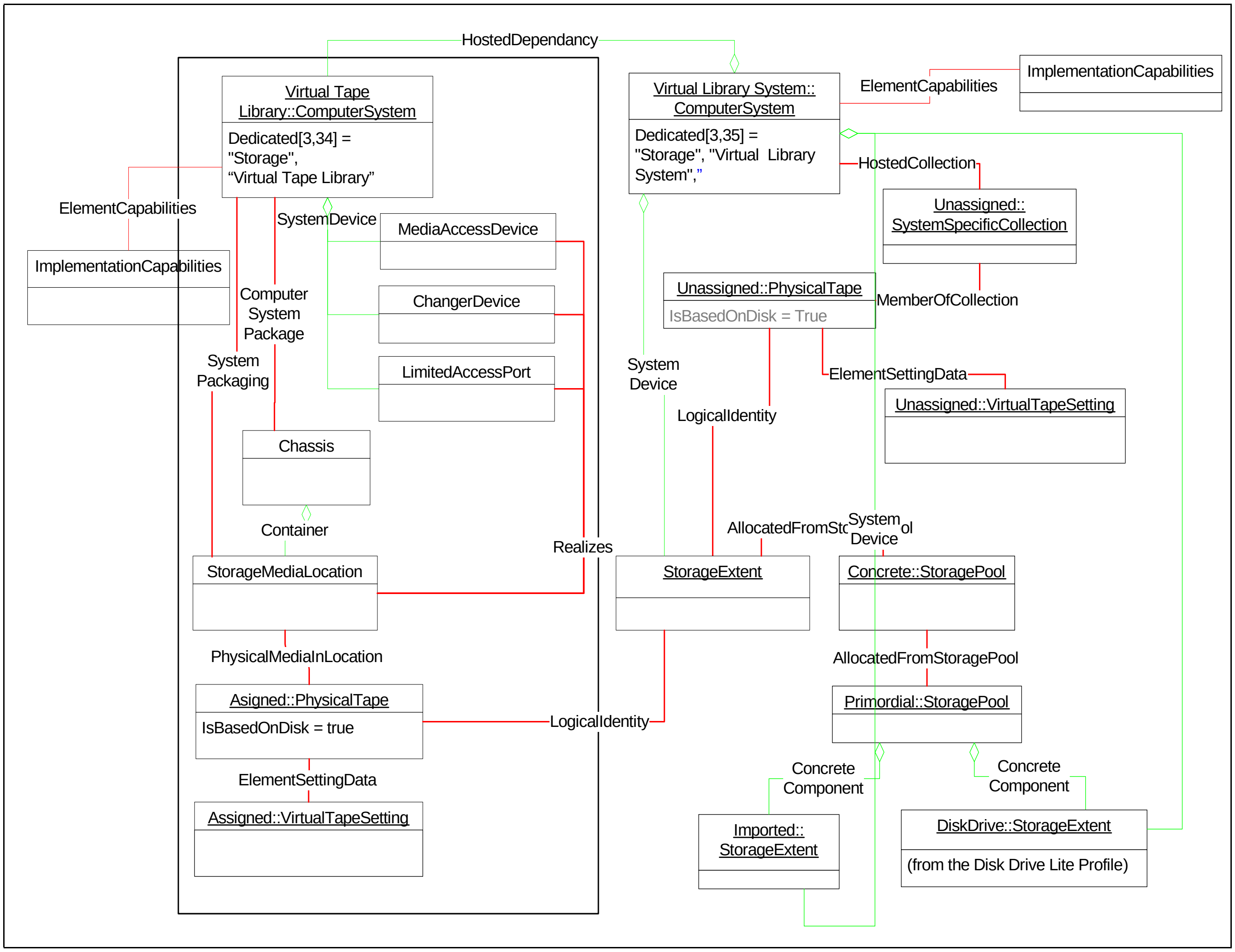


Figure 21 - Virtual Tape Library System

EXPERIMENTAL

All three of theComputerSystems involved (the Virtual Library System, the Virtual Tape Library systems and the Storage Library System) have an associated instance of CIM_ImplementationCapabilities. However, in the case of the Storage Library System, the capabilities is actually an instance of CIM_StorageLibraryCapabilities (a subclass of CIM_ImplementationCapabilities). The capabilities of the system implementations are identified in an instance of CIM_ImplementationCapabilities (or CIM_StorageLibraryCapabilities), which is associated to the ComputerSystem via ElementCapabilities. This includes information on the capacity optimization techniques supported by the implementation and views supported by the implementation. Note that the capabilities specified for a Virtual Tape Library system should be a subset of the capabilities specified for the Virtual Library System.

EXPERIMENTAL

11.1.3.3 Unassigned Tape Collection

When a VTL is deleted using DeleteLibrary() and the SaveTapes parameter is set to true, the PhysicalTape instances modeling the tapes are disconnected from the StorageMediaLocation instances for the VTL and associated to the SystemSpecificCollection associated to the Virtual Library System.

11.1.3.4 Allocated StorageExtent and SystemDevice

CIM requires that all LogicalDevice instances be associated to exactly one ComputerSystem instance via the SystemDevice association. This requirement applies to StorageExtent instances allocated from the concrete StoragePool. When the StorageExtent is instanced (by invoking CreateTapeFromPool()), it is logically owned by a VTL, but at a later time, it may be part of the Unassigned Tape Collection (see 11.1.3.3) which is related to the Virtual Library System. Since StorageExtent.SystemName is a key property, it not possible to re-associate a StorageExtent instance to a different ComputerSystem. To address this, the SystemDevice associations referencing StorageExtent instances shall always be associated to the Virtual Library System ComputerSystem.

11.1.3.5 Block to Tape

Figure 22 details the objects involved in modeling the use of block storage to emulate virtual tapes.

The primordial CIM_StoragePool instances represent the block storage available in the Virtual Library System. Virtual tapes are in turn allocated from these pools. Virtual tapes are modeled by CIM_StorageExtents associated to the pools by CIM_AlocatedFromStoragePool. The virtual tape CIM_StorageExtents are also associated to CIM_PhysicalTape objects.

Imported logical units from disks or arrays are modeled as instances of StorageExtent associated to a primordial storage pool. If this imported storage is from disks, the disks should be modeled using the Disk Drive Lite Profile with the StorageExtent associated to the primordial pool being the StorageExtent instance defined in the Disk Drive Lite Profile.

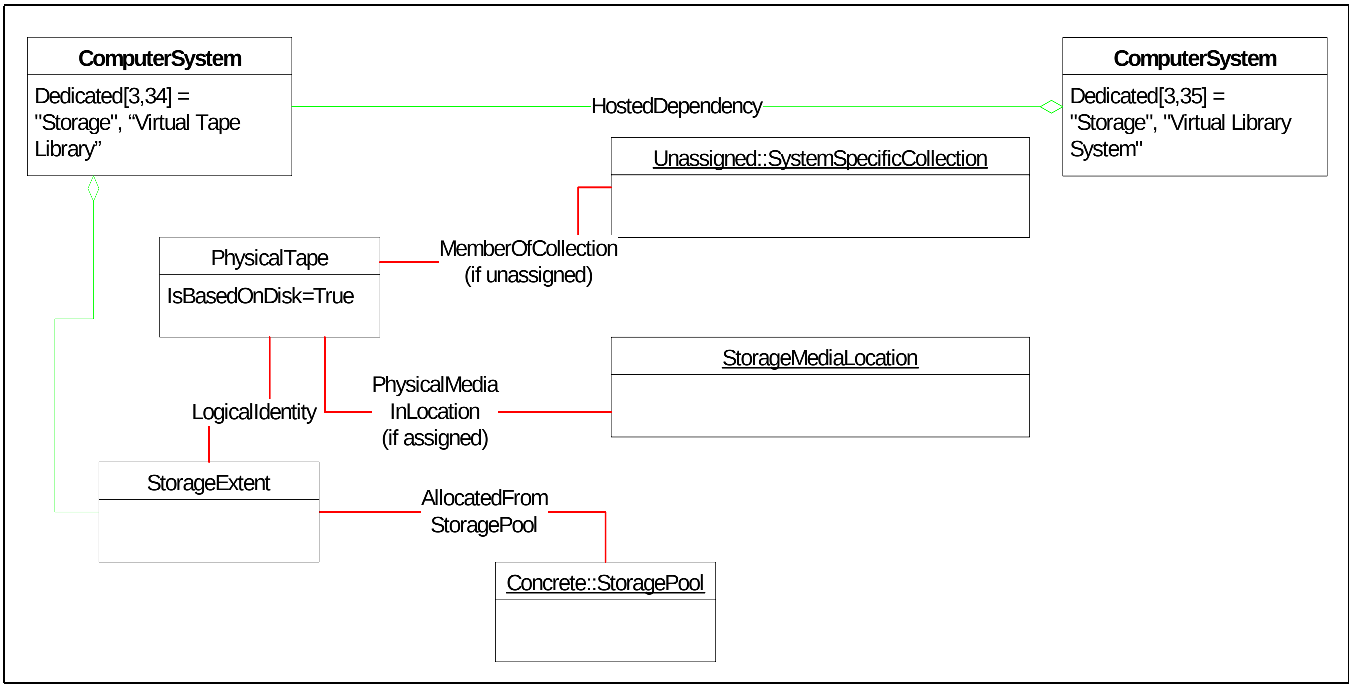

ComputerSystem
Dedicated[3,34] = "Storage", "Virtual Tape Library"
HostedDependency
ComputerSystem
Dedicated[3,35] = "Storage", "Virtual Library System"
Unassigned::SystemSpecificCollection
PhysicalTape
IsBasedOnDisk=True
MemberOfCollection (if unassigned)
StorageMediaLocation
LogicalIdentity
PhysicalMedia InLocation (if assigned)
StorageExtent
AllocatedFrom StoragePool
Concrete::StoragePool

Figure 22 - VTL - Block to Tape

11.1.3.6 Virtual Library model

Virtual libraries shall have a CIM_ComputerSystem object with dedicated values of 3 (Storage) and 34 (Virtual Tape Library). The box on the left of the Figure 21 contains the objects that represent a single Virtual Library. These objects shall be replicated for each virtual Library emulated by the system. This CIM_ComputerSystem object shall scope the objects that are part of an instance. Logical devices that are part of the virtual library shall have CIM_SystemDevice associations back to the CMI_ComputerSystems object.

Each library shall have a CIM_Chassis, CIM_ChangerDevice, one or more CIM_MediaAccessDevice, one or more CIM_LimitedAccessPorts, and many CIM_StorageMediaLocation. These logical objects represent the virtual library the Virtual Library System is emulating.

The CIM_Chassis and the CIM_StorageMediaLocation objects represent the slots in a physical jukebox. They answer to inband and SMI-S move media commands as if the were physical slots.

There shall be one CIM_MediaAccessDevice object for each tape drive the virtual library is emulating. These objects shall be created and destroyed by the configuration commands described in Configuration of hardware (11.1.4.1) and assigned to ports by methods defined in Inband access (11.1.4.2)

CIM_LimitedAccessPorts may be able to eject virtual tapes. Ejecting a virtual tape will cause a copy to physical media and then the ejection of the physical media.

11.1.3.7 Physical Library Model

Physical libraries in the system shall have CIM_ComputerSystem objects with dedicated values of 3 (Storage) and 22 (Media Library). The CIM_ComputerSystem objects are associated by CIM_ConcreteDependency. The physical library shall be modeled by the Storage Library Profile.

11.1.4 Virtual Library System configuration

The Virtual Library System model contains four main functions (Hardware Configuration, Virtual Library Configuration, Virtual Library management, Physical Library management).

11.1.4.1 Configuration of hardware

11.1.4.1.1 Services

The Virtual Library System Service class and Virtual Library System Capabilities class define methods used to configure the hardware of a Virtual Library System. The service contains the methods while the capabilities class contains properties that defines the methods and limits support by the implementation. Figure 23 shows the model for these classes.

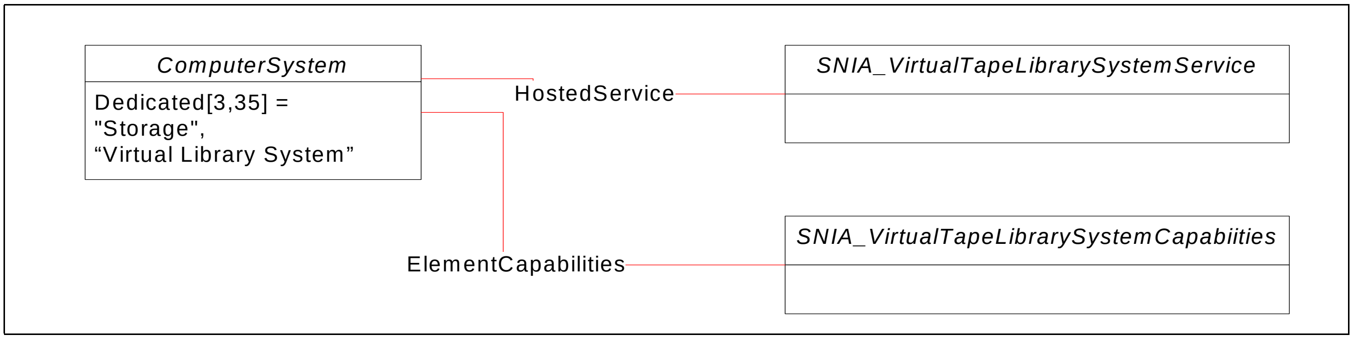


Figure 23 - Virtual Library System-Services

11.1.4.1.2 Array or Disk Configuration

A Virtual Library System uses block storage to hold images of virtual tapes. If the block storage comes from disks, the disks are modeled either using a primordial StoragePool or the Disk Drive Lite Subprofile. If the block storage comes from RAID array systems, the imported LUNs are modeled as instances of StorageExtent associated to a primordial StoragePool.

RescanPhysicalHardware() causes the Virtual Library System to scan for external arrays and tape libraries.

11.1.4.1.3 Physical Tape Library configuration

Some Virtual Library System use physical tape libraries as storage for virtual tape libraries or as the destination of copy operations. The physical Library storage is modeled as either a direct attached storage media library. The SNIA_VirtualTapeLibrarySystemCapabilities class contains the following properties:

SupportsPhysicalLibrary is a uint32 that is set to a value of 2 (None) if the Virtual Library System does not have any physical library support or is set to a value of 3 (Local) if the Virtual Library System has a local library attached or a value of 4 (External) if a cascaded physical can be accessed. The rest of the properties are conditional on the SupportsPhysicalLibrary property being set to a value of either 3 or 4.

The SNIA_VirtualTapeLibrarySystemService class contains the following methods to attach storage media libraries:

ListPLibrary((out)LibraryList[]) is used to list potential tape libraries.

AttachPLibrary(Library) is conditional on the ExternalLibrary property being TRUE. The Library property is the “ID” of the library to attach. The ID is obtained from the ListLibrary() method.

DetachPLibrary(Library) removes access to an external library. The “Library” parameter is a REF to the CIM_ComputerSystem object for the library. NOTE: detaching a library stops all access to it and disconnects all associations to the Media Library model.

11.1.4.1.4 Port model

A Virtual Library System has multiple ports. These ports are used as targets (to provide service to a host) and/or as initiators (to communicate with external arrays and Physical Tape Libraries). The ports shall be modeled using the specializations of the Generic Initiator Port or Generic Target Port Profiles. The CIM_logicalPort.UsageRestriction property shall be used to indicate the port usage.

The Virtual Library System service includes an optional method (SetPortUsage) to configure the usage of the ports. The Virtual Library SystemCababilities.ConfigPort property indicates if this method is supported.

11.1.4.2 Inband access

After the host facing ports (target ports) are defined, the inband access to virtual libraries, physical Libraries, and any other inband access is setup. Figure 24 is an instance diagram showing the model used to map/mask devices to the host facing ports.

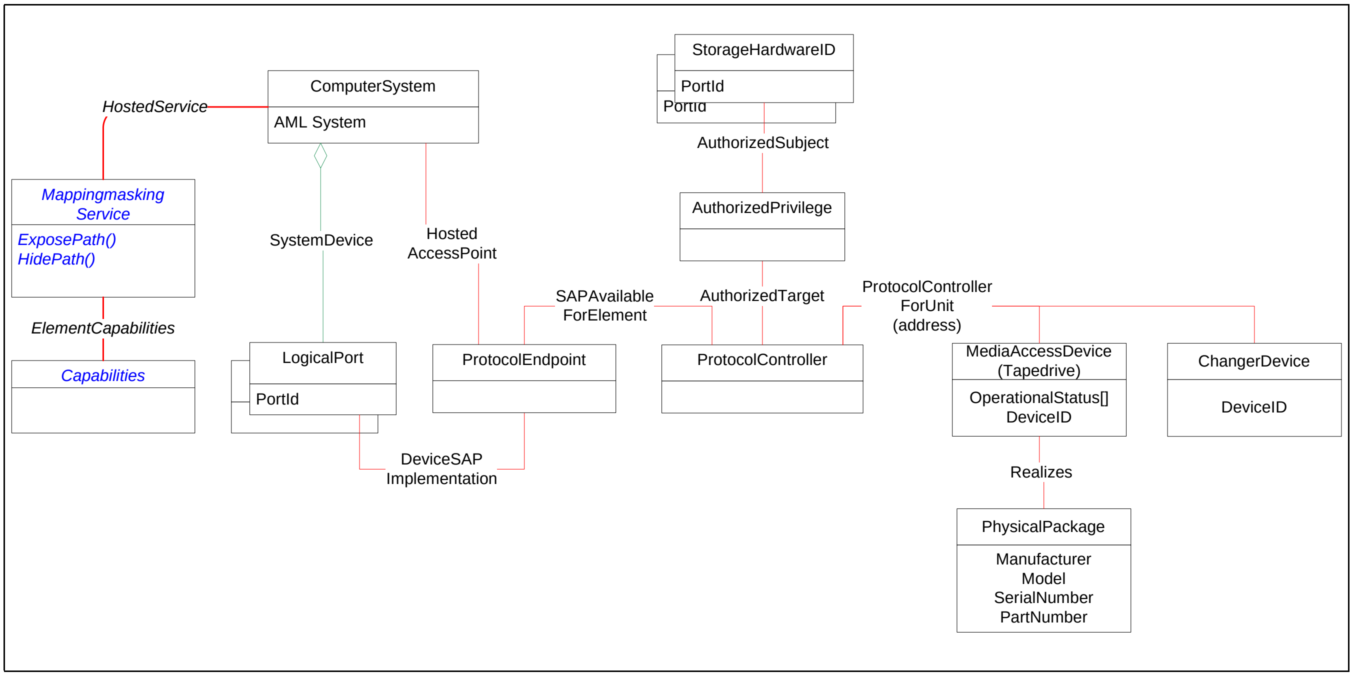


Figure 24 - Drive Mapping

11.1.4.3 Virtual Libraries Configuration

Figure 25 shows the part of the model related to the management of virtual libraries in a Virtual Tape Library System.

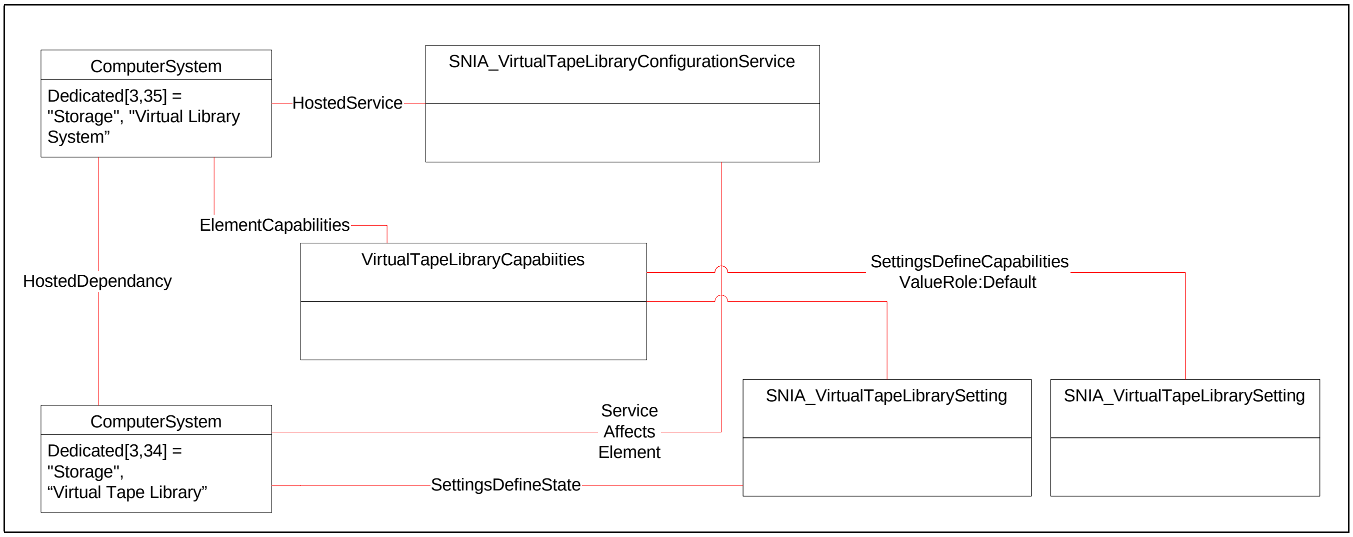

ComputerSystem
Dedicated[3,35] = "Storage", "Virtual Library System"
HostedService
SNIA_VirtualTapeLibraryConfigurationService
ElementCapabilities
VirtualTapeLibraryCapabiities
SettingsDefineCapabilities ValueRole:Default
HostedDependancy
SNIA_VirtualTapeLibrarySetting
SNIA_VirtualTapeLibrarySetting
ComputerSystem
Dedicated[3,34] = "Storage", “Virtual Tape Library”
Service Affects Element
SettingsDefineState

Figure 25 - Virtual Library Services

The SNIA_VirtualTapeLibraryConfigurationService class contains the following methods to manage Virtual media:

CreateLibrary(VirtualTapeLibrarySetting) is a required method. The method creates a virtual library using the information in the VTL setting object passed in. The base setting object is provided by the VTL service (canned). The object is copied and the variables are set. The object is then passed to this method. The VTL is created and the setting object is detached from the VTL service and attached to the VTL CIM_ComputerSystem object.

ModifyLibrary(VirtualTapeLibrarySetting)

ModifyLibrary is a required method. The method takes one parameter a REF to VirtualTapeLibrarySetting object associated to the VTL CIM_ComputerSystem object. The object contains a variable “Modify” that is an array containing a list of variables that may be modified.

DeleteLibrary (Library, SaveTapes) Deletes a virtual library. The parameter “Library” is a REF to the CIM_ComputerSystem of the Virtual Tape Library. The virtual tapes in the slots will also be deleted and their storage returned to the pool.

11.1.4.4 Virtual Tape Service

Figure 26 shows the part of the model related to the management of virtual tapes in a virtual tape library.

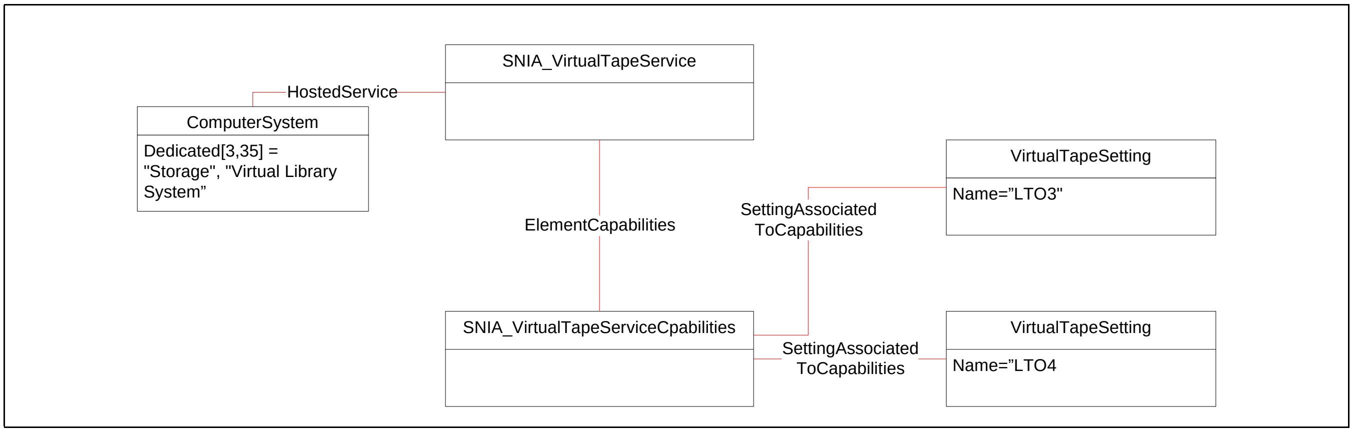

SNIA_VirtualTapeService
HostedService
ComputerSystem
Dedicated[3,35] = "Storage", "Virtual Library System”
VirtualTapeSetting
Name=”LTO3"
ElementCapabilities
SettingAssociated ToCapabilities
SNIA_VirtualTapeServiceCpabilities
SettingAssociated ToCapabilities
VirtualTapeSetting
Name=”LTO4

Figure 26 - Virtual Tape Service

The SNIA_VirtualTapeService class contains the following methods to manage virtual tapes:

CreateTapeFromPool(Pool, Setting, Library, StartingLocation, Count) is required to create virtual tapes from available storage.

Pool is a reference to the Pool instance the virtual tape is to be allocated from.

Setting is a reference to a VirtualTapeLibrarySetting instance that defines the type of tape being emulated.

Library is a reference to the CIM_ComputerSystem instance that represents the Virtual Tape Library.

StartingLocation is an integer with the slot number in it. New tapes will be put in this slot and higher numbered slots.

Count is an integer containing the number of tapes to be created.

ReturnTapeToPool(Tape) removes a tape from it’s location and returns the storage to the Pool.

Tape is a reference to the CIM_PhysicalTape instance that represents the tape to be deleted.

MoveMedia(Source, Destination) moves virtual tapes from one slot to another.

Source is a reference to the CIM_StorageMediaLocation instance that represents the slot containing the virtual tape.

Destination is a reference to the CIM_StorageMediaLocation instance that is the destination of the virtual tape.

11.2 Health and Fault Management Consideration

Not supported in this version of the standard.

11.3 Cascading Considerations

Not supported in this version of the standard.

11.4 Supported Profiles and Packages

Table 70 describes the supported profiles for Virtual Tape Library.

Table 70 - Supported Profiles for Virtual Tape Library

Profile Name	Organization	Version	Requirement	Description
Disk Drive Lite	SNIA	1.6.0	Optional	
FC Initiator Ports	SNIA	1.6.0	Optional	
Health	SNIA	1.2.0	Mandatory	
Software	SNIA	1.4.0	Optional	
Storage Library	SNIA	1.6.0	Optional	
Multiple Computer System	SNIA	1.2.0	Optional	
Masking and Mapping	SNIA	1.6.0	Optional	
Tape Copy Service	SNIA	1.3.0	Optional	Experimental.
Storage Server Asymmetry	SNIA	1.4.0	Optional	
Location	SNIA	1.4.0	Optional	
Library Views	SNIA	1.6.0	Optional	Experimental.
VTL Capacity and Reporting	SNIA	1.6.0	Optional	Experimental.
Indication	SNIA	1.5.0	Support for at least one is mandatory.	Deprecated. See the SNIA Indications Profile
Indications	SNIA	1.6.0		Experimental.
Indications	DMTF	1.2.0		Experimental. See DSP1054, version 1.2.0
FC Target Ports	SNIA	1.4.0	Support for at least one is mandatory.	
SAS Target Ports	SNIA	1.4.0		Experimental.
SPI Target Ports	SNIA	1.4.0		Experimental.

11.5 Methods of the profile

Not defined in this standard.

11.6 Client Considerations and Recipes

None.

11.7 Registered Name and Version

Virtual Tape Library version 1.6.0 (Autonomous Profile)

11.8 CIM Elements

Table 71 describes the CIM elements for Virtual Tape Library.

Table 71 - CIM Elements for Virtual Tape Library

Element Name	Requirement	Description
11.8.1 CIM_AllocatedFromStoragePool (Pool from Concrete Pool)	Optional	Pool allocated from Concrete Pool.
11.8.2 CIM_AllocatedFromStoragePool (Pool from Primordial Pool)	Mandatory	Pool allocated from Primordial Pool.
11.8.3 CIM_AllocatedFromStoragePool (StorageExtent from Concrete Pool)	Mandatory	StorageExtent from Concrete Pool.
11.8.4 CIM_ChangerDevice	Optional	Media changer for a Virtual Tape Library.
11.8.5 CIM_Chassis (Virtual Tape Library)	Optional	Aggregates the virtual locations for tapes and drives of one Virtual Tape Library.
11.8.6 CIM_ComputerSystem (Virtual Library System)	Mandatory	ComputerSystem that represents the Virtual Library System. Associated to RegisteredProfile.
11.8.7 CIM_ComputerSystem (Virtual Tape Library)	Optional	ComputerSystem that represents a Virtual Tape Library.
11.8.8 CIM_ComputerSystemPackage	Mandatory	Associates Chassis to the Virtual Tape Library ComputerSystem.
11.8.9 CIM_ConcreteComponent (StorageExtent from Primordial Pool)	Optional	Associates Primordial StoragePools and StorageExtents imported from arrays or the Disk Drive Lite profile.
11.8.10 CIM_ConcreteDependency (Virtual Library System to MediaLibrary)	Conditional	Conditional requirement: Support for SML profile. Associates the Virtual Library System ComputerSystem object to a Media Library ComputerSystem objects.

Table 71 - CIM Elements for Virtual Tape Library

Element Name	Requirement	Description
11.8.11 CIM_Container (Chassis to StorageMediaLocations)	Mandatory	Associates StorageMediaLocations to the chassis.
11.8.12 CIM_ElementCapabilities (ImplementationCapabilities to Virtual Library System)	Optional	Experimental. Associates the conformant Virtual Library System ComputerSystem to the CIM_ImplementationCapabilities supported by the implementation.
11.8.13 CIM_ElementCapabilities (ImplementationCapabilities to Virtual Tape Library System)	Optional	Experimental. Associates a Virtual Tape Library ComputerSystem to the CIM_ImplementationCapabilities supported by the implementation.
11.8.14 CIM_ElementCapabilities (Virtual Tape Library Capabilities)	Optional	Assocates VirtualTapeLibraryCapabilities and Virtual Library System ComputerSystem.
11.8.15 CIM_ElementCapabilities (Virtual Tape Library System Capabilities)	Optional	Associates VirtualTapeLibraryCapabilities and Virtual Library System ComputerSystem.
11.8.16 CIM_ElementCapabilities (Virtual Tape Service Capabilities)	Optional	Associates VirtualTapeServiceCapabilities and VirtualTapeService.
11.8.17 CIM_ElementSettingData (Physical Tape)	Optional	Associates PhysicalTape and VirtualTapeSetting.
11.8.18 CIM_HostedCollection	Optional	Associates the Virtual Library System ComputerSystem and the unassigned tapes SystemSpecificCollection.
11.8.19 CIM_HostedDependency (Virtual Library System to VirtualLibrary)	Mandatory	Associates the Virtual Library System ComputerSystem object to the Virtual Tape Library ComputerSystem objects.
11.8.20 CIM_HostedService (Virtual Tape Library Configuration Service)	Optional	Associates the VirtualTapeLibraryConfigurationService to the ComputerSystem representing the Virtual Library System.
11.8.21 CIM_HostedService (Virtual Tape Library System Service)	Optional	Associates the VirtualTapeLibrarySystemService to the ComputerSystem representing the Virtual Library System.
11.8.22 CIM_HostedService (Virtual Tape Service)	Optional	Associates VirtualTapeService to the ComputerSystem representing the Virtual Library System.
11.8.23 CIM_HostedStoragePool (Primordial)	Mandatory	Associates Virtual Library System ComputerSystem and StoragePools.
11.8.24 CIM_ImplementationCapabilities (ImplementationCapabilities)	Optional	Experimental. The capabilities of the profile implementation.
11.8.25 CIM_LimitedAccessPort	Optional	The tape export port for a Virtual Tape Library.
11.8.26 CIM_LogicalIdentity	Mandatory	Associates PhysicalTape and StorageExtent.

Table 71 - CIM Elements for Virtual Tape Library

Element Name	Requirement	Description
11.8.27 CIM_MediaAccessDevice	Mandatory	The tape drive for a Virtual Tape Library.
11.8.28 CIM_MemberOfCollection	Optional	Associates the unassigned tapes SystemSpecificCollection and PhysicalTape.
11.8.29 CIM_PhysicalMediaInLocation	Optional	Associates tape to the StorageMediaLocations.
11.8.30 CIM_Product	Optional	Asset information for the system.
11.8.31 CIM_ProductElementComponent (Virtual Tape Library)	Optional	Associates a Virtual Library System or Virtual Tape Library ComputerSystem with Product.
11.8.32 CIM_Realizes (Slots to Changers)	Optional	Associates changers to the StorageMediaLocations.
11.8.33 CIM_Realizes (Slots to Ports)	Optional	Associates access ports to the StorageMediaLocations.
11.8.34 CIM_Realizes (Slots to TapeDrive)	Mandatory	Associates drives to the StorageMediaLocations.
11.8.35 CIM_ServiceAffectsElement	Mandatory	Associates Virtual Tape Library Computer System and VirtualTapeLibraryConfigurationService.
11.8.36 CIM_SettingAssociatedToCapabilities (Setting To Capabilities)	Mandatory	Associates VirtualTapeServiceCapabilities and VirtualTapeSetting.
11.8.37 CIM_SettingsDefineCapabilities	Mandatory	Assocates VirtualTapeLibraryCapabilities and VirtualTapeLibrarySetting.
11.8.38 CIM_SettingsDefineState	Mandatory	Assocates Virtual Tape Library ComputerSystem and VirtualTapeLibrarySetting.
11.8.39 CIM_StorageExtent (Assigned)	Optional	Capacity allocated from an underlying array providing capacity for a virtual tape.
11.8.40 CIM_StorageExtent (Imported)	Optional	A LUN allocated from an underlying array.
11.8.41 CIM_StorageMediaLocation	Optional	The tape and drive locations in a virtual tape library.
11.8.42 CIM_StoragePool (Concrete)	Mandatory	The concrete StoragePool.
11.8.43 CIM_StoragePool (Primordial)	Mandatory	The pool of all storage available from the backing store.
11.8.44 CIM_SystemDevice (System to Primordial StorageExtent)	Optional	Associates StorageExtents to the Virtual Library System.
11.8.45 CIM_SystemDevice (VTL to ChangerDevice)	Optional	Associates ChangerDevice to the Virtual Tape Library ComputerSystem.
11.8.46 CIM_SystemDevice (VTL to LimitedAccessPort)	Optional	Associates LimitedAccessDevice to the Virtual Tape Library ComputerSystem.

Table 71 - CIM Elements for Virtual Tape Library

Element Name	Requirement	Description
11.8.47 CIM_SystemDevice (VTL to MediaAccessDevice)	Mandatory	Associates MediaAccessDevice to the Virtual Tape Library ComputerSystem.
11.8.48 CIM_SystemPackaging (VTL CS to Virtual StorageMediaLocation)	Optional	Experimental. Associates StorageMediaLocations to the Virtual Library Computer System.
11.8.49 CIM_SystemSpecificCollection (Unassigned)	Optional	Collection of unassigned virtual tapes.
11.8.50 SNIA_PhysicalTape (Virtual Tape)	Mandatory	A virtual tape in a Virtual Tape Library.
11.8.51 SNIA_VirtualTapeLibraryCapabilities	Optional	Capabilities related to virtual tapes.
11.8.52 SNIA_VirtualTapeLibraryConfigurationService	Optional	Service used to create Virtual Tape Libraries.
11.8.53 SNIA_VirtualTapeLibrarySetting	Optional	Settings used to create the Virtual Tape Library.
11.8.54 SNIA_VirtualTapeLibrarySystemCapabilities	Optional	Features supported in VirtualTapeLibrarySystemService.
11.8.55 SNIA_VirtualTapeLibrarySystemService	Optional	Service used to set up the Virtual Library System hardware.
11.8.56 SNIA_VirtualTapeService	Optional	Services used to create virtual tapes.
11.8.57 SNIA_VirtualTapeServiceCapabilities	Mandatory	Capabilities associated with VirtualTapeService.
11.8.58 SNIA_VirtualTapeSetting	Optional	Settings for virtual tapes.
SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_ComputerSystem AND ANY SourceInstance.CIM_ComputerSystem::Dedicated[*] = 34	Mandatory	CQL -Virtual Tape Library was created.
SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_PhysicalTape	Mandatory	Virtual Tape was created.
SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_MediaAccessDevice	Mandatory	Virtual Tape Drive was created.
SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_ProtocolControllerForUnit	Mandatory	
SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_ComputerSystem AND ANY SourceInstance.CIM_ComputerSystem::Dedicated[*] = 34	Mandatory	CQL -Virtual Tape Library was deleted.

Table 71 - CIM Elements for Virtual Tape Library

Element Name	Requirement	Description
SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_PhysicalTape	Mandatory	Virtual Tape was deleted.
SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_MediaAccessDevice	Mandatory	Virtual Tape drive was deleted from a Virtual Tape Library.
SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_ProtocolControllerForUnit	Mandatory	
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ComputerSystem AND ANY SourceInstance.CIM_ComputerSystem::Dedicated[*] = 34 AND SourceIsntance.CIM_ComputerSystem::OperationalStatus <> PreviousInstance.CIM_ComputerSystem::OperationalStatus	Mandatory	CQL -Status of a Virtual Tape Library has changed.
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ComputerSystem AND ANY SourceInstance.CIM_ComputerSystem::Dedicated[*] = 35 AND SourceInstance.CIM_ComputerSystem::OperationalStatus <> PreviousInstance.CIM_ComputerSystem::OperationalStatus	Mandatory	CQL -Status of a Virtual Library System has changed.
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ComputerSystem AND ANY SourceInstance.CIM_ComputerSystem::Dedicated[*] = 22 AND SourceInstance.CIM_ComputerSystem::OperationalStatus <> PreviousInstance.CIM_ComputerSystem::OperationalStatus	Mandatory	CQL -Status of an attached Media Library system has changed.
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_PhysicalTape	Mandatory	
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA SNIA_VirtualTapeLibrarySetting	Mandatory	Experimental. Indication that will identify when the settings of a Virtual Tape Library have changed.

11.8.1 CIM_AllocatedFromStoragePool (Pool from Concrete Pool)

Pool allocated from Concrete Pool.

Created By: Static

Modified By: Static
Deleted By: Static
Requirement: Optional

Table 72 describes class CIM_AllocatedFromStoragePool (Pool from Concrete Pool).

Table 72 - SMI Referenced Properties/Methods for CIM_AllocatedFromStoragePool (Pool from Concrete Pool)

Properties	Flags	Requirement	Description & Notes
SpaceConsumed		Mandatory	The space consumed (in bytes) from the pool referenced by Antecedent.
Antecedent		Mandatory	References to the parent pool from which the dependent pool is allocated.
Dependent		Mandatory	Reference to the child StoragePool.

11.8.2 CIM_AllocatedFromStoragePool (Pool from Primordial Pool)

Pool allocated from Primordial Pool.

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

Table 73 describes class CIM_AllocatedFromStoragePool (Pool from Primordial Pool).

Table 73 - SMI Referenced Properties/Methods for CIM_AllocatedFromStoragePool (Pool from Primordial Pool)

Properties	Flags	Requirement	Description & Notes
SpaceConsumed		Mandatory	The space consumed (in bytes) from the pool referenced by Antecedent.
Antecedent		Mandatory	Reference to the parent pool from which the dependent pool is allocated.
Dependent		Mandatory	Reference to the child, concrete StoragePool.

11.8.3 CIM_AllocatedFromStoragePool (StorageExtent from Concrete Pool)

StorageExtent from Concrete Pool.

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

Table 74 describes class CIM_AllocatedFromStoragePool (StorageExtent from Concrete Pool).

Table 74 - SMI Referenced Properties/Methods for CIM_AllocatedFromStoragePool (StorageExtent from Concrete Pool)

Properties	Flags	Requirement	Description & Notes
SpaceConsumed		Mandatory	The space consumed (in bytes) from the pool.
Antecedent		Mandatory	Reference to the parent pool from which the extent is allocated.
Dependent		Mandatory	Reference to StorageExtent.

11.8.4 CIM_ChangerDevice

Media changer for a Virtual Tape Library.

Created By: Extrinsic: CreateLibrary
Modified By: Extrinsic: ModifyLibrary
Deleted By: Extrinsic: DeleteLibrary
Requirement: Optional

Table 75 describes class CIM_ChangerDevice.

Table 75 - SMI Referenced Properties/Methods for CIM_ChangerDevice

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	Key. The CreationClassName of the associated Virtual Tape Library ComputerSystem (dedicated=3,34).
SystemName		Mandatory	Key. The Name of the associated Virtual Tape Library ComputerSystem (dedicated=3,34).
CreationClassName		Mandatory	Key. The name of the class used in creation of the instance.
DeviceID		Mandatory	Key.

11.8.5 CIM_Chassis (Virtual Tape Library)

Aggregates the virtual locations for tapes and drives of one Virtual Tape Library.

Created By: Extrinsic: CreateLibrary
Modified By: Extrinsic: ModifyLibrary
Deleted By: Extrinsic: DeleteLibrary
Requirement: Optional

Table 76 describes class CIM_Chassis (Virtual Tape Library).

Table 76 - SMI Referenced Properties/Methods for CIM_Chassis (Virtual Tape Library)

Properties	Flags	Requirement	Description & Notes
Tag		Mandatory	Key.
CreationClassName		Mandatory	Key. The name of the class used in creation of the instance.
PackageType		Mandatory	Shall be 3 (ChassisFrame).
ChassisPackageType		Mandatory	
Manufacturer		Optional	
Model		Optional	
SerialNumber		Optional	
PartNumber		Optional	
SKU		Optional	
VendorCompatibilityStrings		Optional	
ElementName		Optional	

11.8.6 CIM_ComputerSystem (Virtual Library System)

ComputerSystem that represents the Virtual Library System - the system that creates and manages Virtual Tape Library systems.

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 'Virtual Tape Library', RegisteredOrganization set to 'SNIA', and RegisteredVersion set to '1.6.0'.

Table 77 describes class CIM_ComputerSystem (Virtual Library System).

Table 77 - SMI Referenced Properties/Methods for CIM_ComputerSystem (Virtual Library System)

Properties	Flags	Requirement	Description & Notes
CreationClassName		Mandatory	Key. The name of the class used in creation of the instance.
Name		Mandatory	Key. Unique identifier for the Virtual Library System.
Dedicated		Mandatory	Indicates that this computer system is dedicated to operation as a Virtual Library System. Shall include 3(Storage) and 35(Virtual Library System).

Table 77 - SMI Referenced Properties/Methods for CIM_ComputerSystem (Virtual Library System)

Properties	Flags	Requirement	Description & Notes
NameFormat		Mandatory	Format for Name property. Shall be 'HID' (Hardware Identifier).
OperationalStatus		Mandatory	Shall be 0 (Unknown), 2 (Okay), 3 (Degraded), 4 Stressed), 6 (Error), 7 (Non-recoverable Error), 9 (Starting), 10 (Stopping), or 11 (Stopped).
StatusDescriptions		Optional	Additional information related to the values in OperationalStatus.
ElementName	M	Optional	User friendly name.
PrimaryOwnerContact	M	Optional	Contact details for Virtual Library System owner.
PrimaryOwnerName	M	Optional	Owner of the Virtual Library System.
OtherIdentifyingInfo		Optional	Other data that could be used to identify the Virtual Library System.
IdentifyingDescriptions		Optional	Provides explanations and details for the entries in the OtherIdentifyingInfo property.

11.8.7 CIM_ComputerSystem (Virtual Tape Library)

ComputerSystem that represents a Virtual Tape Library.

Created By: Extrinsic: CreateLibrary
Modified By: Extrinsic: ModifyLibrary
Deleted By: Extrinsic: DeleteLibrary
Requirement: Optional

Table 78 describes class CIM_ComputerSystem (Virtual Tape Library).

Table 78 - SMI Referenced Properties/Methods for CIM_ComputerSystem (Virtual Tape Library)

Properties	Flags	Requirement	Description & Notes
CreationClassName		Mandatory	Key. The name of the class used in creation of the instance.
Name		Mandatory	Key. Unique identifier for the virtual tape library. This should take the form of a string consisting of Vendor+Product+SerialNumber.
Dedicated		Mandatory	Indicates that this computer system is dedicated to operation as a Virtual Tape Library. Shall include 3 (Storage) and 34 (Virtual Tape Library).
NameFormat		Mandatory	SHall be 'HID'.
OperationalStatus		Mandatory	Shall be 0 (Unknown), 2 (Okay), 3 (Degraded), 4 Stressed), 6 (Error), 7 (Non-recoverable Error), 9 (Starting), 10 (Stopping), or 11 (Stopped).

Table 78 - SMI Referenced Properties/Methods for CIM_ComputerSystem (Virtual Tape Library)

Properties	Flags	Requirement	Description & Notes
StatusDescriptions		Optional	Additional information related to the values in OperationalStatus.
ElementName		Mandatory	User friendly name.
PrimaryOwnerContact	M	Optional	Contact details for Virtual Tape Library owner.
PrimaryOwnerName	M	Optional	Owner of the Virtual Tape Library.
OtherIdentifyingInfo		Optional	Other data that could be used to identify the Virtual Tape Library.
IdentifyingDescriptions		Optional	Provides explanations and details for the entries in the OtherIdentifyingInfo property.

11.8.8 CIM_ComputerSystemPackage

Associates Chassis to the Virtual Tape Library ComputerSystem.

Created By: Extrinsic: CreateLibrary
Modified By: Extrinsic: ModifyLibrary
Deleted By: Extrinsic: DeleteLibrary
Requirement: Mandatory

Table 79 describes class CIM_ComputerSystemPackage.

Table 79 - SMI Referenced Properties/Methods for CIM_ComputerSystemPackage

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	Reference to Chassis.
Dependent		Mandatory	Reference to Virtual Tape Library ComputerSystem.

11.8.9 CIM_ConcreteComponent (StorageExtent from Primordial Pool)

Associates Primordial StoragePools and StorageExtents imported from arrays or the Disk Drive Lite profile.

Created By: Static
Modified By: Static
Deleted By: Static
Requirement: Optional

Table 80 describes class CIM_ConcreteComponent (StorageExtent from Primordial Pool).

Table 80 - SMI Referenced Properties/Methods for CIM_ConcreteComponent (StorageExtent from Primordial Pool)

Properties	Flags	Requirement	Description & Notes
GroupComponent		Mandatory	Reference to the primordial pool.
PartComponent		Mandatory	Reference to imported StorageExtent.

11.8.10 CIM_ConcreteDependency (Virtual Library System to MediaLibrary)

Associates the Virtual Library System ComputerSystem object to a Media Library ComputerSystem objects.

Created By: Static
Modified By: Static
Deleted By: Static
Requirement: Support for SML profile.

Table 81 describes class CIM_ConcreteDependency (Virtual Library System to MediaLibrary).

Table 81 - SMI Referenced Properties/Methods for CIM_ConcreteDependency (Virtual Library System to MediaLibrary)

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	Reference to Virtual Library System ComputerSystem.
Dependent		Mandatory	Reference to Storage Library ComputerSystem.

11.8.11 CIM_Container (Chassis to StorageMediaLocations)

Associates StorageMediaLocations to the chassis.

Created By: Extrinsic: CreateLibrary
Modified By: Extrinsic: ModifyLibrary
Deleted By: Extrinsic: DeleteLibrary
Requirement: Mandatory

Table 82 describes class CIM_Container (Chassis to StorageMediaLocations).

Table 82 - SMI Referenced Properties/Methods for CIM_Container (Chassis to StorageMediaLocations)

Properties	Flags	Requirement	Description & Notes
GroupComponent		Mandatory	Reference to Chassis.
PartComponent		Mandatory	Reference to StorageMediaLocation.

11.8.12 CIM_ElementCapabilities (ImplementationCapabilities to Virtual Library System)

Experimental. Associates the conformant Virtual Library System ComputerSystem to the CIM_ImplementationCapabilities supported by the implementation.

Created By: Static
Modified By: Static
Deleted By: Static
Requirement: Optional

Table 83 describes class CIM_ElementCapabilities (ImplementationCapabilities to Virtual Library System).

Table 83 - SMI Referenced Properties/Methods for CIM_ElementCapabilities (ImplementationCapabilities to Virtual Library System)

Properties	Flags	Requirement	Description & Notes
Capabilities		Mandatory	The ImplementationCapabilities.
ManagedElement		Mandatory	The conformant Virtual Library System ComputerSystem that has ImplementationCapabilities.

11.8.13 CIM_ElementCapabilities (ImplementationCapabilities to Virtual Tape Library System)

Experimental. Associates a Virtual Tape Library ComputerSystem to the CIM_ImplementationCapabilities supported by the implementation.

Created By: Static
Modified By: Static
Deleted By: Static
Requirement: Optional

Table 84 describes class CIM_ElementCapabilities (ImplementationCapabilities to Virtual Tape Library System).

Table 84 - SMI Referenced Properties/Methods for CIM_ElementCapabilities (ImplementationCapabilities to Virtual Tape Library System)

Properties	Flags	Requirement	Description & Notes
Capabilities		Mandatory	The ImplementationCapabilities.
ManagedElement		Mandatory	A Virtual Tape Library ComputerSystem that has ImplementationCapabilities.

11.8.14 CIM_ElementCapabilities (Virtual Tape Library Capabilities)

Assocates VirtualTapeLibraryCapabilities and Virtual Library System ComputerSystem.

Created By: Static
Modified By: Static
Deleted By: Static
Requirement: Optional

Table 85 describes class CIM_ElementCapabilities (Virtual Tape Library Capabilities).

Table 85 - SMI Referenced Properties/Methods for CIM_ElementCapabilities (Virtual Tape Library Capabilities)

Properties	Flags	Requirement	Description & Notes
Capabilities		Mandatory	Reference to VirtualTapeLibraryCapabilities.
ManagedElement		Mandatory	Reference to Virtual Library System ComputerSystem.

11.8.15 CIM_ElementCapabilities (Virtual Tape Library System Capabilities)

Associates VirtualTapeLibraryCapabilities and Virtual Library System ComputerSystem.

Created By: Static
Modified By: Static
Deleted By: Static
Requirement: Optional

Table 86 describes class CIM_ElementCapabilities (Virtual Tape Library System Capabilities).

Table 86 - SMI Referenced Properties/Methods for CIM_ElementCapabilities (Virtual Tape Library System Capabilities)

Properties	Flags	Requirement	Description & Notes
Capabilities		Mandatory	Reference to VirtualTapeLibrarySystemCapabilities.
ManagedElement		Mandatory	Reference to the Virtual Library System.

11.8.16 CIM_ElementCapabilities (Virtual Tape Service Capabilities)

Associates VirtualTapeServiceCapabilities and VirtualTapeService.

Created By: Static
Modified By: Static
Deleted By: Static
Requirement: Optional

Table 87 describes class CIM_ElementCapabilities (Virtual Tape Service Capabilities).

Table 87 - SMI Referenced Properties/Methods for CIM_ElementCapabilities (Virtual Tape Service Capabilities)

Properties	Flags	Requirement	Description & Notes
Capabilities		Mandatory	Reference to VirtualTapeServiceCapabilities.
ManagedElement		Mandatory	Reference to VirtualTapeService.

11.8.17 CIM_ElementSettingData (Physical Tape)

Associates PhysicalTape and VirtualTapeSetting.

Requirement: Optional

Table 88 describes class CIM_ElementSettingData (Physical Tape).

Table 88 - SMI Referenced Properties/Methods for CIM_ElementSettingData (Physical Tape)

Properties	Flags	Requirement	Description & Notes
ManagedElement		Mandatory	Reference to PhysicalTape.
SettingData		Mandatory	Reference to VirtualTapeSetting.

11.8.18 CIM_HostedCollection

Associates the Virtual Library System ComputerSystem and the unassigned tapes SystemSpecificCollection.

Requirement: Optional

Table 89 describes class CIM_HostedCollection.

Table 89 - SMI Referenced Properties/Methods for CIM_HostedCollection

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	Reference to Virtual Library System ComputerSystem (Dedicated = 3,35).
Dependent		Mandatory	Reference to SystemSpecificCollection.

11.8.19 CIM_HostedDependency (Virtual Library System to VirtualLibrary)

Associates the Virtual Library System ComputerSystem object to the Virtual Tape Library ComputerSystem objects.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 90 describes class CIM_HostedDependency (Virtual Library System to VirtualLibrary).

Table 90 - SMI Referenced Properties/Methods for CIM_HostedDependency (Virtual Library System to VirtualLibrary)

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	Reference to Virtual Library System ComputerSystem.
Dependent		Mandatory	Reference to Virtual Tape Library ComputerSystem.

11.8.20 CIM_HostedService (Virtual Tape Library Configuration Service)

Associates the VirtualTapeLibraryConfigurationService to the ComputerSystem representing the Virtual Library System.

Created By: Static

Modified By: Static
Deleted By: Static
Requirement: Optional

Table 91 describes class CIM_HostedService (Virtual Tape Library Configuration Service).

Table 91 - SMI Referenced Properties/Methods for CIM_HostedService (Virtual Tape Library Configuration Service)

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	The reference to the ComputerSystem representing the Virtual Library System.
Dependent		Mandatory	The reference to the VirtualTapeLibraryConfigurationService.

11.8.21 CIM_HostedService (Virtual Tape Library System Service)

Associates the VirtualTapeLibrarySystemService to the ComputerSystem representing the Virtual Library System.

Created By: Static
Modified By: Static
Deleted By: Static
Requirement: Optional

Table 92 describes class CIM_HostedService (Virtual Tape Library System Service).

Table 92 - SMI Referenced Properties/Methods for CIM_HostedService (Virtual Tape Library System Service)

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	The reference to the ComputerSystem representing the Virtual Library System.
Dependent		Mandatory	The reference to the VirtualTapeLibrarySystemService.

11.8.22 CIM_HostedService (Virtual Tape Service)

Associates VirtualTapeService to the ComputerSystem representing the Virtual Library System.

Created By: Static
Modified By: Static
Deleted By: Static
Requirement: Optional

Table 93 describes class CIM_HostedService (Virtual Tape Service).

Table 93 - SMI Referenced Properties/Methods for CIM_HostedService (Virtual Tape Service)

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	The reference to the ComputerSystem representing the Virtual Tape Library.
Dependent		Mandatory	The reference to the VirtualTapeService.

11.8.23 CIM_HostedStoragePool (Primordial)

Associates Virtual Library System ComputerSystem and StoragePools.

Requirement: Mandatory

Table 94 describes class CIM_HostedStoragePool (Primordial).

Table 94 - SMI Referenced Properties/Methods for CIM_HostedStoragePool (Primordial)

Properties	Flags	Requirement	Description & Notes
GroupComponent		Mandatory	Reference to ComputerSystem.
PartComponent		Mandatory	Reference to StoragePool.

11.8.24 CIM_ImplementationCapabilities (ImplementationCapabilities)

Experimental. The capabilities (features) of the profile implementation.

Created By: Static
Modified By: Static
Deleted By: Static
Requirement: Optional

Table 95 describes class CIM_ImplementationCapabilities (ImplementationCapabilities).

Table 95 - SMI Referenced Properties/Methods for CIM_ImplementationCapabilities (ImplementationCapabilities)

Properties	Flags	Requirement	Description & Notes
InstanceID		Mandatory	An opaque, unique id for the implementation capability of an implementation.
ElementName		Optional	A provider supplied user-friendly name for this CIM_ImplementationCapabilities element.

Table 95 - SMI Referenced Properties/Methods for CIM_ImplementationCapabilities (ImplementationCapabilities)

Properties	Flags	Requirement	Description & Notes
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".

11.8.25 CIM_LimitedAccessPort

The tape export port for a Virtual Tape Library.

Created By: Extrinsic: CreateLibrary
Modified By: Extrinsic: ModifyLibrary
Deleted By: Extrinsic: DeleteLibrary
Requirement: Optional

Table 96 describes class CIM_LimitedAccessPort.

Table 96 - SMI Referenced Properties/Methods for CIM_LimitedAccessPort

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	Key. The CreationClassName of the associated Virtual Tape Library ComputerSystem (dedicated=3,34).
SystemName		Mandatory	Key. The Name of the associated Virtual Tape Library ComputerSystem (dedicated=3,34).
CreationClassName		Mandatory	Key. The name of the class used in creation of the instance.
DeviceID		Mandatory	Key.
OperationalStatus		Optional	Shall be 0 (Unknown), 2 (Okay), 3 (Degraded), 4 Stressed), 6 (Error), 7 (Non-recoverable Error), 9 (Starting), 10 (Stopping), or 11 (Stopped).
StatusDescriptions		Optional	Additional information related to the values in OperationalStatus.

11.8.26 CIM_LogicalIdentity

Requirement: Mandatory

Table 97 describes class CIM_LogicalIdentity.

Table 97 - SMI Referenced Properties/Methods for CIM_LogicalIdentity

Properties	Flags	Requirement	Description & Notes
SystemElement		Mandatory	Reference to VTL PhysicalTape.
SameElement		Mandatory	

11.8.27 CIM_MediaAccessDevice

The tape drive for a Virtual Tape Library.

Created By: Extrinsic: CreateLibrary
Modified By: Extrinsic: ModifyLibrary
Deleted By: Extrinsic: DeleteLibrary
Requirement: Mandatory

Table 98 describes class CIM_MediaAccessDevice.

Table 98 - SMI Referenced Properties/Methods for CIM_MediaAccessDevice

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	Key. The CreationClassName of the associated Virtual Tape Library ComputerSystem (dedicated=3,34).
CreationClassName		Mandatory	Key. The name of the class used in creation of the instance.
SystemName		Mandatory	Key. The Name of the associated Virtual Tape Library ComputerSystem (dedicated=3,34).
DeviceID		Mandatory	Key.
OperationalStatus		Optional	Shall be 0 (Unknown), 2 (Okay), 3 (Degraded), 4 Stressed), 6 (Error), 7 (Non-recoverable Error), 9 (Starting), 10 (Stopping), or 11 (Stopped).
StatusDescriptions		Optional	Additional information related to the values in OperationalStatus.
NeedsCleaning		Optional	Shall be false for virtual drives.
MountCount		Optional	

11.8.28 CIM_MemberOfCollection

Associates the unassigned tapes SystemSpecificCollection and PhysicalTape.

Requirement: Optional

Table 99 describes class CIM_MemberOfCollection.

Table 99 - SMI Referenced Properties/Methods for CIM_MemberOfCollection

Properties	Flags	Requirement	Description & Notes
Collection		Mandatory	Reference to SystemSpecificCollection.
Member		Mandatory	Reference to Unassigned PhysicalTape.

11.8.29 CIM_PhysicalMediaInLocation

Associates tape (PhysicalTape) to media slots (StorageMediaLocation).

Created By: Extrinsic: CreateLibrary
Modified By: Extrinsic: ModifyLibrary
Deleted By: Extrinsic: DeleteLibrary
Requirement: Optional

Table 100 describes class CIM_PhysicalMediaInLocation.

Table 100 - SMI Referenced Properties/Methods for CIM_PhysicalMediaInLocation

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	Reference to StorageMediaLocation.
Dependent		Mandatory	Reference to PhysicalTape.

11.8.30 CIM_Product

Asset information for the system.

Requirement: Optional

Table 101 describes class CIM_Product.

Table 101 - SMI Referenced Properties/Methods for CIM_Product

Properties	Flags	Requirement	Description & Notes
ElementName	M	Optional	
Name		Mandatory	Key.
IdentifyingNumber		Mandatory	Key.
Vendor		Mandatory	Key.
Version		Mandatory	Key.

11.8.31 CIM_ProductElementComponent (Virtual Tape Library)

Associates a Virtual Library System or Virtual Tape Library ComputerSystem with Product.

Requirement: Optional

Table 102 describes class CIM_ProductElementComponent (Virtual Tape Library).

Table 102 - SMI Referenced Properties/Methods for CIM_ProductElementComponent (Virtual Tape Library)

Properties	Flags	Requirement	Description & Notes
GroupComponent		Mandatory	Reference to CIM_Product.
PartComponent		Mandatory	Reference to ComputerSystem.

11.8.32 CIM_Realizes (Slots to Changers)

Associates changers to the StorageMediaLocations.

Created By: Extrinsic: CreateLibrary
Modified By: Extrinsic: ModifyLibrary
Deleted By: Extrinsic: DeleteLibrary
Requirement: Optional

Table 103 describes class CIM_Realizes (Slots to Changers).

Table 103 - SMI Referenced Properties/Methods for CIM_Realizes (Slots to Changers)

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	Reference to StorageMediaLocation.
Dependent		Mandatory	Reference to ChangerDevice.

11.8.33 CIM_Realizes (Slots to Ports)

Associates access ports to the StorageMediaLocations.

Created By: Extrinsic: CreateLibrary
Modified By: Extrinsic: ModifyLibrary
Deleted By: Extrinsic: DeleteLibrary
Requirement: Optional

Table 104 describes class CIM_Realizes (Slots to Ports).

Table 104 - SMI Referenced Properties/Methods for CIM_Realizes (Slots to Ports)

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	Reference to StorageMediaLocation.
Dependent		Mandatory	Reference to LimitedAccessPort.

11.8.34 CIM_Realizes (Slots to TapeDrive)

Associates drives to the StorageMediaLocations.

Created By: Extrinsic: CreateLibrary
Modified By: Extrinsic: ModifyLibrary

Deleted By: Extrinsic: DeleteLibrary
Requirement: Mandatory

Table 105 describes class CIM_Realizes (Slots to TapeDrive).

Table 105 - SMI Referenced Properties/Methods for CIM_Realizes (Slots to TapeDrive)

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	Reference to StorageMediaLocation.
Dependent		Mandatory	Reference to MediaAccessDevice.

11.8.35 CIM_ServiceAffectsElement

Associates Virtual Tape Library Computer System and VirtualTapeLibraryConfigurationService.

Requirement: Mandatory

Table 106 describes class CIM_ServiceAffectsElement.

Table 106 - SMI Referenced Properties/Methods for CIM_ServiceAffectsElement

Properties	Flags	Requirement	Description & Notes
AffectedElement		Mandatory	Reference to the Virtual Tape Library ComputerSystem (dedicated=3,34).
AffectingElement		Mandatory	Reference to VirtualTapeLibraryConfigurationService.

11.8.36 CIM_SettingAssociatedToCapabilities (Setting To Capabilities)

Associates VirtualTapeServiceCapabilities and VirtualTapeSetting.

Requirement: Mandatory

Table 107 describes class CIM_SettingAssociatedToCapabilities (Setting To Capabilities).

Table 107 - SMI Referenced Properties/Methods for CIM_SettingAssociatedToCapabilities (Setting To Capabilities)

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	Reference to VirtualTapeServiceCapabilities.
Dependent		Mandatory	Reference to VirtualTapeSetting.

11.8.37 CIM_SettingsDefineCapabilities

Assocates VirtualTapeLibraryCapabilities and VirtualTapeLibrarySetting.

Requirement: Mandatory

Table 108 describes class CIM_SettingsDefineCapabilities.

Table 108 - SMI Referenced Properties/Methods for CIM_SettingsDefineCapabilities

Properties	Flags	Requirement	Description & Notes
GroupComponent		Mandatory	Reference to VirtualTapeLibraryCapabilities.
PartComponent		Mandatory	Reference to VirtualTapeLibrarySetting.

11.8.38 CIM_SettingsDefineState

Requirement: Mandatory

Table 109 describes class CIM_SettingsDefineState.

Table 109 - SMI Referenced Properties/Methods for CIM_SettingsDefineState

Properties	Flags	Requirement	Description & Notes
ManagedElement		Mandatory	Reference to Virtual Tape Library ComputerSystem.
SettingData		Mandatory	Reference to VirtualTapeLibrarySetting.

11.8.39 CIM_StorageExtent (Assigned)

Capacity allocated from an underlying array providing capacity for a virtual tape.

Created By: Extrinsic: CreateTapeFromPool
Modified By: Extrinsic
Deleted By: Extrinsic
Requirement: Optional

Table 110 describes class CIM_StorageExtent (Assigned).

Table 110 - SMI Referenced Properties/Methods for CIM_StorageExtent (Assigned)

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	Key. The CreationClassName of the associated Virtual Library System ComputerSystem (dedicated=3,35).
SystemName		Mandatory	Key. The Name of the associated Virtual Library System ComputerSystem (dedicated=3,35).
CreationClassName		Mandatory	Key. The name of the class used in creation of the instance.
DeviceID		Mandatory	Key.
ExtentDiscriminator		Mandatory	Shall be 'SNIA:Allocated'.
DataOrganization		Mandatory	Fixed Block (disk) or Variable Block (tape).
Primordial		Mandatory	Shall be 'false'.
NumberOfBlocks		Mandatory	Used with block size.

Table 110 - SMI Referenced Properties/Methods for CIM_StorageExtent (Assigned)

Properties	Flags	Requirement	Description & Notes
BlockSize		Mandatory	
ExtentStatus		Mandatory	
OperationalStatus		Mandatory	

11.8.40 CIM_StorageExtent (Imported)

A LUN allocated from an underlying array.

Created By: External
Modified By: External
Deleted By: External
Requirement: Optional

Table 111 describes class CIM_StorageExtent (Imported).

Table 111 - SMI Referenced Properties/Methods for CIM_StorageExtent (Imported)

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	Key. The CreationClassName of the associated Virtual Library System ComputerSystem (dedicated=3,35).
SystemName		Mandatory	Key. The Name of the associated Virtual Library System ComputerSystem (dedicated=3,35).
CreationClassName		Mandatory	Key. The name of the class used in creation of the instance.
DeviceID		Mandatory	Key.
Name	CD	Mandatory	Identifier for this volume; based of datapath standards such as SCSI or ATAPI.

Table 111 - SMI Referenced Properties/Methods for CIM_StorageExtent (Imported)

Properties	Flags	Requirement	Description & Notes
NameFormat		Mandatory	The type of identifier in the Name property. The valid values for StorageVolumes are: 1 (Other) 2 (VPD83NAA6) 3 (VPD83NAA5) 4 (VPD83Type2) 5 (VPD83Type1) 6 (VPD83Type0) 7 (SNVM) 8 (NodeWWN) 9 (NAA) 10 (EUI64) 11 (T10VID).
NameNamespace		Mandatory	The namespace that defines uniqueness for the NameFormat.
OtherIdentifyingInfo	CD	Optional	Additional correlatable names.
IdentifyingDescriptions		Optional	
ExtentDiscriminator		Mandatory	Shall includes 'SNIA:Imported' and 'SNIA:Pool Component'.
Primordial		Mandatory	Shall be 'true'.
NumberOfBlocks		Mandatory	The number of blocks of capacity consumed from the parent StoragePool.
BlockSize		Mandatory	Block Size in bytes.
ExtentStatus		Mandatory	
OperationalStatus		Mandatory	

11.8.41 CIM_StorageMediaLocation

The tape and drive locations in a virtual tape library.

Created By: Extrinsic: CreateLibrary
Modified By: Extrinsic: ModifyLibrary
Deleted By: Extrinsic: DeleteLibrary
Requirement: Optional

Table 112 describes class CIM_StorageMediaLocation.

Table 112 - SMI Referenced Properties/Methods for CIM_StorageMediaLocation

Properties	Flags	Requirement	Description & Notes
CreationClassName		Mandatory	Key. The name of the class used in creation of the instance.
Tag		Mandatory	Key.
LocationType		Mandatory	Shall be 2(Slot), 4(MediaAccessDevice), or 6(Limited Access Port).
LocationCoordinates		Mandatory	
MediaTypesSupported		Mandatory	
MediaCapacity		Mandatory	

11.8.42 CIM_StoragePool (Concrete)

The concrete StoragePool. A concrete StoragePool shall be allocated from the Primordial StoragePool. It shall be used for allocating virtual tapes.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 113 describes class CIM_StoragePool (Concrete).

Table 113 - SMI Referenced Properties/Methods for CIM_StoragePool (Concrete)

Properties	Flags	Requirement	Description & Notes
Primordial		Mandatory	Shall be false.
InstanceID		Mandatory	
ElementName		Optional	
PoolID		Mandatory	A unique name in the context of this system that identifies this Pool.
TotalManagedSpace		Mandatory	
RemainingManagedSpace		Mandatory	

11.8.43 CIM_StoragePool (Primordial)

The pool of all storage available from the backing store.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 114 describes class CIM_StoragePool (Primordial).

Table 114 - SMI Referenced Properties/Methods for CIM_StoragePool (Primordial)

Properties	Flags	Requirement	Description & Notes
InstanceID		Mandatory	
PoolID		Mandatory	A unique name in the context of this system that identifies this Pool.
Primordial		Mandatory	Shall be 'true'.
TotalManagedSpace		Mandatory	
RemainingManaged Space		Mandatory	
ElementName		Optional	

11.8.44 CIM_SystemDevice (System to Primordial StorageExtent)

Associates StorageExtents to the Virtual Library System.

Created By: Static
Modified By: Static
Deleted By: Static
Requirement: Optional

Table 115 describes class CIM_SystemDevice (System to Primordial StorageExtent).

Table 115 - SMI Referenced Properties/Methods for CIM_SystemDevice (System to Primordial StorageExtent)

Properties	Flags	Requirement	Description & Notes
PartComponent		Mandatory	Reference to StorageExtent.
GroupComponent		Mandatory	Reference to Virtual Library System ComputerSystem.

11.8.45 CIM_SystemDevice (VTL to ChangerDevice)

Associates ChangerDevice to the Virtual Tape Library ComputerSystem.

Created By: Extrinsic: CreateLibrary
Modified By: Extrinsic: ModifyLibrary
Deleted By: Extrinsic: DeleteLibrary
Requirement: Optional

Table 116 describes class CIM_SystemDevice (VTL to ChangerDevice).

Table 116 - SMI Referenced Properties/Methods for CIM_SystemDevice (VTL to ChangerDevice)

Properties	Flags	Requirement	Description & Notes
PartComponent		Mandatory	Reference to ChangerDevice.
GroupComponent		Mandatory	Reference to Virtual Tape Library ComputerSystem.

11.8.46 CIM_SystemDevice (VTL to LimitedAccessPort)

Associates LimitedAccessDevice to the Virtual Tape Library ComputerSystem.

Created By: Extrinsic: CreateLibrary
Modified By: Extrinsic: ModifyLibrary
Deleted By: Extrinsic: DeleteLibrary
Requirement: Optional

Table 117 describes class CIM_SystemDevice (VTL to LimitedAccessPort).

Table 117 - SMI Referenced Properties/Methods for CIM_SystemDevice (VTL to LimitedAccessPort)

Properties	Flags	Requirement	Description & Notes
PartComponent		Mandatory	Reference to LimitredAccessPort.
GroupComponent		Mandatory	Reference to Virtual Tape Library ComputerSystem.

11.8.47 CIM_SystemDevice (VTL to MediaAccessDevice)

Associates MediaAccessDevice to the Virtual Tape Library ComputerSystem.

Created By: Extrinsic: CreateLibrary
Modified By: Extrinsic: ModifyLibrary
Deleted By: Extrinsic: DeleteLibrary
Requirement: Mandatory

Table 118 describes class CIM_SystemDevice (VTL to MediaAccessDevice).

Table 118 - SMI Referenced Properties/Methods for CIM_SystemDevice (VTL to MediaAccessDevice)

Properties	Flags	Requirement	Description & Notes
PartComponent		Mandatory	Reference to MediaAccessDevice.
GroupComponent		Mandatory	Reference to Virtual Tape Library ComputerSystem.

11.8.48 CIM_SystemPackaging (VTL CS to Virtual StorageMediaLocation)

Experimental. Associates StorageMediaLocations to the Virtual Tape Library Computer System.

Created By: Static

Modified By: Static
Deleted By: Static
Requirement: Optional

Table 119 describes class CIM_SystemPackaging (VTL CS to Virtual StorageMediaLocation).

Table 119 - SMI Referenced Properties/Methods for CIM_SystemPackaging (VTL CS to Virtual StorageMediaLocation)

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	Reference to StorageMediaLocation.
Dependent		Mandatory	Reference to Virtual Library ComputerSystem.

11.8.49 CIM_SystemSpecificCollection (Unassigned)

Collection of unassigned virtual tapes.

Created By: Extrinsic
Modified By: Extrinsic
Deleted By: Extrinsic
Requirement: Optional

Table 120 describes class CIM_SystemSpecificCollection (Unassigned).

Table 120 - SMI Referenced Properties/Methods for CIM_SystemSpecificCollection (Unassigned)

Properties	Flags	Requirement	Description & Notes
InstanceID		Mandatory	Key.
ElementName		Optional	

11.8.50 SNIA_PhysicalTape (Virtual Tape)

A virtual tape in a Virtual Tape Library.

Created By: Extrinsic: CreateTapeFromPool
Modified By: Extrinsic
Deleted By: Extrinsics: DeleteLibrary, MoveMedia
Requirement: Mandatory

Table 121 describes class SNIA_PhysicalTape (Virtual Tape).

Table 121 - SMI Referenced Properties/Methods for SNIA_PhysicalTape (Virtual Tape)

Properties	Flags	Requirement	Description & Notes
Tag		Mandatory	Key.
CreationClassName		Mandatory	Key. The name of the class used in creation of the instance.

Table 121 - SMI Referenced Properties/Methods for SNIA_PhysicalTape (Virtual Tape)

Properties	Flags	Requirement	Description & Notes
IsBasedOnDisk		Mandatory	Shall be 'true'.
Usage		Optional	

11.8.51 SNIA_VirtualTapeLibraryCapabilities

Capabilities related to Virtual tapes.

Created By: Static
Modified By: Static
Deleted By: Static
Requirement: Optional

Table 122 describes class SNIA_VirtualTapeLibraryCapabilities.

Table 122 - SMI Referenced Properties/Methods for SNIA_VirtualTapeLibraryCapabilities

Properties	Flags	Requirement	Description & Notes
InstanceID		Mandatory	Key.
LibraryTypesSupported		Mandatory	
MaxVTLsSupported		Optional	
MaxDrivesSupported		Optional	
MaxAccessPortsSupported		Optional	
IsThinTapeSupported		Mandatory	

11.8.52 SNIA_VirtualTapeLibraryConfigurationService

Service used to create Virtual Tape Libraries.

Created By: Static
Modified By: Static
Deleted By: Static
Requirement: Optional

Table 123 describes class SNIA_VirtualTapeLibraryConfigurationService.

Table 123 - SMI Referenced Properties/Methods for SNIA_VirtualTapeLibraryConfigurationService

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	Key. The CreationClassName of the associated Virtual Library System ComputerSystem (dedicated=3,35).
CreationClassName		Mandatory	Key. The name of the class used in creation of the instance.
SystemName		Mandatory	Key. The Name of the associated Virtual Tape Library ComputerSystem (dedicated=3,34).
Name		Mandatory	Key.
CreateVTL()		Optional	Creates a new Virtual Library.
ModifyVTL()		Optional	Modifies the configurable settings of a Virtual Library.
DeleteVTL()		Optional	Deletes a Virtual Library.

11.8.53 SNIA_VirtualTapeLibrarySetting

Settings used to create the Virtual Tape Library.

Created By: Static
Modified By: Static
Deleted By: Static
Requirement: Optional

Table 124 describes class SNIA_VirtualTapeLibrarySetting.

Table 124 - SMI Referenced Properties/Methods for SNIA_VirtualTapeLibrarySetting

Properties	Flags	Requirement	Description & Notes
InstanceID		Mandatory	Key.
LibraryType		Mandatory	
LibraryName		Optional	If present, this shall be the name of the Virtual Library associated with these settings.
SlotCount		Optional	If present, this shall be the number of slots in the Virtual Library associated with these settings. Shall include the number of locations for virtual tapes (occupied or empty), but not include the count of drives or access ports.
IsThinTape		Mandatory	
ThinTapeSize		Mandatory	
MaxTapeSize		Mandatory	
DriveType		Mandatory	

Table 124 - SMI Referenced Properties/Methods for SNIA_VirtualTapeLibrarySetting

Properties	Flags	Requirement	Description & Notes
DriveCount		Optional	If present, this shall be the number of drives in the Virtual Library associated with these settings.
DriveNames		Optional	
TapeBarcodeRange		Optional	
Modify		Optional	An array containing a list of variables that may be modified; used when modifying the settings of an existing Virtual Tape Library.

11.8.54 SNIA_VirtualTapeLibrarySystemCapabilities

Features supported in VirtualTapeLibrarySystemService.

Created By: Static
Modified By: Static
Deleted By: Static
Requirement: Optional

Table 125 describes class SNIA_VirtualTapeLibrarySystemCapabilities.

Table 125 - SMI Referenced Properties/Methods for SNIA_VirtualTapeLibrarySystemCapabilities

Properties	Flags	Requirement	Description & Notes
InstanceID		Mandatory	Key.
ConfigPort		Mandatory	
ExternalPhysicalLibrary		Mandatory	

11.8.55 SNIA_VirtualTapeLibrarySystemService

Service used to set up the Virtual Library System hardware.

Created By: Extrinsic: CreateLibrary
Modified By: Extrinsic: ModifyLibrary
Deleted By: Extrinsic: DeleteLibrary
Requirement: Optional

Table 126 describes class SNIA_VirtualTapeLibrarySystemService.

Table 126 - SMI Referenced Properties/Methods for SNIA_VirtualTapeLibrarySystemService

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	Key. The CreationClassName of the associated Virtual Library System ComputerSystem (dedicated=3,35).
CreationClassName		Mandatory	Key. The name of the class used in creation of the instance.

Table 126 - SMI Referenced Properties/Methods for SNIA_VirtualTapeLibrarySystemService

Properties	Flags	Requirement	Description & Notes
SystemName		Mandatory	Key. The Name of the associated Virtual Tape Library ComputerSystem (dedicated=3,34).
Name		Mandatory	Key.
SetPortUse()		Optional	Set the port to 'Initiator' or 'Target'.
ListPLibrary()		Optional	Finds the Physical Libraries connected to the Virtual Library System.
AttachPLibrary()		Optional	Allows a Physical Library to be used by the Virtual Library System for copy and export operations.
DetachPLibrary()		Optional	Removes a Physical Library from the context of a Virtual Library System, disallowing copy and export operations to that library.
RescanPhysicalHardware()		Optional	Scans for external Physical Libraries and Block Storage Arrays.

11.8.56 SNIA_VirtualTapeService

Services used to create virtual tapes.

Created By: Static
Modified By: Static
Deleted By: Static
Requirement: Optional

Table 127 describes class SNIA_VirtualTapeService.

Table 127 - SMI Referenced Properties/Methods for SNIA_VirtualTapeService

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	Key. The CreationClassName of the associated Virtual Library System ComputerSystem (dedicated=3,35).
CreationClassName		Mandatory	Key. The name of the class used in creation of the instance.
SystemName		Mandatory	Key. The Name of the associated Virtual Tape Library ComputerSystem (dedicated=3,34).
Name		Mandatory	Key.
CreateTapeFromPool()		Optional	Creates one or more virtual tapes in a Virtual Library System.
DeleteTape()		Optional	Deletes a virtual tape from a Virtual Library System.
MoveMedia()		Optional	Moves a virtual tape from one location to another.

11.8.57 SNIA_VirtualTapeServiceCapabilities

Capabilities associated with VirtualTapeService.

Requirement: Mandatory

Table 128 describes class SNIA_VirtualTapeServiceCapabilities.

Table 128 - SMI Referenced Properties/Methods for SNIA_VirtualTapeServiceCapabilities

Properties	Flags	Requirement	Description & Notes
InstanceID		Mandatory	Key.
SupportedTypes		Mandatory	
SupportedMethods	N	Mandatory	Shall be a combination of 'CreateTapeFromPool', 'DeleteTape', or 'MoveMedia'.

11.8.58 SNIA_VirtualTapeSetting

Settings for virtual tapes.

Created By: Static
Modified By: Static
Deleted By: Static
Requirement: Optional

Table 129 describes class SNIA_VirtualTapeSetting.

Table 129 - SMI Referenced Properties/Methods for SNIA_VirtualTapeSetting

Properties	Flags	Requirement	Description & Notes
InstanceID		Mandatory	Key.
Type		Mandatory	
ElementName		Optional	User friendly name.

EXPERIMENTAL

EXPERIMENTAL

Clause 12: Virtual Tape Library Copy Profile

12.1 Description

12.1.1 Overview

This profile describes a backup application independent way of triggering tape copies in SMI-S.

Research shows that backup applications need to add metadata to tape when triggering tape copies. Our goal is to propose an interface that will allow backup applications to write and read this metadata to and from tape at any time during the tape copy process.

12.2 Tape Copy Services

12.2.1 Summary

Figure 27 summarizes available tape copy services.

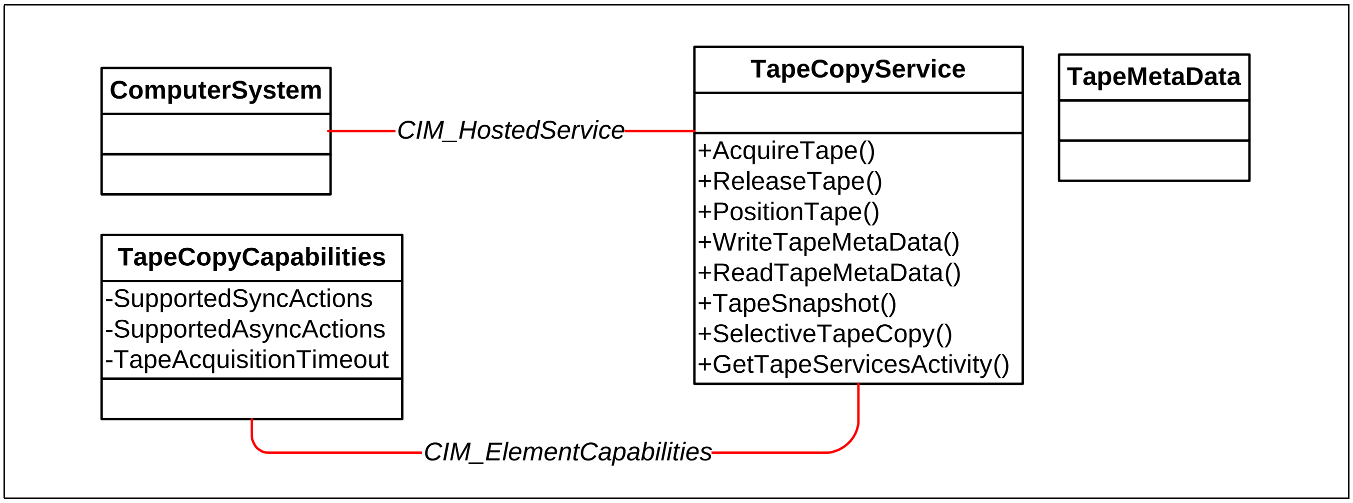


Figure 27 - Tape Copy Services Class Diagram

Tape Copy Services introduces the concept of acquiring and releasing tapes. Any sequence of position, read or write operations described further shall be performed within these two calls. These specifications do not specify provider behavior in the event of concurrent access from multiple clients (See 12.2.2.3 "Concurrency Considerations"). However, AcquireTape shall return an error if the tape is currently acquired. The activity status of a given tape should be obtained via a call to GetTapeServicesActivity() in order to check whether it is safe to call AcquireTape. Note that two calls to AcquireTape will be necessary for tape copy operations (e.g., TapeSnapshot or SelectiveTapeCopy…): one for the source tape and another one for the destination tape.

12.2.2 Definitions

12.2.2.1 TapeMetaData Class

The TapeMetaData class shown in Figure 28 is used to represent metadata and is composed of:

- "An array of strings encoded in hex binary, using the Octetstring qualifier

- "An integer value as to the number of file marks to write before writing metadata.
- "An integer value as to the number of file marks to write after writing metadata.
- "An array of integers or "bit codes" that indicate the block size of each data string.

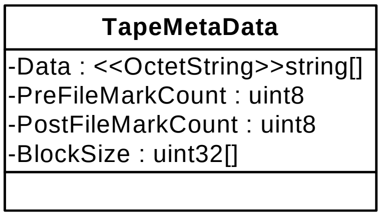

TapeMetaData
-Data : <<OctetString>>string[]
-PreFileMarkCount : uint8
-PostFileMarkCount : uint8
-BlockSize : uint32[]

Figure 28 - TapeMetaData Class Definition

Data is defined as a string array and qualified by Octetstring. This means that every string will be encoded using the cim:cimHexBinary type. In this encoding scheme, every byte of data is encoded in 4 hexadecimal characters leading to a 4:1 negative compression ratio, accounting for the fact that CIM uses the UCS2 character set. This format allows the TapeMetaData object to be packaged as an embedded object (See definition of WriteTapeMetaData in 12.2.2.7 "WriteTapeMetaData".)

12.2.2.2 Considerations on Load/Unload

This interface purposely does not define load and unload calls in order to allow for tape copy logic to reside in the device itself. For instance, the device has unique knowledge of what drives should be used to trigger a copy.

However, upon processing an AcquireTape call, the provider shall ensure that a tape is loaded and ready for data access, at least until ReleaseTape is called. AcquireTape may or may not result in a tape being loaded depending on whether the tape was already accessible.

The unload behavior is undefined. These specifications make no recommendation as to whether a tape should be unloaded after a ReleaseTape call is processed. Some implementations may decide to unload the tape immediately, after an arbitrarily defineTapeMeta DataTimeout attribute has expired, or simply when the drive is required for another task.

12.2.2.3 Concurrency Considerations

It is beyond the scope of this profile to specify concurrency behavior and/or to define a locking mechanism associated to the action of acquiring a tape. However, AcquireTape will fail if called twice (from any client) without an intermediate ReleaseTape. Likewise, ReleaseTape will return an error if called on a tape that was not previously acquired. The activity/status of a given tape can be obtained via the GetTapeServicesActivity operation.

12.2.2.4 Acquire Tape

AcquireTape initializes a 'transaction' for a given tape and informs the provider that a sequence of actions will be performed on that tape. A call to AcquireTape is required before any sequence of actions can be performed, actions that will result in an error if the tape activity isn't "Acquired". See GetTapeCopyServicesActivity (12.2.2.11) for defined activity values. These actions are:

"PositionTape

"WriteMetaData

"ReadMetaData

"TapeSnapshot

"SelectiveTapeCopy

GetTapeServicesActivity is the only call pertaining to a tape that doesn't require that tape to be "Acquired"

For any given tape, additional calls to AcquireTape shall return an error unless ReleaseTape was called or if the AcquireTape timeout has expired.

AcquireTape may or may not load a tape (See 12.2.2.2 "Considerations on Load/Unload")

AcquireTape (dest,timeout, Job)

"dest [IN:CIM_PhysicalTape REF]: destination tape

"timeout[IN:datetime]: Timeout after which an inactive 'transaction' will be released automatically by the provider. This argument is optional: if not specified, the default timeout specified by TapeCopyCapabilities.TapeAcquisitionTimeout will be used by the provider

"Job [OUT: CIM_ConcreteJob REF]: Job identifier

"Return Value[uint16]: Success / Failure

12.2.2.5 Release Tape

ReleaseTape marks the end of a 'transaction'. At this point, the tape becomes available for use by other clients or for a new 'transaction'. ReleaseTape shall fail if called on a tape whose activity is not "Acquired"

ReleaseTape (dest, Job)

"dest [IN:CIM_PhysicalTape REF]: Tape to release

"job [OUT: CIM_ConcreteJob REF]: Job identifier

"Return Value[uint16]: Success / Failure

12.2.2.6 PositionTape

PositionTape is used to position a tape before data gets read or written by one of the following calls: ReadTapeMetaData, WriteTapeMetaData, TapeSnapshot, and SelectiveTapeCopy. PositionTape uses relative positioning and can also be used to rewind the tape by passing 0 as a start position. PositionTape shall fail if called on a tape whose activity is not "Acquired"

PositionTape (dest, startType, start, job)

"dest [IN:CIM_PhysicalTape REF]: Destination tape

"startType [IN:uint16(enumeration)]: start position type ("filemark" or "block")

"start [IN:sint64]: start position for reading. Relative positioning implies that negative values are acceptable. A zero value has a special meaning and will trigger a full rewind of the tape.

"job [OUT: CIM_ConcreteJob REF]: Job identifier

"Return Value[uint16]: Success / Failure.

12.2.2.7 WriteTapeMetaData

WriteTapeMetaData is used to write metadata to tape at a position previously specified by a call to PositionTape. The size of the tape metadata size is limited to 1MB beyond which an error will be returned. WriteTapeMetaData shall fail if called on a tape whose activity is not "Acquired"

WriteTapeMetaData (dest, data, job)

"dest [IN:CIM_PhysicalTape REF]: Destination tape

"data[IN:TapeMetaData]: a TapeMetaData object to be written to tape

"job [OUT: CIM_ConcreteJob REF]: Job identifier

"Return Value[uint16]: Success / Failure

12.2.2.8 ReadTapeMetaData

ReadTapeMetaData is used to read meta data from tape. The tape meta data size is limited to 1MB beyond which an error will be returned.

Note that the data parameter is a reference, which means that clients will have to retrieve the actual data from the provider through CIM access methods (i.e., GetInstance). The lifecycle of tape meta data on the provider is defined as follows:

"A list of TapeMetaData instances will be maintained for every acquired tape.

"The provider will delete a given TapeMetaData instance upon receiving a GetInstance call.

"Upon receiving a ReleaseTape call, the provider will clear its list of TapeMetaData instances, thus ensuring proper memory management.

ReadTapeMetaData shall fail if called on a tape whose activity is not "Acquired"

ReadTapeMetaData (dest, sizeType, size, data, job)

"dest [IN:CIM_PhysicalTape REF]: Destination tape

"sizeType [IN:uint16(enumeration)]: type of the elements to be copied ("filemark" or "block")

"size [IN:uint32]: number of elements of type "sizeType" to be copied

"data [OUT:TapeMetaData REF]: meta data to be read.

"job [OUT: CIM_ConcreteJob REF]: Job identifier

"Return Value[uint16]: Success / Failure

12.2.2.9 TapeSnapshot

TapeSnapshot triggers a snapshot copy. It is used for simple snapshot. TapeSnapshot shall fail if called on a tape whose activity is not "Acquired"

TapeSnapshot (src, dest, copyType, job)

"src [IN:CIM_PhysicalTape REF]: Source tape

"dest [IN:CIM_PhysicalTape REF]: Destination tape

"copyType [IN:uint16(enumeration)]: Type of copy ("SimpleSnapshot")

"job [OUT: CIM_ConcreteJob REF]: Job identifier

"Return Value[uint16]: Success / Failure

12.2.2.10 SelectiveTapeCopy

SelectiveTapeCopy is used for partially copying tape data. It allows a host system to copy all or part of a tape to another tape. SelectiveTapeCopy is to be used in conjunction with WriteTapeMetaData to add metadata to tape

(See 12.3.2 "Selective Tape Copy recipe" for an action sequence example). SelectiveTapeCopy shall fail if called on a tape whose activity is not "Acquired"

SelectiveTapeCopy (src, dest, copyType, src, dest, sizeType, size, job)

"src [IN:CIM_PhysicalTape REF]: Source tape

"dest [IN:CIM_PhysicalTape REF]: Destination tape

"copyType=SelectiveCopy [IN:uint16(enumeration)]: type of copy

"sizeType [IN:uint16(enumeration)]: type of the elements to be copied ("filemark" or "block")

"size [IN:uint32]: number of elements of type "sizeType" to be copied

"job [OUT: CIM_ConcreteJob REF]: Job identifier

"Return Value[uint16]: Success / Failure

12.2.2.11 GetTapeCopyServicesActivity

GetTapeCopyServices indicates what copy-related actions are currently performed on a given tape. In a non-locking scenario, concurrent clients can use this call to check whether copy operations are in progress.

GetTapeCopyServicesActivity (dest, activity, job)

"dest [IN:CIM_PhysicalTape REF]: Destination tape

"activity [OUT:uint16(enumeration)]: type of copy

Activity is an enumeration type defined as follows:

"Idle: The target tape is not currently "acquired"

"Acquired: The target tape is "acquired" and no operation is currently being performed

"Writing: The target tape is "acquired" and tape meta data is being written

"Reading: The target tape is a "acquired" and meta data is being read

"Positioning: The target tape is a "acquired" and being positioned

"Copying_snapshot: The target tape is the source or destination of a snapshot copy

"Copying_selective: The target tape is the source or destination of a selective copy

"Return Value[uint16]: Success / Failure

12.2.2.12 Job Termination

These specifications do not specify means to terminate a running job. This is left up to the Job Control Profile. This has two consequences:

- "Synchronous implementations of the VTL Profile methods cannot be explicitly aborted.
- "Support for the Job Control Profile is conditional: if the provider features one or more asynchronous implementations of the VTL Profile methods, then it shall support the Job Control Profile.

Upon receiving a termination request for a given job, the SMI-S provider shall interrupt the specified job. These specifications do not make any recommendations as whether corrective actions should be taken It makes sense however to let the client application handle the failure, reposition the tape etc. Job termination impacts a job, not a 'transaction'. This means that another job can be started without having to reacquire the tape. As a corollary, this also means that terminating a job doesn't preclude the client application to release the tape to mark the end the 'transaction'.

12.3 Recipes

12.3.1 Simple Snapshot recipe

The simple snapshot feature copies one piece of media to another. The source and destination may be either physical media or virtual media in a single virtual tape library system.

Using an arbitrary tape format, here's a sequence of action that would be used to perform a simple snapshot from Tape1 to Tape2:

```
“Obtain a lock on the destination tape: AcquireTape (Tape2, timeout, &job)
“Optionally, read meta data at the beginning of Tape2 and make sure this is the "right" tape:
    Rewind: PositionTape (Tape2, "filemark", 0)
    ReadTapeMetaData ("filemark", 1, &data, &job)
    Backup app internal validation
"Write meta data at the beginning of tape:
    Construct TapeMetaData object (data)
    Rewind: PositionTape (Tape2, "filemark", 0)
    WriteTapeMetaData(Tape2, data, &job).
“Acquire source tape: AcquireTape (Tape1, timeout, &job)
“Position tape after the first meta data section:
    Rewind: PositionTape (Tape1, "filemark", 0)
    Skip meta data: PositionTape(Tape1, "filemark", 1)
“Perform snapshot: TapeSnapshot(Tape1, Tape2, "SimpleSnapshot", &job)
“Release source and destination tapes
    ReleaseTape (Tape1, &job)
    ReleaseTape (Tape2, &job)
```

Note that the same result could be achieved by using the selective tape copy service passing 0 for the source start position and size arguments. To allow for vendors who do not want to support selective tape copy, I believe the simple snapshot case should remain in the specifications.

12.3.2 Selective Tape Copy recipe

Using an arbitrary tape format, here's a typical sequence of actions that could be performed to do a selective tape copy from Tape1 to Tape2 for n elements of type sizeType:

```
"Obtain a lock on the destination tape: AcquireTape (Tape2, timeout, &job)
"Optionally, read meta data at the beginning of Tape2 and make sure this is the "right" tape:
    Rewind: PositionTape (Tape2, "filemark", 0)
    ReadTapeMetaData (Tape2, "filemark", 1, &data, &job)1
    Backup app internal validation
"Optionally, write meta data at the beginning of tape:
    Construct TapeMetaData object (data)
    Rewind: PositionTape (Tape2, "filemark", 0)
    WriteTapeMetaData(Tape2, data, &job).
"Write meta data for this copy:
    Construct TapeMetaData object (data)
    PositionTape (Tape2, startType, destStartPosition, &job)
    WriteTapeMetaData(Tape2, data, &job)
"Acquire source tape: AcquireTape (Tape1, timeout, &job)
"Position source tape at appropriate location:
    Rewind (only if necessary - this depends on the client application): PositionTape (Tape1, "filemark", 0)
    PositionTape(Tape1, startType, srcStartPosition, &job)
"Do the copy: SelectiveTapeCopy(Tape1, handle2, "SelectiveCopy", sizeType, size, &job)
"Write some more meta data:
    Construct TapeMetaData object (data)
    WriteTapeMetaData(Tape2, data, &job).
"Possibly run other jobs…
"Release source and destination tapes
    ReleaseTape (Tape1, &job)
    ReleaseTape (Tape2, &job)
```

12.4 Health and Fault Management Consideration

Not supported in this version of the standard.

12.5 Cascading Considerations

None

12.6 Registered Name and Version

Tape Copy Service version 1.3.0 (Autonomous Profile)

CIM Schema Version: 2.15.0

12.7 CIM Elements

Table 130 describes the CIM elements for Tape Copy Service.

Table 130 - CIM Elements for Tape Copy Service

Element Name	Requirement	Description
12.7.1 CIM_ElementCapabilities	Mandatory	Association linking the SNIA_TapeCopyService object to the SNIA_TapeCopyCapabilities object.
12.7.2 CIM_HostedService	Mandatory	Association linking a VLSSystem CIM_ComputerSystem object to the SNIA_TapeCopyService object.
12.7.3 SNIA_TapeCopyCapabilities	Mandatory	TapeCopyCapabilities describes functionality supported by TapeCopyService.
12.7.4 SNIA_TapeCopyService	Mandatory	Provides functions needed for 2 types of copy: snapshot copy and selective copy.
12.7.5 SNIA_TapeMetaData	Mandatory	SNIA_TapeMetaData represents backup-application-proprietary meta data that needs to be written or read to/from tape.

12.7.1 CIM_ElementCapabilities

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

Table 131 describes class CIM_ElementCapabilities.

Table 131 - SMI Referenced Properties/Methods for CIM_ElementCapabilities

Properties	Flags	Requirement	Description & Notes
ManagedElement		Mandatory	
Capabilities		Mandatory	

12.7.2 CIM_HostedService

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

Table 132 describes class CIM_HostedService.

Table 132 - SMI Referenced Properties/Methods for CIM_HostedService

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	
Dependent		Mandatory	

12.7.3 SNIA_TapeCopyCapabilities

Created By: Static
Modified By: Static
Requirement: Mandatory

Table 133 describes class SNIA_TapeCopyCapabilities.

Table 133 - SMI Referenced Properties/Methods for SNIA_TapeCopyCapabilities

Properties	Flags	Requirement	Description & Notes
SupportedSyncActions		Mandatory	SupportedSyncActions lists the tape copy services implemented synchronously.
SupportedAsyncActions		Mandatory	SupportedAsyncActions lists the tape copy services implemented asynchronously.
DefaultTimeout		Mandatory	DefaultTimeout is the time after which a transaction initiated by a call to TapeCopyService.AquireTape() will be released automatically by the provider if TapeCopyService.ReleaseTape() wasn't called explicitly.

12.7.4 SNIA_TapeCopyService

Created By: Static
Modified By: Static
Requirement: Mandatory

Table 134 describes class SNIA_TapeCopyService.

Table 134 - SMI Referenced Properties/Methods for SNIA_TapeCopyService

Properties	Flags	Requirement	Description & Notes
AcquireTape()		Optional	AcquireTape initializes a transaction for a given tape and informs the provider that a sequence of actions will be performed on that tape. A transaction is defined as a sequence of actions on a tape, starting with AcquireTape and ending with ReleaseTape. A call to AcquireTape is required before any sequence of actions can be performed, actions that will result in an error if the tape activity is not 'Acquired'. These actions are: PositionTape, WriteMetaData, ReadMetaData, TapeSnapshot, SelectiveTapeCopy. GetTapeServicesActivity is the only call pertaining to a tape that doesn't require that tape to be 'Acquired'. For any given tape, additional calls to AcquireTape shall return an error unless ReleaseTape was called or if the AcquireTape timeout has expired.
ReleaseTape()		Optional	ReleaseTape marks the end of a transaction. At this point, the tape becomes available for use by other clients or for a new transaction. ReleaseTape shall fail if called on a tape whose activity is not 'Acquired'.
PositionTape()		Optional	PositionTape positions a tape before data gets read or written by one of the following calls: ReadTapeMetaData, WriteTapeMetaData, TapeSnapshot, SelectiveTapeCopy. PositionTape uses relative positioning and can also be used to rewind the tape by passing 0 as a start position. PositionTape shall fail if called on a tape whose activity is not 'Acquired'.
WriteTapeMetaData()		Optional	WriteTapeMetaData writes meta data to tape at a the current tape position (specified by PositionTape). The size of the tape meta data size is limited to 1MB beyond which an error will be returned. WriteTapeMetaData shall fail if called on a tape whose activity is not 'Acquired'.
ReadTapeMetaData()		Optional	ReadTapeMetaData reads meta data from tape. The tape meta data size is limited to 1MB beyond which an error will be returned. The metaData parameter is a reference, which means that clients will have to retrieve the actual data from the provider through CIM access methods (i.e. GetInstance). The lifecycle of tape meta data on the provider is defined as follows: (1) A list of TapeMetaData instances will be maintained for every acquired tape. (2) The provider will delete a given TapeMetaData instance upon receiving a GetInstance call. (3) Upon receiving a ReleaseTape call, the provider will clear its list of TapeMetaData instances, thus ensuring proper memory management.
TapeSnapshot()		Optional	TapeSnapshot triggers a snapshot copy. TapeSnapshot shall fail if called on tapes whose activity is not 'Acquired'.

Table 134 - SMI Referenced Properties/Methods for SNIA_TapeCopyService

Properties	Flags	Requirement	Description & Notes
SelectiveTapeCopy()		Optional	SelectiveTapeCopy partially copies data. It allows a host system to copy all or part of a tape to another tape. SelectiveTapeCopy is to be used in conjunction with WriteTapeMetaData to add meta data to tape. SelectiveTapeCopy shall fail if called on a tape whose activity is not 'Acquired'.
GetTapeServicesActivity()		Optional	GetTapeCopyServices indicates what copy-related actions is currently performed on a given tape. GetTapeCopyServices shall be implemented synchronously. Concurrent clients can use this call to check whether copy operations are in progress. Tape activity is returned by way of an OUT parameter, defined as an enumeration of the following values: (1) Idle: the target is not currently acquired. (2) Acquired: the target tape is currently acquired and no operation is currently being performed. (3) Writing: the target tape is acquired and tape meta data is being written. (4) Reading: the target tape is acquired and tape meta data is being read. (5) Positioning: the target tape is acquired and being positioned. (6) SnapshotCopy: the target tape is acquired and is the source or target tape of a snapshot copy. (7) SelectiveCopy: the target tape is acquired and is the source or target tape of a selective copy.

12.7.5 SNIA_TapeMetaData

Created By: Static
Modified By: Static
Requirement: Mandatory

Table 135 describes class SNIA_TapeMetaData.

Table 135 - SMI Referenced Properties/Methods for SNIA_TapeMetaData

Properties	Flags	Requirement	Description & Notes
MetaData		Mandatory	Array of binary blocks.
PreFileMarkCount		Mandatory	Number of filemarks to be written before the meta data blocks.
PostFileMarkCount		Mandatory	Number of filemarks to be written after the meta data blocks.
BlockSizes		Mandatory	Sizes of individual binary blocks.

EXPERIMENTAL

EXPERIMENTAL

Clause 13: Library Views Profile

13.1 Synopsis

Profile Name: Library Views (Component Profile)

Version: 1.6.0

Organization: SNIA

CIM Schema Version: 2.28

Table 136 describes the related profiles for Library Views.

Table 136 - Related Profiles for Library Views

Profile Name	Organization	Version	Requirement	Description
Masking and Mapping	SNIA	1.6.0	Conditional	Conditional requirement: Required if the array property CIM_ViewCapabilities.SupportedViews contains the string 'SNIA:MaskingMappingExposedDeviceView' (and the Masking and Mapping Profile is implemented).

Scoping Class: CIM_ComputerSystem

Central Class: CIM_ViewCapabilities

13.2 Description

13.2.1 Overview

This Profile specifies View Classes for Storage Media Library Profile, Virtual Tape Library Profile and Partitioned Tape Library Profile. (See Clause 4: Storage Library Profile, Clause 10: Partitioned Tape Library Profile, and Clause 11: Virtual Tape Library Profile.)

View classes provide an optimization of retrieval of information provided by multiple (associated) instances in a Profile. There is no support for update of view classes instances. Update of a view class instance can only be accomplished by updating the base class instances from which the view is derived.

13.2.1.1 Goals of View Classes

13.2.1.1.1 Intended Goals

Goals that View Classes are intended to address are:

- Get more data in one call to CIM Server.

The CIM model for arrays and Storage Virtualizers involve a lot of classes and associations. The objective is to allow discovery of the array model using View Classes with a reduction in the number of association traversals required.

- Allow providers to optimize the Request.

In many cases, the data represented by a View Class is actually kept (and returned) by a device as one entity. When the "normalized" CIM model is traversed many calls are made to retrieve that one entity. The provider takes the data from the one entity and carves it up for each CIM request. In many cases this involves retrieving the same entity multiple times. The objective is to allow a Provider to return the single entity in one SMI-S request (for data that is typically kept together by the device).

13.2.1.1.2 Additional Goals

- Do more things in one call to CIM Server.

An example would be retrieval or discovery of model information with fewer calls. However, this goal also extends to updating the CIM model (e.g., configuration actions). The View Classes are NOT intended to help in the latter case. However, View Classes should facilitate access to underlying classes in support of configuration operations.

It is important to note that the View Classes proposal was based directly on experiences relating to the scalability and performance of SMI-S real-world implementations. The focus is on improving performance in large configurations (e.g., thousands of tape cartridges and thousands of tape drives).

13.2.2 Class Diagram for Library Views View Classes

Figure 29 illustrates the class diagram for the Library view classes.

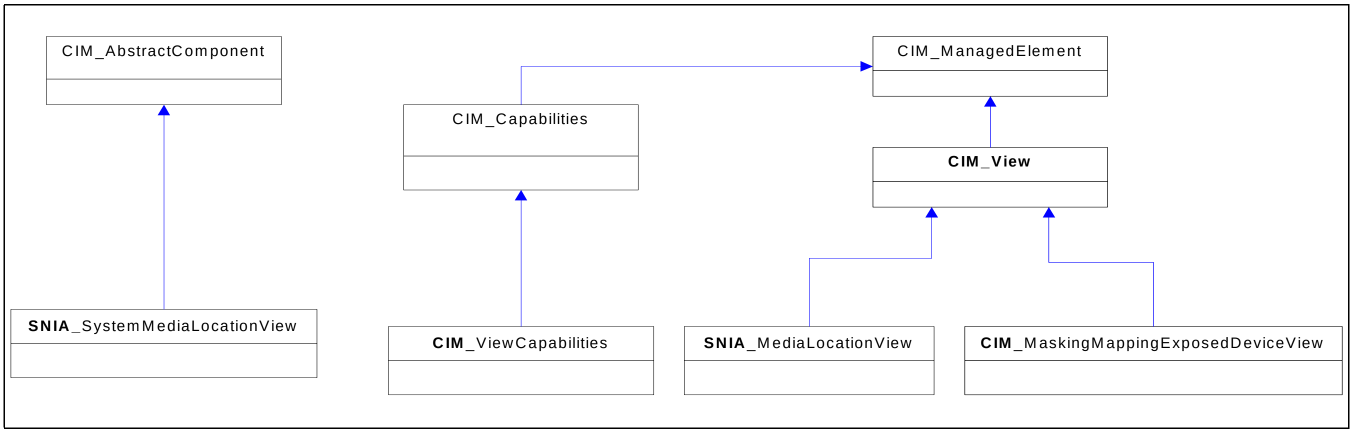


Figure 29 - Class Diagram for Library View Classes

The ViewCapabilities inherits from CIM_Capabilities. The SNIA_MediaLocationView and CIM_MaskingMappingExposedDeviceView classes inherit from CIM_View. The Library Views profile has one subclass of CIM_AbstractComponent: SNIA_SystemMediaLocationView. This associate the view to its scoping system.

13.3 Implementation

13.3.1 View Class Capabilities

The implementation shall identify which view classes are implemented using a set of conditions.

First a client may determine whether or not a profile implementation has implemented any view classes by looking for a RegisteredSubprofile with a RegisteredName of “Library Views”. If this RegisteredSubprofile exists then the profile supports some number of view classes.

Next a client would be able to determine which view classes are supported by an implementation by following the ElementConformsToProfile to the top level system and then following the ElementCapabilities from that system to the CIM_ViewCapabilities instance. There shall be one instance of the CIM_ViewCapabilities class if the profile supports the Library Views Subprofile. The CIM_ViewCapabilities instance shall have an array of strings (SupportedViews) that identify the view classes that are supported. For example, if the SupportedViews array includes the “SNIA:MediaLocationView” string, then the MediaLocationView class shall be supported.

13.3.2 Media Location Views

13.3.2.1 SNIA_MediaLocationView and related associations

The SNIA_MediaLocationView is composed of information drawn from the following base classes:

- CIM_PhysicalMedia
- CIM_StorageMediaLocation

The keys for the SNIA_MediaLocationView are the Tag keys from the base CIM_PhysicalMedia and CIM_StoragemediaLocation classes. There will be one instance of SNIA_MediaLocationView for each instance of CIM_PhysicalMedia.

The following properties describe the SNIA_MediaLocationView class.

- SML.Tag
- SML.LocationType
- SML.LocationCoordinates
- SML.MediaCapacity
- PM.Tag
- PM.Capacity
- PM.MediaType
- PM.MediaDescription
- PM.CleanerMedia
- PM.DualSided
- PM.PhysicalLabels

13.3.2.2 Mandatory, Conditional and Optional Properties of SNIA_MediaLocationView

Properties that are mandatory in the mandatory base classes are mandatory in the SNIA_MediaLocationView class. Properties that are Conditional in the base classes are conditional in the SNIA_MediaLocationView class.

Properties in the base classes that are optional in the base class are optional in the SNIA_MediaLocationView.

13.3.2.3 Associations on SNIA_MediaLocationView

The SNIA_MediaLocationView is "read only." Access to CIM class instances on which the view is based on can be accessed via the CIM_ElementView association.

13.3.3 Masking and Mapping Views

13.3.3.1 The CIM_MaskingMappingExposedDeviceView Association

The CIM_MaskingMappingExposedDeviceView association is composed of information drawn from the following base classes:

- SCSIProtocolController
- SAPAvailableForElement
- ProtocolControllerForUnit

The keys for the CIM_MaskingMappingExposedDeviceView are the references to the LogicalDevice (a MediaAccessDevice) and the reference to the SCSIProtocolEndpoint. There will be one instance of CIM_MaskingMappingExposedDeviceView for each unique combination of MediaAccessDevice and SCSIProtocolEndpoint.

13.3.3.1.1 Mandatory, Conditional and Optional Properties of CIM_MaskingMappingExposedDeviceView Association

In addition to the references to MediaAccessDevice and the SCSIProtocolEndpoint the CIM_MaskingMappingExposedDeviceView association also carries the DeviceID of the SCSIProtocolController and the DeviceNumber and DeviceAccess properties from the ProtocolControllerForUnit association.

In this release of SMI-S, the CIM_MaskingMappingExposedDeviceView is "read only." It would be used to do association traversal from MediaAccessDevice to SCSIProtocolEndpoints that expose the tape drive.

13.3.4 Health and Fault Management Consideration

Not defined in this standard.

13.3.5 Cascading Considerations

Not defined in this standard.

13.4 Methods of the Profile

13.4.1 Extrinsic Methods of the Profile

Not defined in this standard.

13.4.2 Intrinsic Methods of the Profile

The profile supports read methods and association traversal. Specifically, the list of intrinsic operations supported are as follows:

- GetInstance
- Associators
- AssociatorNames
- References
- ReferenceNames
- EnumerateInstances
- EnumerateInstanceNames

View classes are modified by creating, deleting and modifying the base classes from which they are derived. The property values of View classes are derived from the property values of associated classes. This profile does not specify the means to modify, create, or delete those classes. The base class instances may be accessed from the view class instances via association traversal through the CIM_ElementView association

13.5 Use Cases

Not supported in this version of the standard.

13.6 CIM Elements

Table 137 describes the CIM elements for Library Views.

Table 137 - CIM Elements for Library Views

Element Name	Requirement	Description
13.6.1 CIM_ElementCapabilities (View Capabilities)	Mandatory	Associates the top level ComputerSystem to the CIM_ViewCapabilities supported by the implementation.
13.6.2 CIM_MaskingMappingExposedDeviceView	Conditional	Conditional requirement: Required if the array property CIM_ViewCapabilities.SupportedViews contains the string 'SNIA:MaskingMappingExposedDeviceView' (and the Masking and Mapping Profile is implemented). This view associates a Target SCSIProtocolEndpoint and a LogicalDevice (e.g., MediaAccessDevice). This is required if the CIM_ViewCapabilities.SupportedViews includes "SNIA:MaskingMappingExposedDeviceView".
13.6.3 CIM_ViewCapabilities	Mandatory	The CIM_ViewCapabilities identifies the capabilities of the implementation of view classes.

Table 137 - CIM Elements for Library Views

Element Name	Requirement	Description
13.6.4 SNIA_MediaLocationView	Conditional	Conditional requirement: Required if the array property CIM_ViewCapabilities.SupportedViews contains the string 'SNIA:MediaLocationView'. This view combines the information from a PhysicalMedia instance with its StorageMediaLocation instance. This is required if the CIM_ViewCapabilities.SupportedViews includes "SNIA:MediaLocationView".
13.6.5 SNIA_SystemMediaLocationView (MediaLocationViews)	Conditional	Conditional requirement: Required if the array property CIM_ViewCapabilities.SupportedViews contains the string 'SNIA:MediaLocationView'. This association links SNIA_MediaLocationView instances to the scoping system. This is required if the SNIA_MediaLocationView is implemented.

13.6.1 CIM_ElementCapabilities (View Capabilities)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 138 describes class CIM_ElementCapabilities (View Capabilities).

Table 138 - SMI Referenced Properties/Methods for CIM_ElementCapabilities (View Capabilities)

Properties	Flags	Requirement	Description & Notes
Capabilities		Mandatory	The ViewCapabilities.
ManagedElement		Mandatory	The top level ComputerSystem that has the ViewCapabilities.

13.6.2 CIM_MaskingMappingExposedDeviceView

The CIM_MaskingMappingExposedDeviceView instance is a view that is derived from CIM_SAPAvailableForElement, CIM_SCSIProtocolController and CIM_ProtocolControllerForUnit. The CIM_MaskingMappingExposedDeviceView is not subclassed from anything.

Created By: External

Modified By: External

Deleted By: External

Requirement: Required if the array property CIM_ViewCapabilities.SupportedViews contains the string 'SNIA:MaskingMappingExposedDeviceView' (and the Masking and Mapping Profile is implemented).

Table 139 describes class CIM_MaskingMappingExposedDeviceView.

Table 139 - SMI Referenced Properties/Methods for CIM_MaskingMappingExposedDeviceView

Properties	Flags	Requirement	Description & Notes
SPCSystemCreationClassName		Mandatory	The SystemCreationClassName for the SCSIProtocolController used with the underlying SCSIProtocolController instance for the SCSIProtocolEndpoint and MediaAccessDevice, ChangerDevice, or LimitedAccessPort.
SPCSystemName		Mandatory	The SystemName for the SCSIProtocolController used with the underlying SCSIProtocolController instance for the SCSIProtocolEndpoint and MediaAccessDevice, ChangerDevice, or LimitedAccessPort.
SPCCreationClassName		Mandatory	The CreationClassName for the SCSIProtocolController used with the underlying SCSIProtocolController instance for the SCSIProtocolEndpoint and MediaAccessDevice, ChangerDevice, or LimitedAccessPort.
SPCDeviceID		Mandatory	The DeviceID for the SCSIProtocolController used with the underlying SCSIProtocolController instance for the SCSIProtocolEndpoint and MediaAccessDevice, ChangerDevice, or LimitedAccessPort.
PCFUDeviceNumber		Mandatory	The DeviceNumber (LUN) for the MediaAccessDevice, ChangerDevice, or LimitedAccessPort when accessed through the SCSIProtocolEndpoint as reported in the underlying ProtocolControllerForUnit instance for the MediaAccessDevice, ChangerDevice, or LimitedAccessPort.
PCFUDeviceAccess		Mandatory	The DeviceAccess value for the MediaAccessDevice, ChangerDevice, or LimitedAccessPort when accessed through the SCSIProtocolEndpoint as reported in the underlying ProtocolControllerForUnit instance for the MediaAccessDevice, ChangerDevice, or LimitedAccessPort.
ProtocolEndpoint		Mandatory	The Target ProtocolEndpoint through which the LogicalDevice is exposed.
LogicalDevice		Mandatory	The LogicalDevice (e.g., StorageVolume) that is exposed through the Target ProtocolEndpoint.

13.6.3 CIM_ViewCapabilities

The CIM_ViewCapabilities instance defines the capabilities of an implementation support for view classes. The CIM_ViewCapabilities is subclassed from CIM_Capabilities.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 140 describes class CIM_ViewCapabilities.

Table 140 - SMI Referenced Properties/Methods for CIM_ViewCapabilities

Properties	Flags	Requirement	Description & Notes
InstanceID		Mandatory	An opaque, unique id for the view class capability of an implementation.
ElementName		Optional	A provider supplied user-friendly Name for this CIM_ViewCapabilities element.
SupportedViews		Mandatory	This array of strings lists the view classes that are supported by the implementation. Valid string values are 'SNIA:MaskingMappingExposedDeviceView' and 'SNIA:MediaLocationView'.

13.6.4 SNIA_MediaLocationView

The SNIA_MediaLocationView instance is a view that is derived from CIM_PhysicalMedia and CIM_StorageMediaLocation. The SNIA_MediaLocationView is subclassed from CIM_View.

Created By: External

Modified By: External

Deleted By: External

Requirement: Required if the array property CIM_ViewCapabilities.SupportedViews contains the string 'SNIA:MediaLocationView'.

Table 141 describes class SNIA_MediaLocationView.

Table 141 - SMI Referenced Properties/Methods for SNIA_MediaLocationView

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	The SystemCreationClassName for the StorageExtent of the Disk Drive as reported in the underlying system instance.
SystemName		Mandatory	The SystemName for underlying system instance.
PMTag		Mandatory	An opaque identifier of the underlying PhysicalTape.
SMLTag		Mandatory	An opaque identifier of the underlying StorageMediaLocation.
SMLLocationType		Mandatory	LocationType property from StorageMediaLocation class.
SMLLocationCoordinates		Mandatory	LocationCoordinates property from StorageMediaLocation class.
SMLMediaCapacity		Mandatory	MediaCapacity property from StorageMediaLocation class.
PMCapacity		Mandatory	Capacity property from PhysicalMedia class.
PMMediaType		Mandatory	MediaType from PhysclaMedia class.
PMMediaDescription		Mandatory	MediaDescription property from PhysicalMedia class.

Table 141 - SMI Referenced Properties/Methods for SNIA_MediaLocationView

Properties	Flags	Requirement	Description & Notes
PMCleanerMedia		Mandatory	CleanerMedia property from PhysicalMedia class.
PMDualSided		Mandatory	DualSided property from PhysicalMedia class.
PMPhysicalLabels		Mandatory	PhysicalLabels property from PhysicalMedia class.

13.6.5 SNIA_SystemMediaLocationView (MediaLocationViews)

Created By: External

Modified By: Static

Deleted By: External

Requirement: Required if the array property CIM_ViewCapabilities.SupportedViews contains the string 'SNIA:MediaLocationView'.

Table 142 describes class SNIA_SystemMediaLocationView (MediaLocationViews).

Table 142 - SMI Referenced Properties/Methods for SNIA_SystemMediaLocationView (MediaLocationViews)

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	The Computer System that contains this MediaLocationView instance.
Dependent		Mandatory	The SNIA_MediaLocationView instance that is a device on the computer system.

EXPERIMENTAL

EXPERIMENTAL

Clause 14: VTL Capacity and Reporting Profile

14.1 Synopsis

Profile Name: VTL Capacity and Reporting (Component Profile)

Version: 1.6.0

Organization: SNIA

CIM Schema Version: 2.29.0

Table 143 describes the related profiles for VTL Capacity and Reporting.

Table 143 - Related Profiles for VTL Capacity and Reporting

Profile Name	Organization	Version	Requirement	Description
Indication	SNIA	1.5.0	Support for at least one is mandatory.	Deprecated. See the SNIA Indications Profile
Indications	SNIA	1.6.0		Experimental.
Indications	DMTF	1.2.0		Experimental. See DSP1054, version 1.2.0

Scoping class: CIM_ComputerSystem

Central Class: CIM_StatisticalDataService

The VTL Reporting and Capacity profile is a sub profile whose aim is to provide access point in time and historical resource usage information (physical/logical capacity, deduplication ratio etc.) for VTL devices. This sub profile is intended to be supported by virtual tape appliances

14.2 Description

There are two aspects to this sub profile:

Provide point in time resource usage data, including

1) LogicalCapacity

2) PhysicalCapacity

3) AlertThresholds

4) SystemRatio

5) DeduplicationRatio

6) Throughput

Provide access to statistical data through the StatisticalDataService class. This subprofile doesn't allow clients to mine historical device data because it would affect performance and require much work on the client side. Instead, extrinsic methods are provided to fetch graph data. These methods are:

1) GetPhysicalCapacityUsage

2) GetLogicalCapacityUsage

3) GetSystemRatio

4) GetDeduplicationRatio

5) GetReplicationThroughput

6) GetReplicationRatio

7) GetPerformanceStatisticsReadWriteIO

8) GetFCPortStatisticsSRR

9) GetFCPortStatisticsAborts

14.3 Implementation

14.3.1 OverView

Figure 30 shows an overview of the classes of the VTL Capacity and Reporting profile and how they relate to elements in the Virtual Tape Library Profile.

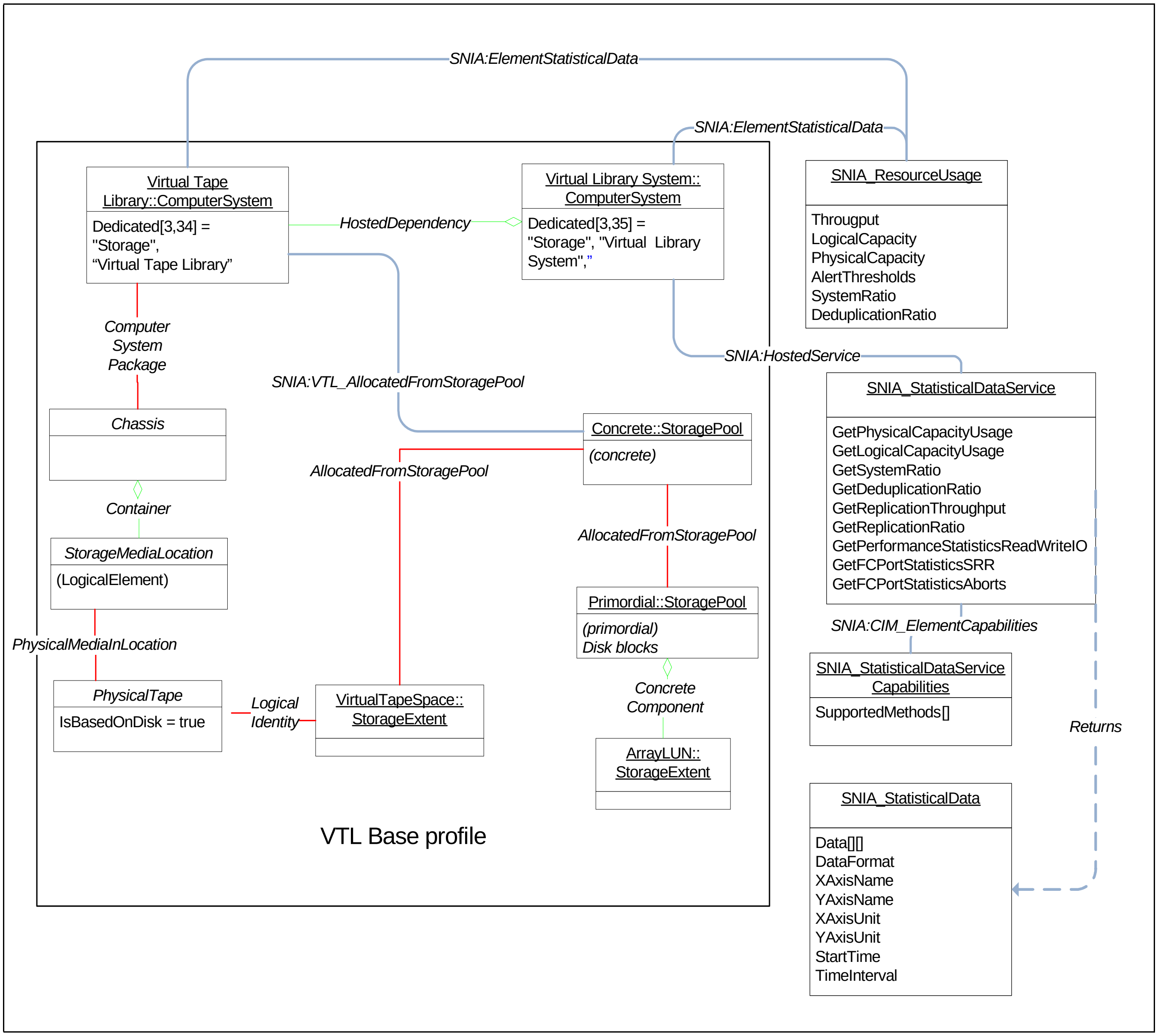


Figure 30 - Overview of the VTL Reporting and Capacity subprofile

14.3.2 Current capacity and performance for VLS & VTL Computer Systems

Figure 2 describes the model used to report point-in-time data. The SNIA_VLSResourceUsage class contains data representing used capacity, compression ratios and deduplication ratios. It also reports settable alert thresholds.

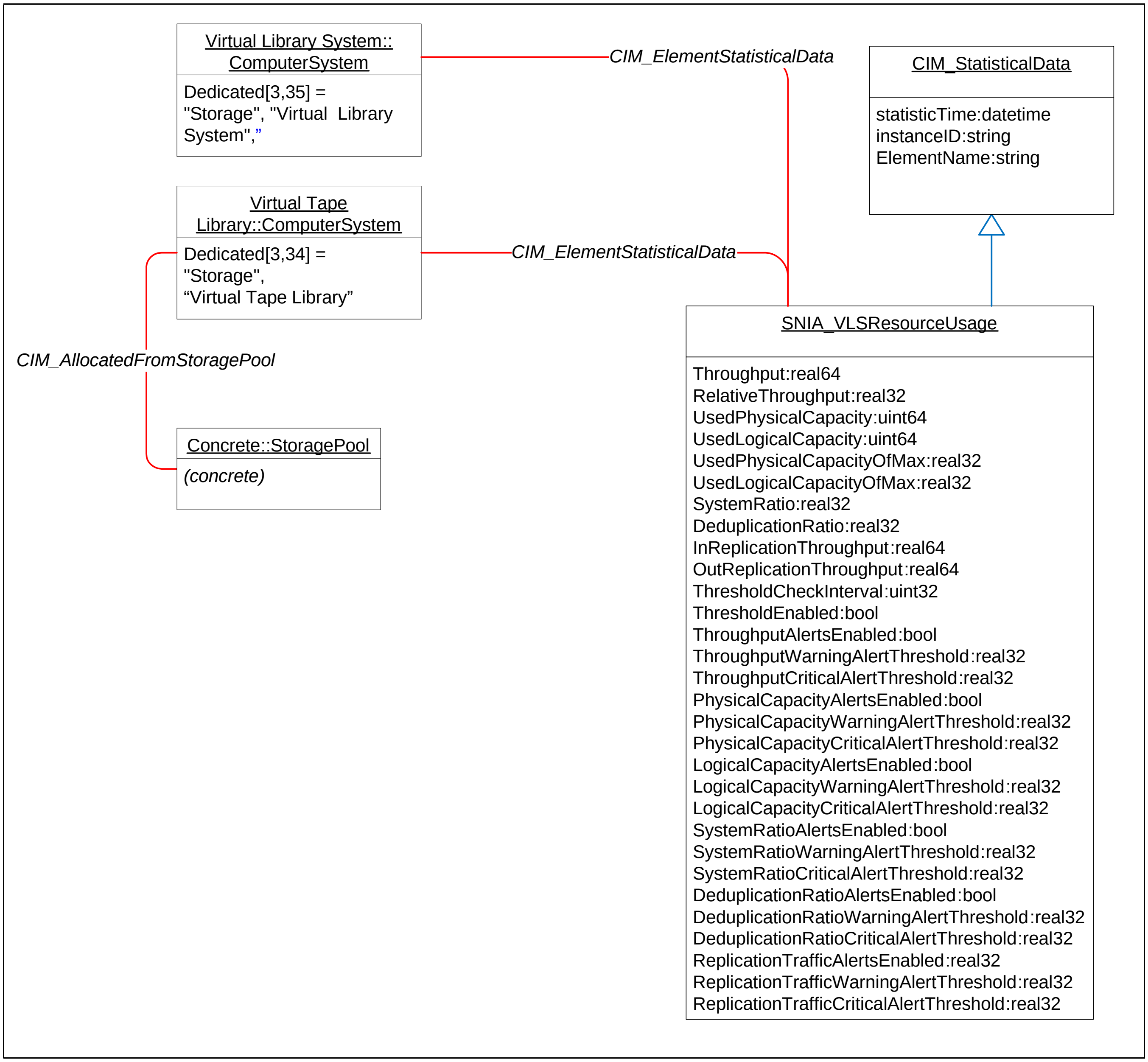

Virtual Library System::
ComputerSystem
Dedicated[3,35] =
"Storage", "Virtual Library
System","
CIM_ElementStatisticalData
CIM_StatisticalData
statisticTime:datetime
instanceID:string
ElementName:string
Virtual Tape
Library::ComputerSystem
Dedicated[3,34] =
"Storage",
“Virtual Tape Library”
CIM_ElementStatisticalData
CIM_AllocatedFromStoragePool
Concrete::StoragePool
(concrete)
SNIA_VLSResourceUsage
Throughput:real64
RelativeThroughput:real32
UsedPhysicalCapacity:uint64
UsedLogicalCapacity:uint64
UsedPhysicalCapacityOfMax:real32
UsedLogicalCapacityOfMax:real32
SystemRatio:real32
DeduplicationRatio:real32
InReplicationThroughput:real64
OutReplicationThroughput:real64
ThresholdCheckInterval:uint32
ThresholdEnabled:bool
ThroughputAlertsEnabled:bool
ThroughputWarningAlertThreshold:real32
ThroughputCriticalAlertThreshold:real32
PhysicalCapacityAlertsEnabled:bool
PhysicalCapacityWarningAlertThreshold:real32
PhysicalCapacityCriticalAlertThreshold:real32
LogicalCapacityAlertsEnabled:bool
LogicalCapacityWarningAlertThreshold:real32
LogicalCapacityCriticalAlertThreshold:real32
SystemRatioAlertsEnabled:bool
SystemRatioWarningAlertThreshold:real32
SystemRatioCriticalAlertThreshold:real32
DeduplicationRatioAlertsEnabled:bool
DeduplicationRatioWarningAlertThreshold:real32
DeduplicationRatioCriticalAlertThreshold:real32
ReplicationTrafficAlertsEnabled:real32
ReplicationTrafficWarningAlertThreshold:real32
ReplicationTrafficCriticalAlertThreshold:real32

Figure 31 - Resource Usage Centric View

14.3.3 Historical resource usage

Figure 32 illustrates the elements of the VTLStatisticalDataService and its relationship to the Virtual Library System.

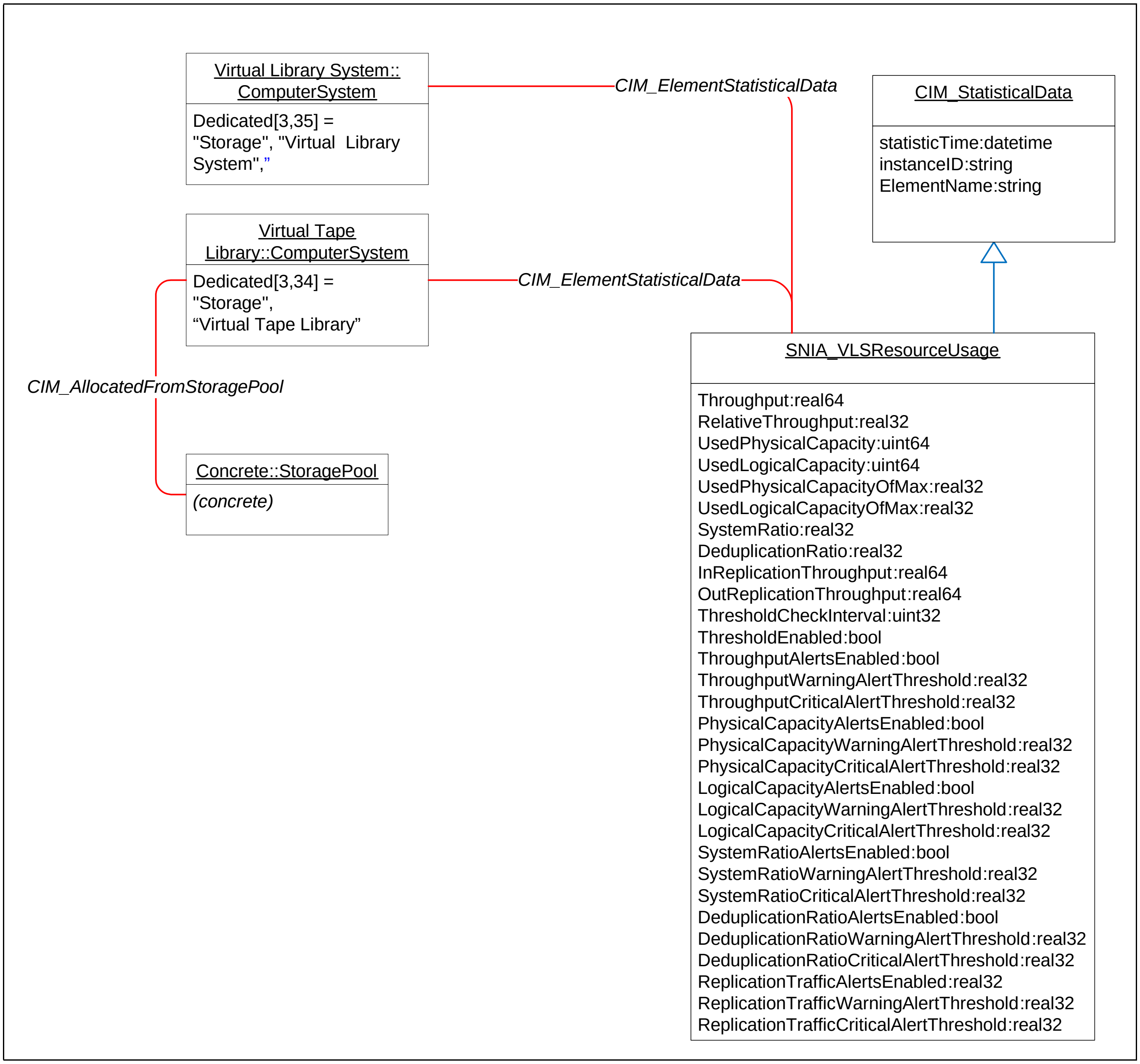

Virtual Library System::
ComputerSystem
Dedicated[3,35] =
"Storage", "Virtual Library
System","
CIM_ElementStatisticalData
CIM_StatisticalData
statisticTime:datetime
instanceID:string
ElementName:string
Virtual Tape
Library::ComputerSystem
Dedicated[3,34] =
"Storage",
"Virtual Tape Library"
CIM_ElementStatisticalData
SNIA_VLSResourceUsage
CIM_AllocatedFromStoragePool
Concrete::StoragePool
(concrete)
Throughput:real64
RelativeThroughput:real32
UsedPhysicalCapacity:uint64
UsedLogicalCapacity:uint64
UsedPhysicalCapacityOfMax:real32
UsedLogicalCapacityOfMax:real32
SystemRatio:real32
DeduplicationRatio:real32
InReplicationThroughput:real64
OutReplicationThroughput:real64
ThresholdCheckInterval:uint32
ThresholdEnabled:bool
ThroughputAlertsEnabled:bool
ThroughputWarningAlertThreshold:real32
ThroughputCriticalAlertThreshold:real32
PhysicalCapacityAlertsEnabled:bool
PhysicalCapacityWarningAlertThreshold:real32
PhysicalCapacityCriticalAlertThreshold:real32
LogicalCapacityAlertsEnabled:bool
LogicalCapacityWarningAlertThreshold:real32
LogicalCapacityCriticalAlertThreshold:real32
SystemRatioAlertsEnabled:bool
SystemRatioWarningAlertThreshold:real32
SystemRatioCriticalAlertThreshold:real32
DeduplicationRatioAlertsEnabled:bool
DeduplicationRatioWarningAlertThreshold:real32
DeduplicationRatioCriticalAlertThreshold:real32
ReplicationTrafficAlertsEnabled:real32
ReplicationTrafficWarningAlertThreshold:real32
ReplicationTrafficCriticalAlertThreshold:real32

Figure 32 - Statistical Data Service

14.3.4 Health and Fault Management Consideration

14.4 Methods

SNIAVLS_VTLStatisticalDataService

All methods in this section return an error code defined as follows: { "Job Completed with No Error", "Not Supported", "Unknown", "Timeout", "Failed", "Invalid Parameter", "Method Parameters Checked - Job Started", "Statistics Format Not Supported", "Method Reserved", "Vendor Specific"}

Graphs are returned in the 'Data' OUT parameter.

Every method call accepts Start time, End Time and Duration parameters that defines the range and granularity of the data that will be returned. This 'context' parameter is used by the provider as a hint, hence 'context' might be different after the method call than what was passed in by the client application. Rules are:

1) StartTime, EndTime, and Duration are used to specify a time window for the data to be collected. One, two or three of these parameters can be specified.

2) If all three are specified, Duration is ignored.

3) Invalid cases such as StartTime > EndTime, or times set in the future will result in the invoked method to return an error.

4) If two out of the three parameters are specified, the provider will calculate the third one accordingly.

5) If only StartTime is specified, EndTime will be set to the current time.

6) If only EndTime is specified, StartTime will be set to the earliest data available.

7) If only duration is specified, then EndTime will be set to the current time.

GetPhysicalCapacityUsage (CIM_StoragePool REF, CIM_ComputerSystem REF)

GetPhysicalCapacityUsage(CIM_StoragePool REF, CIM_ComputerSystem REF): Provides capacity on a per storage pool and virtual library basis. If the primordial pool is used, data is returned for all pools. If the VLS Computer System is used, capacity is returned for the entire appliance.

```
GetPhysicalCapacityUsage(

        [IN(false), OUT, Description("Reference to the job (null if synchronous call)"]

        CIM_ConcreteJob REF Job,

        [IN, Description ("Specifies the format of the 'Data' output parameter')

        ValueMap{"2"},

        Values("CSV")]

        Uint16 DataFormat,

        [IN, Description ("Reference to the storage pool, concrete or primordial")]

CIM_StoragePool REF Pool,

       [IN, Description (

       "Specifies the start time from when the data is required."

       "Both StartTime and Duration cannot be null at the same time.")]

       datetime StartTime,

       [IN, Description (

       "Specifies the duration in seconds to generate statistical data."

       "Both StartTime and Duration cannot be null at the same time.")]

       uint64 Duration,

       [IN, Description (
       "Specifies the end time up to which the statistical data will be generated."
       "Current time will be assumed whenever EndTime is specified null.")]
       datetime EndTime,
 [IN, Description ("Reference to a VTL or the VLS")]
CIM_ComputerSystem REF Library,
        [IN (false), OUT, Description("Graph Object")]
        SNIAVLS_VTLStatisticalData Data);
        "GetLogicalCapacityUsage(CIM_ComputerSystem REF)
```

GetLogicalCapacityUsage(CIM_ComputerSystem REF)

Provides logical capacity usage for the entire VLS appliance or a specific VTL depending on the ComputerSystem reference passed in.

```
GetLogicalCapacityUsage(
         [IN(false), OUT, Description("Reference to the job (null if synchronous call)"]
         CIM_ConcreteJob REF Job,
         [IN, Description ("Specifies the format of the 'Data' output parameter')
         ValueMap{"2"},
         Values("CSV")]
         Uint16 DataFormat,
         [IN, Description ("Reference to a VTL or the VLS")]
CIM_ComputerSystem REF Library,
        [IN, Description (
        "Specifies the start time from when the data is required."
        "Both StartTime and Duration cannot be null at the same time.")]
        datetime StartTime,
        [IN, Description (
        "Specifies the duration in seconds to generate statistical data."
        "Both StartTime and Duration cannot be null at the same time.")]
        uint64 Duration,
        [IN, Description (
        "Specifies the end time up to which the statistical data will be generated."
        "Current time will be assumed whenever EndTime is specified null.")]
        datetime EndTime,
         [IN (false), OUT, Description("Graph object")]
         SNIAVLS_VTLStatisticalData Data);
         "GetSystemRatio(CIM_ComputerSystem, REF)
```

GetSystemRatio(CIM_ComputerSystem REF)

Provides an historic of the compression ratio for the entire VLS appliance or a specific VTL depending on the ComputerSystem reference passed in

```
GetSystemRatio (
    [IN(false), OUT, Description("Reference to the job (null if synchronous call)"]
    CIM_ConcreteJob REF Job,
    [IN, Description ("Specifies the format of the 'Data' output parameter')
    ValueMap{"2"},
    Values("CSV")]
    Uint16 DataFormat,
    [IN, Description ("Reference to a VTL or the VLS")]
CIM_ComputerSystem REF Library
   [IN, Description (
   "Specifies the start time from when the data is required."
   "Both StartTime and Duration cannot be null at the same time.")]
   datetime StartTime,
   [IN, Description (
   "Specifies the duration in seconds to generate statistical data."
   "Both StartTime and Duration cannot be null at the same time.")]
   uint64 Duration,
   [IN, Description (
   "Specifies the end time up to which the statistical data will be generated."
   "Current time will be assumed whenever EndTime is specified null.")]
   datetime EndTime,
    [IN (false), OUT, Description("Graph object")]
    SNIAVLS_VTLStatisticalData Data);
    "GetDeduplicationRatio(CIM_ComputerSystem REF)
```

GetDeduplicationRatio(CIM_ComputerSystem REF)

Provides an historic of the deduplication ratio for the entire VLS appliance or a specific VTL depending on the ComputerSystem reference passed in

```
GetDeduplicationRatio (
        [IN(false), OUT, Description("Reference to the job (null if synchronous call)"]
        CIM_ConcreteJob REF Job,
        [IN, Description ("Specifies the format of the 'Data' output parameter')
        ValueMap{"2"},
        Values("CSV")]
        Uint16 DataFormat,
        [IN, Description ("Reference to a VTL or the VLS")]
CIM_ComputerSystem REF Library
       [IN, Description (
       "Specifies the start time from when the data is required."
       "Both StartTime and Duration cannot be null at the same time.")]
       datetime StartTime,
       [IN, Description (
       "Specifies the duration in seconds to generate statistical data."
       "Both StartTime and Duration cannot be null at the same time.")]
       uint64 Duration,
       [IN, Description (
       "Specifies the end time up to which the statistical data will be generated."
       "Current time will be assumed whenever EndTime is specified null.")]
       datetime EndTime,
        [IN (false), OUT, Description("Graph object")]
        SNIAVLS_VTLStatisticalData Data);
```

GetReplicationThroughput (in_out:Uint16, CIM_ComputerSystem REF)

GetReplicationThrougput provides a historic of the incoming or outgoing replication throughput in MB/s

```
GetReplicationThrougput (
        [IN(false), OUT, Description("Reference to the job (null if synchronous call)"]
        CIM_ConcreteJob REF Job,
        [IN, Description ("Specifies the format of the 'Data' output parameter')
        ValueMap{"2"},
        Values("CSV")]
        Uint16 DataFormat,
          [IN, Description ("Reference to a VTL or the VLS")]
CIM_ComputerSystem REF Library
        [IN, Description (
        "Specifies the start time from when the data is required."
        "Both StartTime and Duration cannot be null at the same time.")]
        datetime StartTime,
        [IN, Description (
        "Specifies the duration in seconds to generate statistical data."
        "Both StartTime and Duration cannot be null at the same time.")]
        uint64 Duration,
        [IN, Description (
        "Specifies the end time up to which the statistical data will be generated."
        "Current time will be assumed whenever EndTime is specified null.")]
        datetime EndTime,
          [IN (false), OUT, Description("Graph object")]
          SNIAVLS_VTLStatisticalData Data);
```

GetReplicationRatio (CIM_ComputerSystem)

GetReplicationRatio provides a historic of the deduplication ratio over the replication link.

```
GetReplicationRatio (
          [IN(false), OUT, Description("Reference to the job (null if synchronous call)"]
          CIM_ConcreteJob REF Job,
          [IN, Description ("Specifies the format of the 'Data' output parameter')
          ValueMap{"2"},
          Values("CSV")]
          Uint16 DataFormat,
          [IN, Description ("Reference to a VTL or the VLS")]
CIM_ComputerSystem REF Library
        [IN, Description (
        "Specifies the start time from when the data is required."
        "Both StartTime and Duration cannot be null at the same time.")]
        datetime StartTime,
        [IN, Description (
        "Specifies the duration in seconds to generate statistical data."
        "Both StartTime and Duration cannot be null at the same time.")]
        uint64 Duration,
        [IN, Description (
        "Specifies the end time up to which the statistical data will be generated."
        "Current time will be assumed whenever EndTime is specified null.")]
        datetime EndTime,
          [IN (false), OUT, Description("Graph object")]
          SNIAVLS_VTLStatisticalData Data);
```

GetPerformanceStatisticsReadWriteIO (TapeDrive, CIM_ComputerSystem)

GetPerformanceStatisticsReadWriteIO gives the Tape drive performance.

```
GetPerformanceStatisticsReadWriteIO (
        [IN, Description ("Specifies the format of the 'Data' output parameter"),
        ValueMap{"2"},
        Values("CSV")]
        uint16 DataFormat,
        [IN, Description ("Reference to a VTL or the VLS")]
        CIM_ComputerSystem REF Library,
        [IN, Description (
        "Specifies the start time from when the data is required."
        "Both StartTime and Duration cannot be null at the same time.")]
        datetime StartTime,
        [IN, Description (
        "Specifies the duration in seconds to generate statistical data."
        "Both StartTime and Duration cannot be null at the same time.")]
        uint64 Duration,
        [IN, Description (
        "Specifies the end time up to which the statistical data will be generated."
        "Current time will be assumed whenever EndTime is specified null.")]
        datetime EndTime,
        [IN (false), OUT, Description("Graph object")]
        SNIAVLS_VTLStatisticalData REF Data,
        [IN, Description ("Reference to the Tape Drive for which the performance data is requested,
              if null performance statistics for the Virtual Library will be returned")]
        CIM_MediaAccessDevice REF tapeDrive,
        [IN (false), OUT, Description("Reference to the job (null if synchronous call")]
        CIM_ConcreteJob REF Job);
```

GetFCPortStatisticsSRR (CIM_LogicalPort)

GetFCPortStatisticsSRR provides the Sequence Retransmission Request performance statistics of a FC Port in a VLS.

```
GetFCPortStatisticsSRR (
        [IN, Description ("Specifies the format of the 'Data' output parameter"),
        ValueMap{"2"},
        Values("CSV")]
        uint16 DataFormat,
        [IN, Description (
        "Specifies the start time from when the data is required."
        "Both StartTime and Duration cannot be null at the same time.")]
        datetime StartTime,
        [IN, Description (
        "Specifies the duration in seconds to generate statistical data."
        "Both StartTime and Duration cannot be null at the same time.")]
        uint64 Duration,
        [IN, Description (
        "Specifies the end time up to which the statistical data will be generated."
        "Current time will be assumed whenever EndTime is specified null.")]
        datetime EndTime,
        [IN (false), OUT, Description("Graph object")]
        SNIAVLS_VTLStatisticalData REF Data,
        [IN, Description("Port for which the Aborts metric is requested")]
        CIM_LogicalPort REF port,
        [IN (false), OUT, Description("Reference to the job (null if synchronous call")]
        CIM_ConcreteJob REF Job);
```

GetFCPortStatisticsAborts (CIM_LogicalPort)

GetFCPortStatisticsAborts provides the abort statistics of a FC Port in a VLS.

```
GetFCPortStatisticsAborts (
        [IN, Description ("Specifies the format of the 'Data' output parameter"),
        ValueMap{"2"},
        Values("CSV")]
        uint16 DataFormat,
        [IN, Description (
        "Specifies the start time from when the data is required."
        "Both StartTime and Duration cannot be null at the same time.")]
        datetime StartTime,
        [IN, Description (
        "Specifies the duration in seconds to generate statistical data."
        "Both StartTime and Duration cannot be null at the same time.")]
        uint64 Duration,
        [IN, Description (
        "Specifies the end time up to which the statistical data will be generated."
        "Current time will be assumed whenever EndTime is specified null.")]
        datetime EndTime,
        [IN (false), OUT, Description("Graph object")]
        SNIAVLS_VTLStatisticalData REF Data,
        [IN, Description("Port for which the Aborts metric is requested")]
        CIM_LogicalPort REF port,
        [IN (false), OUT, Description("Reference to the job (null if synchronous call")]
        CIM_ConcreteJob REF Job);
```

14.5 Use Cases

Not in this version

14.6 CIM Elements

Table 144 describes the CIM elements for VTL Capacity and Reporting.

Table 144 - CIM Elements for VTL Capacity and Reporting

Element Name	Requirement	Description
14.6.1 CIM_ElementCapabilities (VTLStatisticalDataService to VTLStatisticalDataServiceCapabilities)	Mandatory	Associates an instance of VTLStatisticalData to its VTLStatisticalDataServiceCapabilities.
14.6.2 CIM_ElementStatisticalData (FCPort to FCPortResourceUsage)	Optional	Associates an instance of FCPort to its FCPortResourceUsage.
14.6.3 CIM_ElementStatisticalData (MediaAccessDevice to TapeDriveResourceUsage)	Optional	Associates an instance of MediaAccessDevice (Tape Drive) to its TapeDriveResourceUsage instance.
14.6.4 CIM_ElementStatisticalData (Virtual Tape Library or Virutal Library System to VTLResourceUsage)	Optional	Associates an instance of a virtual tape library or Virtual Library System ComputerSystem to its VTLResourceUsage instance.
14.6.5 CIM_FCPortResourceUsage	Optional	The CIM_FCPortResourceUsage class contains data representing FC port speed, SRR and Abort count, min, and max numbers.
14.6.6 CIM_FilterCollection (VTL Capacity and Reporting Predefined FilterCollection)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='5' (Predefined Filter Collections). This is a collection of predefined IndicationFilters to which a client may subscribe.
14.6.7 CIM_HostedCollection (VTL System to predefined FilterCollection)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='5' (Predefined Filter Collections).
14.6.8 CIM_HostedService	Mandatory	Associates an instance of a virtual library system ComputerSystem to the VTLStatisticalDataService instance.
14.6.9 CIM_IndicationFilter (DeduplicationRatioCriticalAlertThreshold)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the 'pre-defined' CIM_IndicationFilter instance for the DeduplicationRatioCriticalAlertThreshold.

Table 144 - CIM Elements for VTL Capacity and Reporting

Element Name	Requirement	Description
14.6.10 CIM_IndicationFilter (DeduplicationRatioWarningAlertThreshold)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the 'pre-defined' CIM_IndicationFilter instance for the DeduplicationRatioWarningAlertThreshold.
14.6.11 CIM_IndicationFilter (LogicalCapacityCriticalAlertThreshold)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the 'pre-defined' CIM_IndicationFilter instance for the LogicalCapacityCriticalAlertThreshold.
14.6.12 CIM_IndicationFilter (LogicalCapacityWarningAlertThreshold)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the 'pre-defined' CIM_IndicationFilter instance for the LogicalCapacityWarningAlertThreshold.
14.6.13 CIM_IndicationFilter (PhysicalCapacityCriticalAlertThreshold)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the 'pre-defined' CIM_IndicationFilter instance for the PhysicalCapacityCriticalAlertThreshold.
14.6.14 CIM_IndicationFilter (PhysicalCapacityWarningAlertThreshold)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the 'pre-defined' CIM_IndicationFilter instance for the PhysicalCapacityWarningAlertThreshold.
14.6.15 CIM_IndicationFilter (ReplicationTrafficCriticalAlertThreshold)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the 'pre-defined' CIM_IndicationFilter instance for the ReplicationTrafficCriticalAlertThreshold.

Table 144 - CIM Elements for VTL Capacity and Reporting

Element Name	Requirement	Description
14.6.16 CIM_IndicationFilter (ReplicationTrafficWarningAlertThreshold)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the 'pre-defined' CIM_IndicationFilter instance for the ReplicationTrafficWarningAlertThreshold.
14.6.17 CIM_IndicationFilter (SystemRatioCriticalAlertThreshold)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the 'pre-defined' CIM_IndicationFilter instance for the SystemRatioCriticalAlertThreshold.
14.6.18 CIM_IndicationFilter (SystemRatioWarningAlertThreshold)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the 'pre-defined' CIM_IndicationFilter instance for the SystemRatioWarningAlertThreshold.
14.6.19 CIM_IndicationFilter (ThroughputCriticalAlertThreshold)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the 'pre-defined' CIM_IndicationFilter instance for the ThroughputCriticalAlertThreshold.
14.6.20 CIM_IndicationFilter (ThroughputWarningAlertThreshold)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the 'pre-defined' CIM_IndicationFilter instance for the ThroughputWarningAlertThreshold.

Table 144 - CIM Elements for VTL Capacity and Reporting

Element Name	Requirement	Description
14.6.21 CIM_MemberOfCollection (Predefined Filter Collection to VTL Capacity and Reporting Filters)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='5' (Predefined Filter Collections). This associates the VTL Capacity and Reporting predefined FilterCollection to the predefined Filters supported by the VTL Capacity and Reporting implementation.
14.6.22 CIM_TapeDriveResourceUsage	Optional	The CIM_TapeDriveResourceUsage class contains data representing Tape Drive Read and Write speed.
14.6.23 CIM_VTLResourceUsage	Optional	The CIM_VTLResourceUsage class provides the ability to track the current resource usage of the device in order to detect when the device resources are approaching their limits (and thus further device hardware is needed or some backups needs to be moved to another device). All these device resources need to be tracked at both the whole-device level but also at the virtual library level. Tracking resource usage at the library level is critical because the VLS can be partitioned into multiple logical libraries that use physically separate resources (e.g. different front-end FC ports, different storage pools) so if the device shows 50% full this could mean one library is 100% full and another library is 0% full. It contains data representing used capacity, compression ratios and deduplication ratios at VTL or VLS level.also support the ability to set resource thresholds so that alerts are generated when the device approaches a warning or critical resource threshold (to alert the user they may need to add hardware or reduce workload).
14.6.24 CIM_VTLStatisticalData	Mandatory	The CIM_VTLStatisticalData class is used to represent graph contains data representing the history of specified resource via CIM_VTLStatisticalDataService methods.

Table 144 - CIM Elements for VTL Capacity and Reporting

Element Name	Requirement	Description
14.6.25 CIM_VTLStatisticalDataService	Mandatory	The CIM_VTLStatisticalDataService class provides Historical Resource Usage Service. This class has set of functions to collect historical values for Front-end port performance, Physical disk usage, Logical backup data, Overall system compression ratio, Deduplication efficiency and Replication traffic. All the functions returns graph object Data point history of specified resource.
14.6.26 CIM_VTLStatisticalDataServiceCapabilities	Mandatory	Virtual Library System Statistical Data Capabilities.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity="SNIA" and MessageID="SML72"	Optional	CQL -This alert is issued when the ThroughputWarningAlertThreshold is exceeded. See *Storage Management Technical Specification, Part 7 Media Libraries, 1.6.0 Rev 4* 14.6.20 CIM_IndicationFilter (ThroughputWarningAlertThreshold) Also see *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.72 Message: Throughput Threshold Warning Alert.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity="SNIA" and MessageID="SML73"	Optional	CQL -This alert is issued when the ThroughputCriticalAlertThreshold is exceeded. See *Storage Management Technical Specification, Part 7 Media Libraries, 1.6.0 Rev 4* 14.6.19 CIM_IndicationFilter (ThroughputCriticalAlertThreshold) Also see *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.73 Message: Throughput Threshold Critical Alert.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity="SNIA" and MessageID="SML74"	Optional	CQL -This alert is issued when the PhysicalCapacityWarningAlertThreshold is exceeded. See *Storage Management Technical Specification, Part 7 Media Libraries, 1.6.0 Rev 4* 14.6.14 CIM_IndicationFilter (PhysicalCapacityWarningAlertThreshold) Also see *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.74 Message: Physical Capacity Threshold Warning Alert.

Table 144 - CIM Elements for VTL Capacity and Reporting

Element Name	Requirement	Description
SELECT * FROM CIM_AlertIndication WHERE OwningEntity="SNIA" and MessageID="SML75"	Optional	CQL -This alert is issued when the PhysicalCapacityCriticalAlertThreshold is exceeded. See *Storage Management Technical Specification, Part 7 Media Libraries, 1.6.0 Rev 4* 14.6.13 CIM_IndicationFilter (PhysicalCapacityCriticalAlertThreshold) Also see *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.75 Message: Physical Capacity Threshold Critical Alert.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity="SNIA" and MessageID="SML76"	Optional	CQL -This alert is issued when the LogicalCapacityWarningAlertThreshold is exceeded. See *Storage Management Technical Specification, Part 7 Media Libraries, 1.6.0 Rev 4* 14.6.12 CIM_IndicationFilter (LogicalCapacityWarningAlertThreshold) Also see *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.76 Message: Logical Capacity Threshold Warning Alert.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity="SNIA" and MessageID="SML77"	Optional	CQL -This alert is issued when the LogicalCapacityCriticalAlertThreshold is exceeded. See *Storage Management Technical Specification, Part 7 Media Libraries, 1.6.0 Rev 4* 14.6.11 CIM_IndicationFilter (LogicalCapacityCriticalAlertThreshold) Also see *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.77 Message: Logical Capacity Threshold Critical Alert.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity="SNIA" and MessageID="SML78"	Optional	CQL -This alert is issued when the system ratio falls below the SystemRatioWarningAlertThreshold. See *Storage Management Technical Specification, Part 7 Media Libraries, 1.6.0 Rev 4* 14.6.18 CIM_IndicationFilter (SystemRatioWarningAlertThreshold) Also see *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.78 Message: System Ratio Threshold Warning Alert.

Table 144 - CIM Elements for VTL Capacity and Reporting

Element Name	Requirement	Description
SELECT * FROM CIM_AlertIndication WHERE OwningEntity="SNIA" and MessageID="SML79"	Optional	CQL -This alert is issued when the system ratio falls below the SystemRatioCriticalAlertThreshold. See *Storage Management Technical Specification, Part 7 Media Libraries, 1.6.0 Rev 4* 14.6.17 CIM_IndicationFilter (SystemRatioCriticalAlertThreshold) Also see *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.79 Message: System Ratio Threshold Critical Alert.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity="SNIA" and MessageID="SML80"	Optional	CQL -This alert is issued when the deduplication ratio falls below the DeduplicationRatioWarningAlertThreshold. See *Storage Management Technical Specification, Part 7 Media Libraries, 1.6.0 Rev 4* 14.6.10 CIM_IndicationFilter (DeduplicationRatioWarningAlertThreshold) Also see *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.80 Message: Deduplication Ratio Threshold Warning Alert.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity="SNIA" and MessageID="SML81"	Optional	CQL -This alert is issued when the deduplication ratio falls below the DeduplicationRatioCriticalAlertThreshold. See *Storage Management Technical Specification, Part 7 Media Libraries, 1.6.0 Rev 4* 14.6.9 CIM_IndicationFilter (DeduplicationRatioCriticalAlertThreshold) Also see *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.81 Message: Deduplication Ratio Threshold Critical Alert.

Table 144 - CIM Elements for VTL Capacity and Reporting

Element Name	Requirement	Description
SELECT * FROM CIM_AlertIndication WHERE OwningEntity="SNIA" and MessageID="SML82"	Optional	CQL -This alert is issued when the ReplicationTrafficWarningAlertThreshold is exceeded. See *Storage Management Technical Specification, Part 7 Media Libraries, 1.6.0 Rev 4* 14.6.16 CIM_IndicationFilter (ReplicationTrafficWarningAlertThreshold) Also see *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.82 Message: Replication Traffic Threshold Warning Alert.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity="SNIA" and MessageID="SML83"	Optional	CQL -This alert is issued when the ReplicationTrafficCriticalAlertThreshold is exceeded. See *Storage Management Technical Specification, Part 7 Media Libraries, 1.6.0 Rev 4* 14.6.15 CIM_IndicationFilter (ReplicationTrafficCriticalAlertThreshold) Also see *Storage Management Technical Specification, Part 1 Common Architecture, 1.6.0 Rev 4* 8.4.3.83 Message: Replication Traffic Threshold Critical Alert.

14.6.1 CIM_ElementCapabilities (VTLStatisticalDataService to VTLStatisticalDataServiceCapabilities)

Associates an instance of VTLStatisticalData to its VTLStatisticalDataServiceCapabilities.

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

Table 145 describes class CIM_ElementCapabilities (VTLStatisticalDataService to VTLStatisticalDataServiceCapabilities).

Table 145 - SMI Referenced Properties/Methods for CIM_ElementCapabilities (VTLStatisticalDataService to VTLStatisticalDataServiceCapabilities)

Properties	Flags	Requirement	Description & Notes
ManagedElement		Mandatory	A reference to an instance of the VTLStatisticalDataService.
Capabilities		Mandatory	A reference to an VTLStatisticalDataServiceCapabilities instance.

14.6.2 CIM_ElementStatisticalData (FCPort to FCPortResourceUsage)

Associates an instance of FCPort to its FCPortResourceUsage.

Created By: Static
Modified By: Static
Deleted By: Static
Requirement: Optional

Table 146 describes class CIM_ElementStatisticalData (FCPort to FCPortResourceUsage).

Table 146 - SMI Referenced Properties/Methods for CIM_ElementStatisticalData (FCPort to FCPortResourceUsage)

Properties	Flags	Requirement	Description & Notes
ManagedElement		Mandatory	A reference to an instance of FCPort.
Stats		Mandatory	A reference to an FCPortResourceUsage instance.

14.6.3 CIM_ElementStatisticalData (MediaAccessDevice to TapeDriveResourceUsage)

Associates an instance of MediaAccessDevice (Tape Drive) to its TapeDriveResourceUsage instance.

Created By: Static
Modified By: Static
Deleted By: Static
Requirement: Optional

Table 147 describes class CIM_ElementStatisticalData (MediaAccessDevice to TapeDriveResourceUsage).

Table 147 - SMI Referenced Properties/Methods for CIM_ElementStatisticalData (MediaAccessDevice to TapeDriveResourceUsage)

Properties	Flags	Requirement	Description & Notes
ManagedElement		Mandatory	A reference to an instance of MediaAccessDevice (Tape Drive).
Stats		Mandatory	A reference to an CIM_TapeDriveResourceUsage instance.

14.6.4 CIM_ElementStatisticalData (Virtual Tape Library or Virutal Library System to VTLResourceUsage)

Associates an instance of a virtual tape library or virtual library system ComputerSystem to its VTLResourceUsage instance.

Created By: Static
Modified By: Static
Deleted By: Static
Requirement: Optional

Table 148 describes class CIM_ElementStatisticalData (Virtual Tape Library or Virutal Library System to VTLResourceUsage).

Table 148 - SMI Referenced Properties/Methods for CIM_ElementStatisticalData (Virtual Tape Library or Virutal Library System to VTLResourceUsage)

Properties	Flags	Requirement	Description & Notes
ManagedElement		Mandatory	A reference to a Virtual Tape Library or Virtual Library System ComputerSystem.
Stats		Mandatory	A reference to an VTLResourceUsage instance.

14.6.5 CIM_FCPortResourceUsage

The CIM_FCPortResourceUsage class contains data representing FC port speed, SRR and Abort count, min, and max numbers.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Optional

Table 149 describes class CIM_FCPortResourceUsage.

Table 149 - SMI Referenced Properties/Methods for CIM_FCPortResourceUsage

Properties	Flags	Requirement	Description & Notes
InstanceID		Mandatory	An opaque string that uniquely identifies instances of this class.
StatisticsTime		Mandatory	The timestamp for the data contained in this instance.
ElementName		Optional	A user friendly name for the statistics instance.
PortNumber		Mandatory	The number assigned to this FCPort.
PortActive		Optional	This Boolean is set to TRUE if this FCPort is active.
PortSpeed		Optional	The Speed of this FC Port.
SRRCount		Optional	The number of SRRs (Sequence Retransmission Requests) for this FCPort.
AbortCount		Optional	The number of Aborts for this FCPort.
MinSRRCount		Optional	The minimum number of SRRs for this FCPort.
MaxSRRCount		Optional	The maximum number of SRRs for this FCPort.
MinAbortCount		Optional	The minimum number of Aborts for this FCPort.
MaxAbortCount		Optional	The maximum number of Aborts for this FCPort.

14.6.6 CIM_FilterCollection (VTL Capacity and Reporting Predefined FilterCollection)

Experimental. This is a collection of predefined IndicationFilters to which a client may subscribe. A VTL Capacity and Reporting implementation shall indicate support for predefined FilterCollections by the SNIA_IndicationConfigurationCapabilities.FeaturesSupported = '5' (Predefined Filter Collections).

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='5' (Predefined Filter Collections).

Table 150 describes class CIM_FilterCollection (VTL Capacity and Reporting Predefined FilterCollection).

Table 150 - SMI Referenced Properties/Methods for CIM_FilterCollection (VTL Capacity and Reporting Predefined FilterCollection)

Properties	Flags	Requirement	Description & Notes
InstanceID		Mandatory	Shall specify the unique identifier for an instance of this class within the Implementation namespace.
CollectionName		Mandatory	The value of CollectionName shall be 'SNIA:VTL Capacity and Reporting'.

14.6.7 CIM_HostedCollection (VTL System to predefined FilterCollection)

Experimental.

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='5' (Predefined Filter Collections).

Table 151 describes class CIM_HostedCollection (VTL System to predefined FilterCollection).

Table 151 - SMI Referenced Properties/Methods for CIM_HostedCollection (VTL System to predefined FilterCollection)

Properties	Flags	Requirement	Description & Notes
Dependent		Mandatory	Reference to the predefined FilterCollection for the VTL Capacity and Reporting.
Antecedent		Mandatory	A reference to a Virtual Library System ComputerSystem.

14.6.8 CIM_HostedService

Associates an instance of a virtual library system ComputerSystem to the VTLStatisticalDataService instance.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 152 describes class CIM_HostedService.

Table 152 - SMI Referenced Properties/Methods for CIM_HostedService

Properties	Flags	Requirement	Description & Notes
Dependent		Mandatory	A reference to a Virtual Library System ComputerSystem.
Antecedent		Mandatory	A reference to an VTLStatisticalDataService instance.

14.6.9 CIM_IndicationFilter (DeduplicationRatioCriticalAlertThreshold)

Experimental. This is the 'pre-defined' CIM_IndicationFilter instance for the DeduplicationRatioCriticalAlertThreshold. The alert defined by this filter will be triggered when the deduplication ratio falls below the DeduplicationRatioCriticalAlertThreshold. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters).

Table 153 describes class CIM_IndicationFilter (DeduplicationRatioCriticalAlertThreshold).

Table 153 - SMI Referenced Properties/Methods for CIM_IndicationFilter (DeduplicationRatioCriticalAlertThreshold)

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:VTL Capacity and Reporting:DeduplicationRatioCriticalAlertThreshold'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).

Table 153 - SMI Referenced Properties/Methods for CIM_IndicationFilter (DeduplicationRatioCriticalAlertThreshold)

Properties	Flags	Requirement	Description & Notes
Query		Mandatory	SELECT * FROM CIM_AlertIndication WHERE OwningEntity="SNIA" and MessageID="SML81".
QueryLanguage		Mandatory	This shall be 'DMTF:CQL'.
ElementName	N	Optional	See the ElementName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).

14.6.10 CIM_IndicationFilter (DeduplicationRatioWarningAlertThreshold)

Experimental. This is the 'pre-defined' CIM_IndicationFilter instance for the DeduplicationRatioWarningAlertThreshold. The alert defined by this filter will be triggered when the deduplication ratio falls below the DeduplicationRatioWarningAlertThreshold. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters).

Table 154 describes class CIM_IndicationFilter (DeduplicationRatioWarningAlertThreshold).

Table 154 - SMI Referenced Properties/Methods for CIM_IndicationFilter (DeduplicationRatioWarningAlertThreshold)

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:VTL Capacity and Reporting:DeduplicationRatioWarningAlertThreshold'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).

Table 154 - SMI Referenced Properties/Methods for CIM_IndicationFilter (DeduplicationRatio-WarningAlertThreshold)

Properties	Flags	Requirement	Description & Notes
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
Query		Mandatory	SELECT * FROM CIM_AlertIndication WHERE OwningEntity="SNIA" and MessageID="SML80".
QueryLanguage		Mandatory	This shall be 'DMTF:CQL'.
ElementName	N	Optional	See the ElementName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).

14.6.11 CIM_IndicationFilter (LogicalCapacityCriticalAlertThreshold)

Experimental. This is the 'pre-defined' CIM_IndicationFilter instance for the LogicalCapacityCriticalAlertThreshold. The alert defined by this filter will be triggered when the LogicalCapacityCriticalAlertThreshold has been exceeded. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters).

Table 155 describes class CIM_IndicationFilter (LogicalCapacityCriticalAlertThreshold).

Table 155 - SMI Referenced Properties/Methods for CIM_IndicationFilter (LogicalCapacityCriticalAlertThreshold)

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:VTL Capacity and Reporting:LogicalCapacityCriticalAlertThreshold'.

Table 155 - SMI Referenced Properties/Methods for CIM_IndicationFilter (LogicalCapacityCriticalAlertThreshold)

Properties	Flags	Requirement	Description & Notes
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
Query		Mandatory	SELECT * FROM CIM_AlertIndication WHERE OwningEntity="SNIA" and MessageID="SML77".
QueryLanguage		Mandatory	This shall be 'DMTF:CQL'.
ElementName	N	Optional	See the ElementName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).

14.6.12 CIM_IndicationFilter (LogicalCapacityWarningAlertThreshold)

Experimental. This is the 'pre-defined' CIM_IndicationFilter instance for the LogicalCapacityWarningAlertThreshold. The alert defined by this filter will be triggered when the LogicalCapacityWarningAlertThreshold has been exceeded. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters).

Table 156 describes class CIM_IndicationFilter (LogicalCapacityWarningAlertThreshold).

Table 156 - SMI Referenced Properties/Methods for CIM_IndicationFilter (LogicalCapacityWarningAlertThreshold)

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).

Table 156 - SMI Referenced Properties/Methods for CIM_IndicationFilter (LogicalCapacityWarningAlertThreshold)

Properties	Flags	Requirement	Description & Notes
SystemName		Mandatory	See the SystemName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:VTL Capacity and Reporting:LogicalCapacityWarningAlertThreshold'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
Query		Mandatory	SELECT * FROM CIM_AlertIndication WHERE OwningEntity="SNIA" and MessageID="SML76".
QueryLanguage		Mandatory	This shall be 'DMTF:CQL'.
ElementName	N	Optional	See the ElementName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).

14.6.13 CIM_IndicationFilter (PhysicalCapacityCriticalAlertThreshold)

Experimental. This is the 'pre-defined' CIM_IndicationFilter instance for the PhysicalCapacityCriticalAlertThreshold. The alert defined by this filter will be triggered when the PhysicalCapacityCriticalAlertThreshold has been exceeded. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters).

Table 157 describes class CIM_IndicationFilter (PhysicalCapacityCriticalAlertThreshold).

Table 157 - SMI Referenced Properties/Methods for CIM_IndicationFilter (PhysicalCapacityCriticalAlertThreshold)

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:VTL Capacity and Reporting:PhysicalCapacityCriticalAlertThreshold'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
Query		Mandatory	SELECT * FROM CIM_AlertIndication WHERE OwningEntity="SNIA" and MessageID="SML75".
QueryLanguage		Mandatory	This shall be 'DMTF:CQL'.
ElementName	N	Optional	See the ElementName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).

14.6.14 CIM_IndicationFilter (PhysicalCapacityWarningAlertThreshold)

Experimental. This is the 'pre-defined' CIM_IndicationFilter instance for the PhysicalCapacityWarningAlertThreshold. The alert defined by this filter will be triggered when the PhysicalCapacityWarningAlertThreshold has been exceeded. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters).

Table 158 describes class CIM_IndicationFilter (PhysicalCapacityWarningAlertThreshold).

Table 158 - SMI Referenced Properties/Methods for CIM_IndicationFilter (PhysicalCapacityWarningAlertThreshold)

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:VTL Capacity and Reporting:PhysicalCapacityWarningAlertThreshold'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
Query		Mandatory	SELECT * FROM CIM_AlertIndication WHERE OwningEntity="SNIA" and MessageID="SML74".
QueryLanguage		Mandatory	This shall be 'DMTF:CQL'.
ElementName	N	Optional	See the ElementName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).

14.6.15 CIM_IndicationFilter (ReplicationTrafficCriticalAlertThreshold)

Experimental. This is the 'pre-defined' CIM_IndicationFilter instance for the ReplicationTrafficCriticalAlertThreshold. The alert defined by this filter will be triggered when the ReplicationTrafficCriticalAlertThreshold has been exceeded. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters).

Table 159 describes class CIM_IndicationFilter (ReplicationTrafficCriticalAlertThreshold).

Table 159 - SMI Referenced Properties/Methods for CIM_IndicationFilter (ReplicationTrafficCriticalAlertThreshold)

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:VTL Capacity and Reporting:ReplicationTrafficCriticalAlertThreshold'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
Query		Mandatory	SELECT * FROM CIM_AlertIndication WHERE OwningEntity="SNIA" and MessageID="SML83".
QueryLanguage		Mandatory	This shall be 'DMTF:CQL'.
ElementName	N	Optional	See the ElementName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).

14.6.16 CIM_IndicationFilter (ReplicationTrafficWarningAlertThreshold)

Experimental. This is the 'pre-defined' CIM_IndicationFilter instance for the ReplicationTrafficWarningAlertThreshold. The alert defined by this filter will be triggered when the ReplicationTrafficWarningAlertThreshold has been exceeded. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters).

Table 160 describes class CIM_IndicationFilter (ReplicationTrafficWarningAlertThreshold).

Table 160 - SMI Referenced Properties/Methods for CIM_IndicationFilter (ReplicationTrafficWarningAlertThreshold)

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:VTL Capacity and Reporting:ReplicationTrafficWarningAlertThreshold'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
Query		Mandatory	SELECT * FROM CIM_AlertIndication WHERE OwningEntity="SNIA" and MessageID="SML82".
QueryLanguage		Mandatory	This shall be 'DMTF:CQL'.
ElementName	N	Optional	See the ElementName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).

14.6.17 CIM_IndicationFilter (SystemRatioCriticalAlertThreshold)

Experimental. This is the 'pre-defined' CIM_IndicationFilter instance for the SystemRatioCriticalAlertThreshold. The alert defined by this filter will be triggered when the system ratio falls below the SystemRatioCriticalAlertThreshold. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters).

Table 161 describes class CIM_IndicationFilter (SystemRatioCriticalAlertThreshold).

Table 161 - SMI Referenced Properties/Methods for CIM_IndicationFilter (SystemRatioCriticalAlertThreshold)

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:VTL Capacity and Reporting:SystemRatioCriticalAlertThreshold'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
Query		Mandatory	SELECT * FROM CIM_AlertIndication WHERE OwningEntity="SNIA" and MessageID="SML79".
QueryLanguage		Mandatory	This shall be 'DMTF:CQL'.
ElementName	N	Optional	See the ElementName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).

14.6.18 CIM_IndicationFilter (SystemRatioWarningAlertThreshold)

Experimental. This is the 'pre-defined' CIM_IndicationFilter instance for the SystemRatioWarningAlertThreshold. The alert defined by this filter will be triggered when the system ratio falls below the SystemRatioWarningAlertThreshold. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters).

Table 162 describes class CIM_IndicationFilter (SystemRatioWarningAlertThreshold).

Table 162 - SMI Referenced Properties/Methods for CIM_IndicationFilter (SystemRatioWarningAlertThreshold)

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:VTL Capacity and Reporting:SystemRatioWarningAlertThreshold'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
Query		Mandatory	SELECT * FROM CIM_AlertIndication WHERE OwningEntity="SNIA" and MessageID="SML78".
QueryLanguage		Mandatory	This shall be 'DMTF:CQL'.
ElementName	N	Optional	See the ElementName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).

14.6.19 CIM_IndicationFilter (ThroughputCriticalAlertThreshold)

Experimental. This is the 'pre-defined' CIM_IndicationFilter instance for the ThroughputCriticalAlertThreshold. The alert defined by this filter will be triggered when the ThroughputCriticalAlertThreshold has been exceeded. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters).

Table 163 describes class CIM_IndicationFilter (ThroughputCriticalAlertThreshold).

Table 163 - SMI Referenced Properties/Methods for CIM_IndicationFilter (ThroughputCriticalAlertThreshold)

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:VTL Capacity and Reporting:ThroughputCriticalAlertThreshold'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
Query		Mandatory	SELECT * FROM CIM_AlertIndication WHERE OwningEntity="SNIA" and MessageID="SML73".
QueryLanguage		Mandatory	This shall be 'DMTF:CQL'.
ElementName	N	Optional	See the ElementName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).

14.6.20 CIM_IndicationFilter (ThroughputWarningAlertThreshold)

Experimental. This is the 'pre-defined' CIM_IndicationFilter instance for the ThroughputWarningAlertThreshold. The alert defined by this filter will be triggered when the ThroughputWarningAlertThreshold has been exceeded. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters).

Table 164 describes class CIM_IndicationFilter (ThroughputWarningAlertThreshold).

Table 164 - SMI Referenced Properties/Methods for CIM_IndicationFilter (ThroughputWarningAlertThreshold)

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:VTL Capacity and Reporting:ThroughputWarningAlertThreshold'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).
Query		Mandatory	SELECT * FROM CIM_AlertIndication WHERE OwningEntity="SNIA" and MessageID="SML72".
QueryLanguage		Mandatory	This shall be 'DMTF:CQL'.
ElementName	N	Optional	See the ElementName definition in section *Storage Management Technical Specification, Part 2 Common Profiles, 1.6.0 Rev 4* 42.8.3 CIM_IndicationFilter (pre-defined).

14.6.21 CIM_MemberOfCollection (Predefined Filter Collection to VTL Capacity and Reporting Filters)

Experimental. This associates the VTL Capacity and Reporting predefined FilterCollection to the predefined Filters supported by the VTL Capacity and Reporting implementation.

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='5' (Predefined Filter Collections).

Table 165 describes class CIM_MemberOfCollection (Predefined Filter Collection to VTL Capacity and Reporting Filters).

Table 165 - SMI Referenced Properties/Methods for CIM_MemberOfCollection (Predefined Filter Collection to VTL Capacity and Reporting Filters)

Properties	Flags	Requirement	Description & Notes
Collection		Mandatory	Reference to the VTL Capacity and Reporting predefined FilterCollection.
Member		Mandatory	Reference to the predefined IndicationFilters of the VTL Capacity and Reporting implementation.

14.6.22 CIM_TapeDriveResourceUsage

The CIM_TapeDriveResourceUsage class contains data representing Tape Drive Read and Write speed.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Optional

Table 166 describes class CIM_TapeDriveResourceUsage.

Table 166 - SMI Referenced Properties/Methods for CIM_TapeDriveResourceUsage

Properties	Flags	Requirement	Description & Notes
InstanceID		Mandatory	An opaque string that uniquely identifies instances of this class.
StatisticsTime		Mandatory	The timestamp for the data contained in this instance.
ElementName		Optional	A user friendly name for the statistics instance.
ReadSpeed		Mandatory	Speed in Mbytes/sec.
WriteSpeed		Mandatory	Speed in Mbytes/sec.

14.6.23 CIM_VTLResourceUsage

The CIM_VTLResourceUsage class provides the ability to track the current resource usage of the device in order to detect when the device resources are approaching their limits (and thus further device hardware is needed or some backups needs to be moved to another device). All these device resources need to be tracked at both the whole-device level but also at the virtual library level. Tracking resource usage at the library level is critical because the VLS can be partitioned into multiple logical libraries that use physically separate resources (e.g. different front-end FC ports, different storage pools) so if the device shows 50% full this could mean one library is 100% full and another library is 0% full. It contains data representing used capacity, compression ratios and deduplication ratios at VTL or VLS level.also support the ability to set resource thresholds so that alerts are generated when the device approaches a warning or critical resource threshold (to alert the user they may need to add hardware or reduce workload).

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Optional

Table 167 describes class CIM_VTLResourceUsage.

Table 167 - SMI Referenced Properties/Methods for CIM_VTLResourceUsage

Properties	Flags	Requirement	Description & Notes
InstanceID		Mandatory	An opaque string that uniquely identifies instances of this class.
StatisticsTime		Mandatory	The timestamp for the data contained in this instance.
ElementName		Optional	A user friendly name for the statistics instance.
Throughput		Optional	Performance in Mbytes/sec.
RelativeThroughput		Optional	Performance in percentage of the maximum performance allowed for this VTL or VLS. Must be a floating point value between 0 and 1.
UsedPhysicalCapacity		Optional	Used physical capacity in Mbytes. Physical capacity is the amount of data actually stored on the disk, whether or not deduplicated.
UsedLogicalCapacity		Optional	Used logical capacity in Mbytes. Logical capacity is the amount of data the backup application wrote (before deduplication and compression), while the physical capacity is the amount of data actually stored on the disk.
UsedPhysicalCapacityOfMax		Optional	Used physical disk capacity as a percentage value of the maximum physical disk capacity available for this VTL or VLS. Must be a floating point value between 0 and 1.
UsedLogicalCapacityOfMax		Optional	Used logical disk capacity as a percentage value of the maximum physical disk capacity available for this VTL or VLS. Must be a floating point value between 0 and 1.
SystemRatio		Optional	Compression ratio for this VTL or VLS.The ratio of the Logical Data divided by the Used Capacity.
DeduplicationRatio		Optional	Deduplication ratio for this VTL or VLS. The storage capacity saved by deduplication , where the sum of all pre-deduplicated backup data is compared with the actual amount of storage the deduplicated data requires. For example, a value 10 means that ten times more data is being stored than the actual physical space it would require.
InReplicationThroughput		Optional	Incoming replication traffic in Mbytes/s.
OutReplicationThroughput		Optional	Outgoing replication traffic in Mbytes/s.
ThresholdCheckInterval		Optional	Expressed in seconds, this field indicates the frequency used to compare threshold to actual values.

Table 167 - SMI Referenced Properties/Methods for CIM_VTLResourceUsage

Properties	Flags	Requirement	Description & Notes
ThresholdEnabled		Optional	This field indicates if Threshold enabled or not.
ThroughputAlertsEnabled		Optional	A true value indicates that throughput alerts governed by ThroughputWarningAlertThreshold and ThroughputCriticalAlertThreshold will be issued.
ThroughputWarningAlertThreshold		Optional	Throughput threshold after which a warning alert will be triggered. Readable and writable. Must be a floating point value between 0 and 1.
ThroughputCriticalAlertThreshold		Optional	Throughput threshold after which a critical alert will be triggered.Readable and writable. Must be a floating point value between 0 and 1.
PhysicalCapacityAlertsEnabled		Optional	A true value indicates that physical capacity alerts governed by PhysicalCapacityWarningAlertThreshold and PhysicalCapacityCriticalAlertThreshold will be issued.
PhysicalCapacityWarningAlertThreshold		Optional	Physical capacity threshold after which a warning alert will be triggered. Readable and writable. Must be a floating point value between 0 and 1.
PhysicalCapacityCriticalAlertThreshold		Optional	Physical capacity threshold after which a critical alert will be triggered. Readable and writable. Must be a floating point value between 0 and 1.
LogicalCapacityAlertsEnabled		Optional	A true value indicates that logical capacity alerts governed by LogicalCapacityWarningAlertThreshold and LogicalCapacityCriticalAlertThreshold will be issued.
LogicalCapacityWarningAlertThreshold		Optional	Logical capacity threshold after which a warning alert will betriggered. Readable and writable. Must be a floating point value between 0 and 1.
LogicalCapacityCriticalAlertThreshold		Optional	Logical capacity threshold after which a critical alert will betriggered. Readable and writable. Must be a floating point value between 0 and 1.
SystemRatioAlertsEnabled		Optional	A true value indicates that system ratio alerts governed by SystemRatioWarningAlertThreshold and SystemRatioCriticalAlertThreshold will be issued.
SystemRatioWarningAlertThreshold		Optional	A warning-level alert will be issued if the system ratio falls below the specified threshold.
SystemRatioCriticalAlertThreshold		Optional	A critical-level alert will be issued if the system ratio falls below the specified threshold.
DeduplicationRatioAlertsEnabled		Optional	A true value indicates that deduplication ratio alerts governed by DeduplicationRatioWarningAlertThreshold and DeduplicationRatioPhysicalCapacityCriticalAlertThreshold will be issued.
DeduplicationRatioWarningAlertThreshold		Optional	A warning-level alert will be issued if the deduplication ratio falls below the specified threshold.

Table 167 - SMI Referenced Properties/Methods for CIM_VTLResourceUsage

Properties	Flags	Requirement	Description & Notes
DeduplicationRatioCriticalAlertThreshold		Optional	A critical-level alert will be issued if the deduplication ratio falls below the specified threshold.
ReplicationTrafficAlertsEnabled		Optional	A true value indicates that replication traffic alerts governed by ReplicationTrafficWarningAlertThresholdand ReplicationTrafficCriticalAlertThreshold will be issued.
ReplicationTrafficWarningAlertThreshold		Optional	A warning-level alert will be issued if the replication traffic is beyond the specified threshold.
ReplicationTrafficCriticalAlertThreshold		Optional	A critical-level alert will be issued if the replication traffic is beyond the specified threshold.

14.6.24 CIM_VTLStatisticalData

The CIM_VTLStatisticalData class is used to represent graph contains data representing the history of specified resource via CIM_VTLStatisticalDataService methods.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 168 describes class CIM_VTLStatisticalData.

Table 168 - SMI Referenced Properties/Methods for CIM_VTLStatisticalData

Properties	Flags	Requirement	Description & Notes
InstanceID		Mandatory	An opaque string that uniquely identifies instances of this class.
StatisticsTime		Mandatory	The timestamp for the data contained in this instance.
ElementName		Optional	A user friendly name for the statistics instance.
Data		Optional	A string that contains the graph data In CSV format.
DataFormat		Optional	The format of the data as requested in the graph generation methods. Ex: \2\'(CSV).'
XAxisName		Optional	The name of the X axis.
YAxisName		Optional	The name of the Y axis.
XAxisUnit		Optional	The unit used for the X axis (e.g. seconds).
YAxisUnit		Optional	The unit used for the Y axis (e.g. Mbytes/s).
StartTime		Optional	Begin time of the data point collection.
TimeInterval		Optional	The time interval in seconds between consecutive points in the data returned.

14.6.25 CIM_VTLStatisticalDataService

The CIM_VTLStatisticalDataService class provides Historical Resource Usage Service. This class has set of functions to collect historical values for Front-end port performance, Physical disk usage, Logical backup data, Overall system compression ratio, Deduplication efficiency and Replication traffic. All the functions returns graph object Data point history of specified resource.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 169 describes class CIM_VTLStatisticalDataService.

Table 169 - SMI Referenced Properties/Methods for CIM_VTLStatisticalDataService

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	Key.
CreationClassName		Mandatory	Key.
SystemName		Mandatory	Key.
Name		Mandatory	Key.
GetPhysicalCapacityUsage()		Optional	GetPhysicalCapacityUsage Provides capacity on a per storage pool and virtual library basis. If the primordial pool is used, data is returned for all pools. If the VLS Computer System is used, capacity is returned for the entire appliance.
GetLogicalCapacityUsage()		Optional	GetLogicalCapacityUsage Provides logical capacity usage for the entire VLS appliance or a specific VTL depending on the ComputerSystem reference passed in.
GetLogicalCapacityUsage()		Optional	GetLogicalCapacityUsage Provides logical capacity usage for the entire VLS appliance or a specific VTL depending on the ComputerSystem reference passed in.
GetSystemRatio()		Optional	GetSystemRatio Provides an historic of the compression ratio for the entire VLS appliance or a specific VTL depending on on the ComputerSystem reference passed in.
GetDeduplicationRatio()		Optional	GetDeduplicationRatio(CIM_ComputerSystem REF): Provides an historicof the deduplication ratio for the entire VLS appliance or a specificVTL depending on on the ComputerSystem reference passed in.
GetReplicationThroughput()		Optional	GetReplicationThroughput provides a historic of the incoming or outgoing replication throughput in MB/s.
GetReplicationRatio()		Optional	GetReplicationRatio provides a historic of the deduplication ratio over the replication link.
GetPerformanceStatisticsReadWriteIO()		Optional	GetPerformanceStatisticsReadWriteIO gives the front-end port performance.

Table 169 - SMI Referenced Properties/Methods for CIM_VTLStatisticalDataService

Properties	Flags	Requirement	Description & Notes
GetFCPortStatisticsSRR()		Optional	GetFCPortStatisticsSRR provides the Sequence RetransmissionRequest performance statistics of a FC Port in a VLS.
GetFCPortStatisticsAborts()		Optional	GetFCPortStatisticsAborts provides the abort statistics of a FC Port in a VLS.

14.6.26 CIM_VTLStatisticalDataServiceCapabilities

Virtual Library System Statistical Data Capabilities.

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

Table 170 describes class CIM_VTLStatisticalDataServiceCapabilities.

Table 170 - SMI Referenced Properties/Methods for CIM_VTLStatisticalDataServiceCapabilities

Properties	Flags	Requirement	Description & Notes
InstanceID		Mandatory	An opaque string that uniquely identifies instances of this class.
SupportedSyncActions		Optional	SupportedSyncActions lists the statistical services implemented synchronously. The valid values are "0" (GetPhysicalCapacityUsage), "1" (GetLogicalCapacityUsage), "2" (GetSystemRatio), "3" (GetDeduplicationRatio), "4" (GetReplicationThroughput), "5" (GetReplicationRatio), "6" (GetPerformanceStatisticsReadWriteIO), "7" (GetFCPortStatisticsSRR) or "8" (GetFCPortStatisticsAborts). If this property is NULL, the no methods are supported.
SupportedAsyncActions		Optional	SupportedAsyncActions lists the statistical services implemented asynchronously. The valid values are "0" (GetPhysicalCapacityUsage), "1" (GetLogicalCapacityUsage), "2" (GetSystemRatio), "3" (GetDeduplicationRatio), "4" (GetReplicationThroughput), "5" (GetReplicationRatio), "6" (GetPerformanceStatisticsReadWriteIO), "7" (GetFCPortStatisticsSRR) or "8" (GetFCPortStatisticsAborts). If this property is NULL, the no methods are supported.

EXPERIMENTAL

Annex A (informative) SMI-S Information Model

This standard is based on DMTF’s CIM schema, version 2.29. The DMTF schema is available in the machinereadable 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/cim_schema_v2290

Most SMI-S Profiles are primarily based on the DMTF Final MOFs. Content marked as “Experimental” or “Implemented” may be based on DMTF’s Experimental MOFs. Some SMI-S Experimental Profiles may also use classes with a SNIA_ prefix; MOFs from these classes are available from SNIA.