
**Information technology — Storage
management —**

**Part 7:
Host elements**





COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2021

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier; Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see <http://patents.iec.ch>).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by SNIA (as Storage Management Technical Specification, Part 7 Host Elements, Version 1.8.0, Revision 5) and drafted in accordance with its editorial rules. It was adopted, under the JTC 1 PAS procedure, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

This second edition cancels and replaces the first edition (ISO/IEC 24775-7:2014), which has been technically revised.

The main changes compared to the previous edition are as follows:

- USAGE text was revised to address code (now included in the front matter for all SNIA specifications)
- All recipes and their references were deleted.
- Instances of subprofile were changed to profile. In the annex, instances of subprofile were changed to component profile (TSG meeting voice vote).
- Profile versions and related text were updated. (TSG meeting voice vote).
- Indications have been replaced by DMTF Indications, and all affected clauses updated. (TSG meeting voice vote).
- Instances of Experimental within profiles already labeled as Experimental were removed to avoid confusion and redundancy. (Editorial change)
- CIM/XML was changed to CIM-XML (Response to ballot comments).

- Annex: SMI-S Information Model.
- The CIM schema version was changed to 2.51 for V1.8.0 Rev3.
- Disk Partition Profile
 - Added descriptions of the references in CIM_BasedOn (Partition to Extent), CIM_BasedOn (Partition to Partition), CIM_LogicalDiskBasedOnPartition (LogicalDisk to Partition) and CIM_SystemDevice (System to LogicalDisk).
 - Made CIM_LogicalDiskBasedOnPartition (LogicalDisk to Partition) and CIM_SystemDevice (System to LogicalDisk) Conditional (rather than Optional), since CIM_LogicalDisk is Optional. Storage HBA Profile (SMI TWG Reviews).
 - Fixed the version numbers on the Related Profiles to match what the profiles claim.
 - Added Descriptions to the references in CIM_ProductElementComponent, CIM_Realizes and CIM_SystemDevice.
- FC HBA Profile
 - Removed the profile content and added text re it being Deprecated. (CORE-SMIS-SCR-000).
- Host Discovered Resources Profile
 - Promoted the maturity level from DRAFT to EXPERIMENTAL for update to remove SNIA_ classes and use DMTF CIM_ classes.
 - Added the related profiles table to the spec.
 - Moved NameFormat from LogicalDevice to LogicalDisk to match the mof.
 - Moved NameNamespace from LogicalDevice to LogicalDisk to match the mof.
 - Added descriptions on the References for CIM_HostedAccessPoint and CIM_SystemDevice.
 - Marked the SB Multipath Management Related Profile as Deprecated (SMIS-180-Errata-SCR00002).
- Host Hardware RAID Controller Profile
 - Fixed the version numbers on the Related Profiles to match what the profiles claim.
 - Added two ElementSoftwareIdentity associations to cover the 3 SoftwareIdentity classes (one Mandatory and one Optional).
 - Added descriptions on the References for CIM_AssociatedAlarm, CIM_ProductPhysicalComponent, CIM_ProtocolControllerForUnit (Extent or MediaAccessDevice), CIM_Realizes (Associates PhysicalPackage to PortController) and CIM_SAPAvailableForElement.
 - Fixed the description of Dependent in CIM_ControlledBy.
 - Added Key properties to the CIM_SoftwareIdentity CIM Element tables.
 - Changed the version of the profile to 1.7.0.
- iSCSI Initiator
 - Removed all occurrences of “(Host Hardware RAID Controller)” (CORE-SMIS-SCR-000).
 - Removed MATURITY="Draft" from the Launch In Context supported profile.
 - Added Descriptions for References in 2 CIM_BindsTo associations, CIM_ControlledBy, 2 CIM_DeviceSAPImplementations and CIM_ElementSoftwareIdentity.
 - Added Descriptions for References in 2 CIM_EndpointOfNetworkPipes, CIM_NetworkPipeComposition, CIM_ProductPhysicalComponent and CIM_Realizes.

- Added Descriptions for references in CIM_SAPAvailableForElement.
- Added Keys to CIM_PhysicalPackage.
- Changed the Central Class from PortController to CIM_ComputerSystem (TSG-SMIS-SCR00333).
- Memory Configuration Profile
 - Updated due to class and property name changes in DMTF Multi-type Memory profile (SMIS-170-Draft-SCR00006).
 - Introduced staged request concept.
 - Added use cases.
 - Addressed comments from internal and partner reviews.
 - Changed diagrams approved in Core TWG 3/23/15.
 - Changed SystemCreateClassName in class CIM_MemoryConfigurationService to SystemCreationClassName.
 - Changed RemainingCapacity in class CIM_ResourcePool to CurrentlyConsumedResource.
 - Changed ChangeableType in class CIM_MemoryAllocationSettingData to ResourceType.
 - Fixed the description of Replication in class CIM_MemoryAllocationSettingData.
 - Added a description for ChannelCount in class CIM_MemoryAllocationSettingData.
- Persistent Memory Configuration Profile
 - Added new material (most of profile content) to manage NVDIMMs (SMIS-170-Draft-SCR00006).
 - Changed diagrams approved in Core TWG 3/23/15- Changed RemainingCapacity in class CIM_ResourcePool to Reserved.
 - Changed SecurityFeatures in class CIM_PersistentMemoryCapabilities to SecurityFeatures.
 - Removed ChannelCount in class CIM_PersistentMemoryNamespaceSettingData, since it is not in the mof and not referenced anywhere in the profile text.
 - Added three methods to CIM_PersistentMemoryService to support the profile text.
 - Fixed the ManagedElement reference in CIM_ElementConformsToProfile.
 - Fixed the references in CIM_SystemDevice.
 - Fixed the references in CIM_BasedOn.
- SB Multipath Management Profile
 - Promoted the maturity level from DRAFT to EXPERIMENTAL for update to remove SNIA_ classes and use DMTF CIM_ classes (TSG-SMIS-SCR00315.001).
 - Added reference to SMI-S Version 1.6.1 Revision 5. No content in this deprecated profile SB Multipath.
- SCSI Multipath Management Profile
 - Added keys to CIM_SCSIMultipathConfigurationCapabilities and CIM_SCSIMultipathSettings (SMIS-170-Draft-SCR00006).
- Annex A (informative) SMI-S Information Model
 - DMTF's CIM schema version changed to 2.45.0.
- References

- Added DMTF DSP1054 v1.2.2, Indications Profile (and changed version to 1.2.2 throughout book).
- Updated DSP 1071, Multi-type System Memory Profile reference here and in profiles.
- Added reference to DSP 1119, 1.0.0b Diagnostics Job Control Profile.
- Removed reference to DSP 1119\.
- Removed 2.3.

A list of all parts in the ISO/IEC 24775 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

INTENDED AUDIENCE

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

CHANGES TO THE SPECIFICATION

Each publication of this specification is uniquely identified by a three-level identifier, comprised of a version number, a release number and an update number. The current identifier for this specification is version 1.8.0. 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 document unless its initial architecture has been completed and reviewed. Some content included in this document has complete and reviewed design, but lacks implementation experience and the maturity gained through implementation experience. This content is included in order to gain wider review and to gain implementation experience. This material is referred to as “Experimental”. It is presented here as an aid to implementers who are interested in likely future developments within the SMI specification. The contents of an Experimental profile may change as implementation experience is gained. There is a high likelihood that the changed content will be included in an upcoming revision of the specification. Experimental material can advance to a higher maturity level as soon as implementations are available. Figure 1 is a sample of the typographical convention for Experimental content.

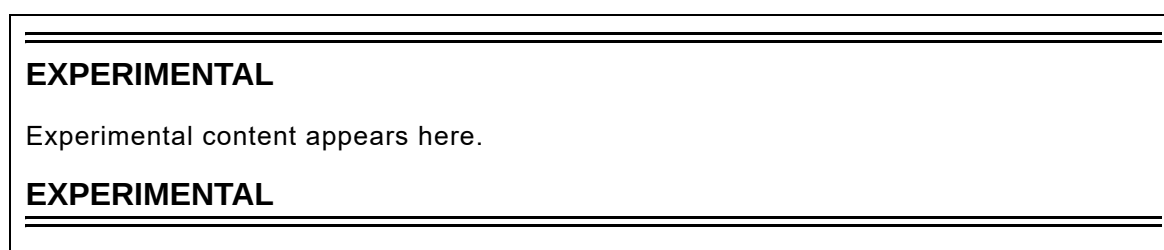


Figure 1 - Experimental Maturity Level Tag

Implemented Maturity Level

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

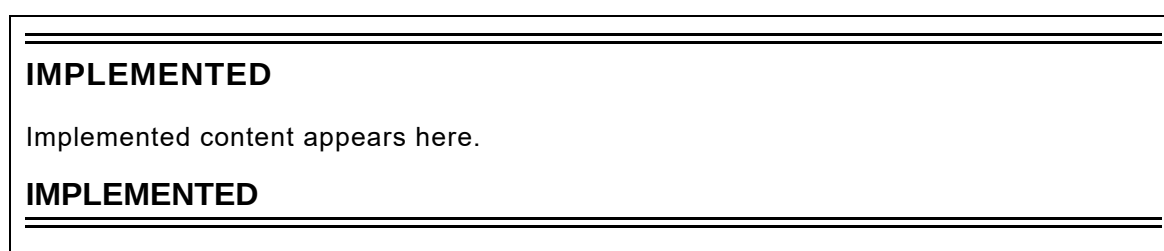


Figure 2 - Implemented Maturity Level Tag

Stable Maturity Level

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

maturity level shall not rely on any content that is Experimental. Figure 3 is a sample of the typographical convention for Implemented content.

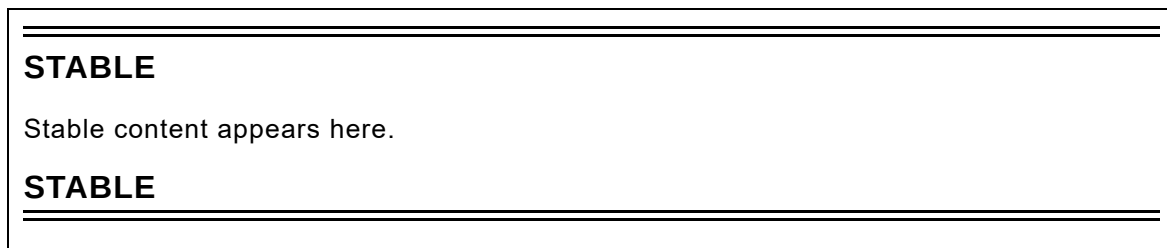


Figure 3 - Stable Maturity Level Tag

Finalized Maturity Level

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

Deprecated Material

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

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



Figure 4 - Deprecated Tag

Contents

List of Figures	15
List of Tables	17
Foreword	23
1 Scope	25
2 Normative References	27
2.1 Overview	27
2.2 Approved references	27
2.3 References under development	27
3 Terms, Definitions, Symbols, Abbreviations, and Conventions	29
3.1 General	29
3.2 Terms and Definitions	29
4 Disk Partition Profile	31
4.1 Description	31
4.2 Health and Fault Management Considerations	37
4.3 Supported Profiles and Packages	38
4.4 Methods of the Profile	38
4.5 Client Considerations and Recipes	39
4.6 CIM Elements	39
5 FC HBA Profile	49
6 Storage HBA Profile	51
6.1 Synopsis	51
6.2 Description	51
6.3 Implementation	51
6.4 Methods of the Profile	57
6.5 Use Cases	58
6.6 CIM Elements	58
7 Host Discovered Resources Profile	63
7.1 Description	63
7.2 Health and Fault Management Considerations	68
7.3 Cascading Considerations	68
7.4 Extrinsic Methods of the Profile	68
7.5 Use Cases	69
7.6 CIM Elements	69
8 Host Hardware RAID Controller Profile	79
8.1 Synopsis	79
8.2 Description	80
8.3 Implementation	80
8.4 Methods	94
8.5 Use Cases	94
8.6 CIM Elements	94
9 iSCSI Initiator Profile	109
9.1 Description	109
9.2 Health and Fault Management Considerations	112
9.3 Methods of the Profile	112
9.4 Use Cases	112
9.5 CIM Elements	113

10	SCSI Multipath Management Profile	125
10.1	Description	125
10.2	Health and Fault Management Considerations	128
10.3	Methods of the Profile	128
10.4	Use Cases	129
10.5	CIM Elements	129
11	SB Multipath Management Profile	137
12	Memory Configuration Profile	139
12.1	Synopsis	139
12.2	Description	139
12.3	Implementation	141
12.4	Methods	143
12.5	Use Cases	148
12.6	CIM Elements	153
13	Persistent Memory Configuration Profile	161
13.1	Synopsis	161
13.2	Overview	161
13.3	Implementation	163
13.4	Methods	164
13.5	Use Cases	171
13.6	CIM Elements	175
	Annex A (informative) SMI-S Information Model	183
	Annex B (Informative) Host Profile Deployment Guidelines	185

LIST OF FIGURES

Figure 1 - Experimental Maturity Level Tag	10
Figure 2 - Implemented Maturity Level Tag	10
Figure 3 - Stable Maturity Level Tag	11
Figure 4 - Deprecated Tag	11
Figure 5 - Disk Partition Class Hierarchy	32
Figure 6 - Disk Partition Class Diagram	33
Figure 7 - Disk MBR Partition Example	34
Figure 8 - MBR Partition Instance Diagram	35
Figure 9 - MBR and VTOC Partition Instance Diagram	36
Figure 10 - Partition Instance Diagram for Size/Address Rules	37
Figure 11 - Model Overview	52
Figure 12 - Profile Registration Profile	53
Figure 13 - Software Inventory Profile in Storage HBA	55
Figure 14 - HBA Card with Physical Classes	56
Figure 15 - Host Discovered Resources Block Diagram	64
Figure 16 - Host Discovered Resources Class Diagram	66
Figure 17 - Single SPI Disk Model	66
Figure 18 - Three FCP Logical Unit Instance Diagram	67
Figure 19 - ATA Discovered Resource Model	67
Figure 20 - SB Host Discovered Resources	68
Figure 21 - Host Hardware RAID Controller Package Diagram	80
Figure 22 - Host Hardware RAID resources scoped to HHRC ComputerSystem	81
Figure 23 - Alarms in Host Hardware RAID Controller	82
Figure 24 - Profile Registration with Host Hardware RAID Controller and Base Server Profiles	83
Figure 25 - Implementation of Physical Asset Profile	84
Figure 26 - Block Services Package in Host Hardware RAID Controller	85
Figure 27 - DAPort Profile in Host Hardware Controller	87
Figure 28 - Software Inventory Profile in Host Hardware RAID Controller	87
Figure 29 - Initiator Port profiles and Disk Drive Lite Profile	88
Figure 30 - Model for Imported Disks	89
Figure 31 - Imported Virtual Volumes	90
Figure 32 - Device "Pass Through" Example	91
Figure 33 - Example of Mutli-Function Controllers	93
Figure 34 - iSCSI Product and Package Model	110
Figure 35 - iSCSI Sessions and Connections Model	111
Figure 36 - iSCSI Initiator Node	112
Figure 37 - Multipath Management Class Diagram	126
Figure 38 - Four Path Instance Diagram	127
Figure 39 - Memory Configuration Class Diagram	140
Figure 40 - Use Case - Profile Registration	148
Figure 41 - Use Case - Memory Configuration Capabilities	149
Figure 42 - Use Case - Memory Capability Discovery	149
Figure 43 - Use Case - Allocation	150
Figure 44 - Use Case - Allocation Complete	151

Figure 45 - Use Case - Before Deallocation 152

Figure 46 - Use Case - After Deallocation 152

Figure 47 - Persistent Memory Configuration: Class Diagram..... 162

Figure 48 - Use Case: Profile Registration 172

Figure 49 - Use Case - Detect Capabilities..... 173

Figure 50 - Use Case - Create Namespace..... 174

Figure B.1 Profile Registration and FC HBA Profiles 186

Figure B.2 Profile Registration and Storage HBA Profiles 187

Figure B.3 Profile Registration and Host Hardware RAID Controller Profiles 188

Figure B.4 Deploying FC HBA with Storage HBA (SAS) Profiles 189

LIST OF TABLES

Table 1 - Capabilities Properties.....	36
Table 2 - CIM Elements for Disk Partition.....	39
Table 3 - SMI Referenced Properties/Methods for CIM_BasedOn (Partition to Extent).....	40
Table 4 - SMI Referenced Properties/Methods for CIM_BasedOn (Partition to Partition).....	41
Table 5 - SMI Referenced Properties/Methods for CIM_DiskPartition	41
Table 6 - SMI Referenced Properties/Methods for CIM_DiskPartitionConfigurationCapabilities	42
Table 7 - SMI Referenced Properties/Methods for CIM_DiskPartitionConfigurationService	42
Table 8 - SMI Referenced Properties/Methods for CIM_ElementCapabilities	43
Table 9 - SMI Referenced Properties/Methods for CIM_ElementConformsToProfile (DiskPartitionConfigura- tionService to Disk Partition RegisteredProfile)43	
Table 10 - SMI Referenced Properties/Methods for CIM_GPTDiskPartition	43
Table 11 - SMI Referenced Properties/Methods for CIM_GenericDiskPartition	44
Table 12 - SMI Referenced Properties/Methods for CIM_HostedService	44
Table 13 - SMI Referenced Properties/Methods for CIM_InstalledPartitionTable (Capabilities to Extent).....	45
Table 14 - SMI Referenced Properties/Methods for CIM_InstalledPartitionTable (Capabilities to Partition).....	45
Table 15 - SMI Referenced Properties/Methods for CIM_LogicalDisk	45
Table 16 - SMI Referenced Properties/Methods for CIM_LogicalDiskBasedOnPartition (LogicalDisk to Parti- tion)46	
Table 17 - SMI Referenced Properties/Methods for CIM_StorageExtent.....	46
Table 18 - SMI Referenced Properties/Methods for CIM_SystemDevice (System to Extent)	47
Table 19 - SMI Referenced Properties/Methods for CIM_SystemDevice (System to LogicalDisk).....	47
Table 20 - SMI Referenced Properties/Methods for CIM_SystemDevice (System to Partition)	47
Table 21 - SMI Referenced Properties/Methods for CIM_VTOCDiskPartition	48
Table 22 - Related Profiles for Storage HBA	51
Table 23 - CIM_PortController	57
Table 24 - CIM_SystemDevice	58
Table 25 - CIM Elements for Storage HBA.....	58
Table 26 - SMI Referenced Properties/Methods for CIM_ControlledBy	59
Table 27 - SMI Referenced Properties/Methods for CIM_PortController	59
Table 28 - SMI Referenced Properties/Methods for CIM_Product	60
Table 29 - SMI Referenced Properties/Methods for CIM_ProductElementComponent	60
Table 30 - SMI Referenced Properties/Methods for CIM_Realizes	60
Table 31 - SMI Referenced Properties/Methods for CIM_SystemDevice.....	61
Table 32 - Related Profiles for HostDiscoveredResources	63
Table 33 - CIM Elements for Host Discovered Resources	69
Table 34 - SMI Referenced Properties/Methods for CIM_ATAInitiatorTargetLogicalUnitPath	70
Table 35 - SMI Referenced Properties/Methods for CIM_ATAProtocolEndpoint	71
Table 36 - SMI Referenced Properties/Methods for CIM_ElementConformsToProfile (LogicalDevice to Host Discovered Resources RegisteredProfile)71	
Table 37 - SMI Referenced Properties/Methods for CIM_HostedAccessPoint	72
Table 38 - SMI Referenced Properties/Methods for CIM_LogicalDevice (LogicalDevice)	72
Table 39 - SMI Referenced Properties/Methods for CIM_LogicalDisk (LogicalDevice)	72
Table 40 - SMI Referenced Properties/Methods for CIM_SCSIArbitraryLogicalUnit (LogicalDevice).....	73
Table 41 - SMI Referenced Properties/Methods for CIM_SCSIInitiatorTargetLogicalUnitPath	74
Table 42 - SMI Referenced Properties/Methods for CIM_SCSIProtocolEndpoint.....	74
Table 43 - SMI Referenced Properties/Methods for CIM_StorageExtent (LogicalDevice).....	75
Table 44 - SMI Referenced Properties/Methods for CIM_SystemDevice.....	75

Table 45 - SMI Referenced Properties/Methods for CIM_TapeDrive (LogicalDevice)	75
Table 46 - SMI Referenced Properties/Methods for CIM_SBInitiatorTargetLogicalUnitPath	76
Table 47 - SMI Referenced Properties/Methods for CIM_SBProtocolEndpoint	77
Table 48 - Related Profiles for Host Hardware RAID Controller	79
Table 49 - CIM Elements for Host Hardware RAID Controller	94
Table 50 - SMI Referenced Properties/Methods for CIM_AlarmDevice	96
Table 51 - SMI Referenced Properties/Methods for CIM_AssociatedAlarm	96
Table 52 - SMI Referenced Properties/Methods for CIM_ComputerSystem (Host Hardware RAID Controller)97	
Table 53 - SMI Referenced Properties/Methods for CIM_ComputerSystemPackage (Associates PhysicalPackage to ComputerSystem)97	
Table 54 - SMI Referenced Properties/Methods for CIM_ControlledBy	98
Table 55 - SMI Referenced Properties/Methods for CIM_ElementCapabilities (ImplementationCapabilities to System)98	
Table 56 - SMI Referenced Properties/Methods for CIM_ImplementationCapabilities (ImplementationCapabilities)99	
Table 57 - SMI Referenced Properties/Methods for CIM_LogicalIdentity	99
Table 58 - SMI Referenced Properties/Methods for CIM_MediaAccessDevice	100
Table 59 - SMI Referenced Properties/Methods for CIM_PortController	100
Table 60 - SMI Referenced Properties/Methods for CIM_Product	101
Table 61 - SMI Referenced Properties/Methods for CIM_ProductPhysicalComponent	101
Table 62 - SMI Referenced Properties/Methods for CIM_ProtocolControllerForUnit (Extent or MediaAccessDevice)101	
Table 63 - SMI Referenced Properties/Methods for CIM_ProtocolControllerForUnit (Volume)	102
Table 64 - SMI Referenced Properties/Methods for CIM_Realizes (Associates PhysicalPackage to PortController)102	
Table 65 - SMI Referenced Properties/Methods for CIM_SAPAvailableForElement	103
Table 66 - SMI Referenced Properties/Methods for CIM_SCSIProtocolController	103
Table 67 - SMI Referenced Properties/Methods for CIM_SoftwareIdentity (Driver)	103
Table 68 - SMI Referenced Properties/Methods for CIM_SoftwareIdentity (FCode/BIOS)	104
Table 69 - SMI Referenced Properties/Methods for CIM_SoftwareIdentity (Firmware)	104
Table 70 - SMI Referenced Properties/Methods for CIM_StorageExtent	104
Table 71 - SMI Referenced Properties/Methods for CIM_SystemComponent	105
Table 72 - SMI Referenced Properties/Methods for CIM_SystemDevice (Associates System to AlarmDevice)105	
Table 73 - SMI Referenced Properties/Methods for CIM_SystemDevice (Associates controller system to PortController)106	
Table 74 - SMI Referenced Properties/Methods for CIM_SystemDevice (System to SCSIProtocolController) .106	
Table 75 - SMI Referenced Properties/Methods for CIM_ElementSoftwareIdentity (Driver)	107
Table 76 - SMI Referenced Properties/Methods for CIM_ElementSoftwareIdentity (Firmware or FCode/BIOS)107	
Table 77 - Supported Profiles for iSCSI Initiator	109
Table 78 - iSCSI Terminology	109
Table 79 - OperationalStatus Values	112
Table 80 - CIM Elements for iSCSI Initiator	113
Table 81 - SMI Referenced Properties/Methods for CIM_BindsTo (TCPProtocolEndpoint to IPProtocolEndpoint)114	
Table 82 - SMI Referenced Properties/Methods for CIM_BindsTo (iSCSIProtocolEndpoint to TCPProtocolEndpoint)114	
Table 83 - SMI Referenced Properties/Methods for CIM_ComputerSystem	114
Table 84 - SMI Referenced Properties/Methods for CIM_ControlledBy	115

Table 85 - SMI Referenced Properties/Methods for CIM_DeviceSAPImplementation (EthernetPort to IPProtocolEndpoint)115	
Table 86 - SMI Referenced Properties/Methods for CIM_DeviceSAPImplementation (EthernetPort to iSCSIProtocolEndpoint)116	
Table 87 - SMI Referenced Properties/Methods for CIM_ElementSoftwareIdentity.....	116
Table 88 - SMI Referenced Properties/Methods for CIM_EndpointOfNetworkPipe (Between iSCSI Session and iSCSIProtocolEndpoint)116	
Table 89 - SMI Referenced Properties/Methods for CIM_EndpointOfNetworkPipe (Between iSCSI connection and TCPProtocolEndpoint)117	
Table 90 - SMI Referenced Properties/Methods for CIM_InstalledSoftwareIdentity	117
Table 91 - SMI Referenced Properties/Methods for CIM_NetworkPipeComposition	117
Table 92 - SMI Referenced Properties/Methods for CIM_PhysicalPackage	118
Table 93 - SMI Referenced Properties/Methods for CIM_PortController	118
Table 94 - SMI Referenced Properties/Methods for CIM_Product	118
Table 95 - SMI Referenced Properties/Methods for CIM_ProductPhysicalComponent	119
Table 96 - SMI Referenced Properties/Methods for CIM_Realizes	119
Table 97 - SMI Referenced Properties/Methods for CIM_SAPAvailableForElement	119
Table 98 - SMI Referenced Properties/Methods for CIM_SCSIProtocolController	120
Table 99 - SMI Referenced Properties/Methods for CIM_SoftwareIdentity	120
Table 100 - SMI Referenced Properties/Methods for CIM_SystemDevice (to EthernetPort)	120
Table 101 - SMI Referenced Properties/Methods for CIM_SystemDevice (to PortController)	121
Table 102 - SMI Referenced Properties/Methods for CIM_SystemDevice (to ProtocolController).....	121
Table 103 - SMI Referenced Properties/Methods for CIM_iSCSIConnection	121
Table 104 - SMI Referenced Properties/Methods for CIM_iSCSISession.....	123
Table 105 - CIM Elements for SCSI Multipath Management.....	129
Table 106 - SMI Referenced Properties/Methods for CIM_ConcreteComponent.....	130
Table 107 - SMI Referenced Properties/Methods for CIM_ConcreteDependency	130
Table 108 - SMI Referenced Properties/Methods for CIM_ElementCapabilities	131
Table 109 - SMI Referenced Properties/Methods for CIM_ElementConformsToProfile (SCSIPathConfigurationService to SCSI Multipath Management RegisteredProfile)131	
Table 110 - SMI Referenced Properties/Methods for CIM_ElementSettingData.....	131
Table 111 - SMI Referenced Properties/Methods for CIM_ElementSoftwareIdentity (Driver).....	132
Table 112 - SMI Referenced Properties/Methods for CIM_ElementSoftwareIdentity (MP API Plugin)	132
Table 113 - SMI Referenced Properties/Methods for CIM_HostedService	132
Table 114 - SMI Referenced Properties/Methods for CIM_MemberOfCollection	133
Table 115 - SMI Referenced Properties/Methods for CIM_Product	133
Table 116 - SMI Referenced Properties/Methods for CIM_SCSIMultipathConfigurationCapabilities.....	133
Table 117 - SMI Referenced Properties/Methods for CIM_SCSIMultipathSettings.....	134
Table 118 - SMI Referenced Properties/Methods for CIM_SCSIPathConfigurationService.....	134
Table 119 - SMI Referenced Properties/Methods for CIM_SCSITargetPortGroup	135
Table 120 - SMI Referenced Properties/Methods for CIM_ServiceAvailableToElement.....	135
Table 121 - SMI Referenced Properties/Methods for CIM_SoftwareIdentity (Driver).....	136
Table 122 - SMI Referenced Properties/Methods for CIM_SoftwareIdentity (MP API Plugin)	136
Table 123 - Related Profiles	139
Table 124 - Operations: CIM_MemoryConfigurationService	143
Table 125 - CIM_MemoryConfigurationService.AllocateFromPool() Method: Return Code Values.....	143
Table 126 - CIM_MemoryConfigurationService.AllocateFromPool() Method: Parameters	144
Table 127 - CIM_MemoryConfigurationService.ReturnToPool() Method: Return Code Values.....	144
Table 128 - CIM_MemoryConfigurationService.ReturnToPool() Method: Parameters.....	144
Table 129 - Operations: CIM_ResourcePool.....	145

Table 130 - Operations: CIM_MemoryCapabilities	145
Table 131 - Operations: CIM_MemoryConfigurationCapabilities	145
Table 132 - Operations: CIM_MemoryAllocationSettingData	146
Table 133 - Operations: CIM_ElementCapabilities	146
Table 134 - Operations: CIM_ServiceAffectsElement	146
Table 135 - Operations: CIM_ConcreteComponent	147
Table 136 - Operations: CIM_ElementAllocatedFromPool	147
Table 137 - Operations: CIM_HostedService	147
Table 138 - Operations: CIM_ElementSettingData	147
Table 139 - Operations: CIM_ElementConformsToProfile	148
Table 140 - CIM Elements	153
Table 141 - Class: CIM_RegisteredProfile	153
Table 142 - Class: CIM_MemoryConfigurationService	154
Table 143 - Class: CIM_ResourcePool	154
Table 144 - Class: CIM_MemoryCapabilities	155
Table 145 - Class: CIM_MemoryConfigurationCapabilities	155
Table 146 - Class: CIM_MemoryAllocationSettingData	156
Table 147 - Class: CIM_ElementSettingData - Use 1	157
Table 148 - Class: CIM_ElementSettingData - Use 2	157
Table 149 - Class: CIM_HostedService	158
Table 150 - Class: CIM_ElementAllocatedFromPool	158
Table 151 - Class: CIM_ConcreteComponent	159
Table 152 - Class: CIM_ElementCapabilities use 1	159
Table 153 - Class CIM_ElementCapabilities use 2	160
Table 154 - Related Profiles	161
Table 155 - Operations: CIM_ResourcePool	164
Table 156 - Operations: CIM_PersistentMemoryCapabilities	165
Table 157 - Operations: CIM_PersistentMemoryService	165
Table 158 - AllocateFromPool() Method: Return Code Values	166
Table 159 - AllocateFromPool() Method: Parameters	166
Table 160 - ReturnToPool() Method: Return Code Values	166
Table 161 - ReturnToPool() Method: Parameters	167
Table 162 - ModifyNamespace() Method: Return Code Values	167
Table 163 - ModifyNamespace() Method: Parameters	168
Table 164 - Operations: CIM_PersistentConfigurationCapabilities	168
Table 165 - Operations: CIM_PersistentMemoryNamespace	168
Table 166 - Operations: CIM_PersistentMemoryNamespaceSettingData	169
Table 167 - Operations: CIM_ElementCapabilities	169
Table 168 - Operations: CIM_ServiceAffectsElement	170
Table 169 - Operations: CIM_ConcreteComponent	170
Table 170 - Operations: CIM_ElementAllocatedFromPool	170
Table 171 - Operations: CIM_HostedService	170
Table 172 - Operations: CIM_ElementSettingData	171
Table 173 - Operations: CIM_ElementConformsToProfile	171
Table 174 - Operations: CIM_SystemDevice	171
Table 175 - CIM Elements for Persistent Memory Configuration	175
Table 176 - Class: CIM_RegisteredProfile	175
Table 177 - Class: CIM_ResourcePool	176
Table 178 - Class: CIM_PersistentMemoryCapabilities	176

Table 179 - Class: CIM_PersistentMemoryService	177
Table 180 - Class: CIM_PersistentConfigurationCapabilities	177
Table 181 - Class: CIM_PersistentMemoryNamespace	178
Table 182 - Class: CIM_PersistentMemoryNamespaceSettingData	178
Table 183 - CIM_ElementCapabilities	179
Table 184 - CIM_ElementCapabilities	179
Table 185 - Class: CIM_ServiceAffectsElement	180
Table 186 - Class: CIM_ConcreteComponent	180
Table 187 - Class: CIM_ElementAllocatedFromPool	180
Table 188 - Class: CIM_HostedService	180
Table 189 - Class CIM_ElementSettingData	181
Table 190 - Class: CIM_ElementConformsToProfile	181
Table 191 - Class: CIM_SystemDevice	182
Table 192 - Class: CIM_BasedOn	182

FOREWORD

The host-based storage portion of the Storage Management Technical Specification contains profiles and other clauses for management of host-based storage devices. Host-based storage devices provide storage capabilities to a host computer system. Examples of these devices include Fiber Channel Host Bus Adapters, Serial Attached SCSI Host Bus Adapters, RAID Controllers, JBODs (Just-A-Bunch-Of-Disks) and Operating System-discovered storage resources. The host-based profiles describe the manageability required for each device and the connectivity to the host computer system. The host-based profiles leverage existing profiles within this specification, as well as other profiles from the Distributed Management Task Force, where applicable, to create a comprehensive management model.

Parts of this Standard

This standard is subdivided in the following parts:

- *Storage Management Technical Specification, Part 1 Overview, 1.8.0 Rev 4*
- *Storage Management Technical Specification, Part 2 Common Architecture, 1.8.0 Rev 4*
- *Storage Management Technical Specification, Part 3 Common Profiles, 1.8.0 Rev 4*
- *Storage Management Technical Specification, Part 4 Block Devices, 1.8.0 Rev 4*
- *Storage Management Technical Specification, Part 5 Filesystems, 1.8.0 Rev 4*
- *Storage Management Technical Specification, Part 6 Fabric, 1.8.0 Rev 4*
- *Storage Management Technical Specification, Part 7 Host Elements, 1.8.0 Rev 4*
- *Storage Management Technical Specification, Part 8 Media Libraries, 1.8.0 Rev 4*

SNIA Web Site

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

SNIA Address

Requests for interpretation, suggestions for improvement and addenda, or defect reports are welcome. They should be sent via the SNIA Feedback Portal at <http://www.snia.org/feedback/> or by mail to the Storage Networking Industry Association, 4360 ArrowsWest Drive, Colorado Springs, Colorado 80907, U.S.A.

1 Scope

The host-base storage portion of the Storage Management Technical Specification defines management profiles for autonomous, component and abstract profiles for management of host-based storage devices. The autonomous profiles describe the management of a stand-alone host-based storage entity. The component profiles describe management of aspects of host-based storage entities that may be used by other autonomous profiles. Finally, this section describes abstract profiles that may be used as a basis for creating additional Host-based autonomous profiles.

This version of the Host-based Storage portion of the Storage Management Technical Specification includes autonomous profiles:

- "The Host Discovered Resources Profile

This profile defines the model for the storage devices presented to an operating system running on a host computer system. In addition, this profile describes the map of storage associated to a host-computer system that a client application can discover.

- "The Fibre Channel HBA Profile

This profile defines the model and functions of a Fibre Channel HBA that exports block storage to a host computer system from a SAN device (Fibre Channel switch, array, tape library, etc.).

- iSCSI Initiator Profile

This profile defines the model and functions necessary to manage an iSCSI initiator.

Component profiles used by autonomous profiles to describe aspects of host-based storage elements and services. The component profiles defined in this version of the specification include:

- Host Hardware RAID Controller Profile

This profile defines the model and functions of a host-based RAID controller that exports block storage to a host computer system from locally attached storage devices (internal hard drives, JBODs, etc.)

- Storage HBA Profile

This profile defines the model and functions of a SAS, SATA, SPI, or Fibre Channel HBA that exports storage to a host computer system from a SAN device (Fibre Channel switch, array, tape library, etc.).

- Disk Partition Profile

The Disk Partition profile models partition (or slice) configuration services provided by operating systems on some platforms.

- SB Multipath Management Profile

The SB Multipath Management Profile models paths (connections between host controllers and device ports) for environments supporting the SB (Single Byte) command protocol.

- SCSI Multipath Management Profile

The SCSI Multipath Management profile models paths (connections between host controllers, device ports, and logical units) for environments supporting the SCSI command protocol.

2 Normative References

2.1 Overview

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

Systems Management: Data Storage Management (XDSM) API - ISBN: 1-85912-190-X

DMTF DSP1054 Indications Profile 1.2.2

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

2.3 References under development

EXPERIMENTAL

DMTF documents that are works in progress.

DMTF DSP1002, Diagnostics Profile 2.1.0

http://dmtof.org/sites/default/files/standards/documents/DSP1002_2.1.0a.pdf

DMTF DSP 1071, Multi-type System Memory Profile 1.0.0

http://www.dmtf.org/sites/default/files/standards/documents/DSP1071_1.0.0.pdf

DMTF DSP1104, FC HBA Diagnostics Profile 1.1.0a

http://dmtof.org/sites/default/files/standards/documents/DSP1104_1.1.0a.pdf

EXPERIMENTAL

Storage Management Technical Specification, Part 2 Common Architecture, 1.8.0 Rev 4

Storage Management Technical Specification, Part 3 Common Profiles, 1.8.0 Rev 4

3 Terms, Definitions, Symbols, Abbreviations, and Conventions

3.1 General

For the purposes of this document, the terms, definitions, symbols, abbreviations, and conventions given in *Storage Management Technical Specification, Part 2 Common Architecture, 1.8.0 Rev 4* and the following apply.

3.2 Terms and Definitions

3.2.1

Fibre Channel

a serial I/O bus capable of supporting multiple protocols, including access to open system storage (FCP protocol), access to mainframe storage (**FICON**^{™1} protocol), and IP

3.2.2

Host Bus Adapter (HBA)

an I/O adapter that connects a host I/O bus to a computer's memory system

3.2.3

host computer system

any computer system to which disks, disk subsystems, or file servers are attached and accessible for data storage and I/O.

3.2.4

JBOD

an acronym for "Just a Bunch Of Disks." A cabinet or enclosure of disks

3.2.5

logical disk

block storage on which file systems are built

A logical disk would be formatted for a particular file system.

3.2.6

Operating System (OS)

software that manages the resources of a host computer system.

3.2.7

RAID

an Acronym for Redundant Array of Independent Disks, a family of techniques for managing multiple disks to deliver desirable cost, data availability, and performance characteristics to host environments

3.2.8

storage device enclosure

a cabinet or enclosure of storage media devices.

Note 1 to entry: Example: JBOD

3.2.9

storage volume

unit of capacity served from a block storage device

1.FICON[™] is an example of a suitable product available commercially. This information is given for the convenience of users of this standard and does not constitute an endorsement of this product by SNIA or any standards organization.

EXPERIMENTAL

4 Disk Partition Profile

4.1 Description

4.1.1 Synopsis

Profile Name: Disk Partition (Component Profile)

Version: 1.6.0

Organization: SNIA

Central Class: DiskPartitionConfigurationService

Scoping Class: Base Server ComputerSystem

Related Profiles: Not defined in this document.

4.1.2 Overview

This profile models partition (or slice) configuration services provided by operating systems on some platforms. Some operating systems do not use this type of partitioning. On the operating systems that do, the operating system disk drivers treat partitions as virtual disks. The types of valid partitions are determined by the operating system and the partitioning tools.

We need to consider several operating system variants related to operating system partitions

- On some platforms (e.g., Solaris, Windows), a raw disk volume needs to be partitioned before an application (i.e., a filesystem) uses it. There may be just a single partition on the volume. In these platforms, there is not a name that represents an entire disk volume if that disk volume has multiple partitions.
- On other platforms (e.g., Linux), an application resides on a partition or on the entire disk volume.
- Different operating systems have incompatible partitioning approaches and on-disk data structures (e.g. disk labels or partition tables). This specification refers to these approaches as styles. Each style may be supported by multiple operating systems, and most operating systems support multiple styles. The styles supported in this profile are MBR (used on all operating systems running on X86 hardware), vtoc (Solaris and other operating systems with a BSD heritage), and GPT (an emerging style that supports multi-terabyte disk volumes).
- Some styles support multiple tiers of partitions - a partition at one tier may have sub-partitions. On Windows, extended partitions are also a second tier with MBR partitions at each tier.
- Some operating systems utilize two tiers of partitions with different styles at different tiers. For example, BSD-derived Unix variants running on X86 platforms: the lower tier is the X86 BIOS-supported MBR partitions; BSD-style slices can be installed on one of the MBR partitions.
- Some operating systems (AIX, HP_UX) have no equivalent to partitioning.
- Some partition styles have a fixed number of partitions (dependent on the partition type); the user can't create or delete partitions, just adjust the properties of one of the pre-defined partitions.

A partitioned disk volume has an associated partition table. The partition table contains information about the partitions on the disk volume – the starting address, length, and (in some cases) the type of the partition. In certain cases, a partition table can be associated with a partition; allowing multiple tiers of partitions.

In order for storage applications (e.g., logical volume managers, filesystems, databases) to use a disk volume, the operating system provides a name for the volume. These names appear to be filenames but are part of one (or a few) special namespaces managed by the operating system. Windows drive letters and Unix `/dev/` directories are examples of the special namespaces. Any extent that is consumable by storage applications is modeled as LogicalDisk; the LogicalDisk.Name property provides this special filename. The exported extent resulting from a partition is a LogicalDisk; on systems that do not require partitions, each usable disk volume has a LogicalDisk instance that models the operating system name. Extents that are not available for storage applications are modeled as StorageExtent (or StorageExtent subclasses other than LogicalDisk) instances and have a name derived from the underlying hardware and partition number.

Operating systems may have different partition styles. The most common style is the MBR (Master Boot Record) style used on x86 PCs. This style supports four primary partitions on a disk volume with an optional second-tier (extended/logical partitions). Solaris uses a style called VTOC that is derived from and similar to BSD partitions. VTOC supports eight partitions. On Solaris X86, VTOC is installed in one X86 MBR primary partition for compatibility with other x86 operating systems. GPT is a new set of interfaces for x86 64-bit environments and includes a partitioning style. Of particular note is that GPT partitions can exceed the two-terabyte limit associated with other partition styles. So many vendors are migrating towards GPT as an option for supporting larger volumes. This profile includes separate specialized subclasses for MBR, VTOC, and GPT partitions. Their relationship is summarized in Figure 7: "Disk Partition Class Hierarchy".

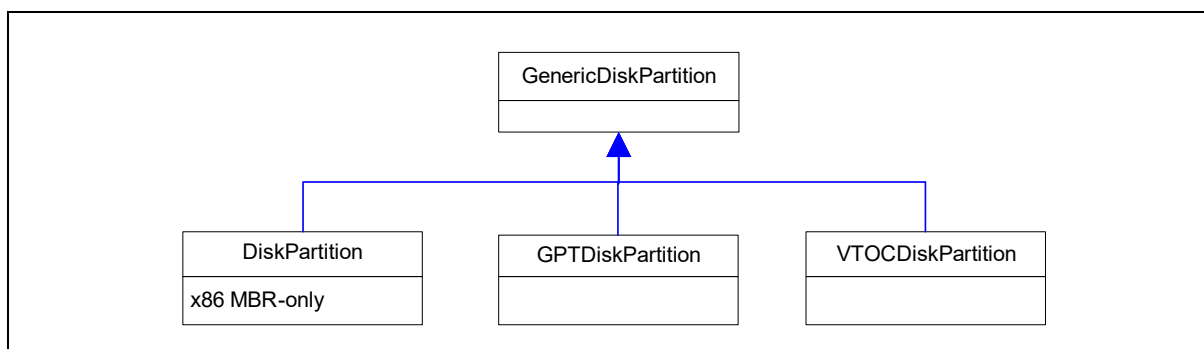


Figure 7 - Disk Partition Class Hierarchy

This profile includes a partition configuration service class that allows a client to create partition tables and modify partitions. It also includes a partition configuration capabilities class that describes the partition configuration capabilities of the system. Separate capabilities instances describe each partition style supported on the system. There shall be one instance of DiskPartitionConfigurationService, as shown in Figure 8: "Disk Partition Class Diagram".

All references to ComputerSystem in the Disk Partition Profile implies a single instance for a customer server or storage system as defined in the Base Server Profile. See Annex A: (Informative) Host Profile Deployment Guidelines, 1.6.0 Rev 3 for information on the use of host profiles with Base Server profile.

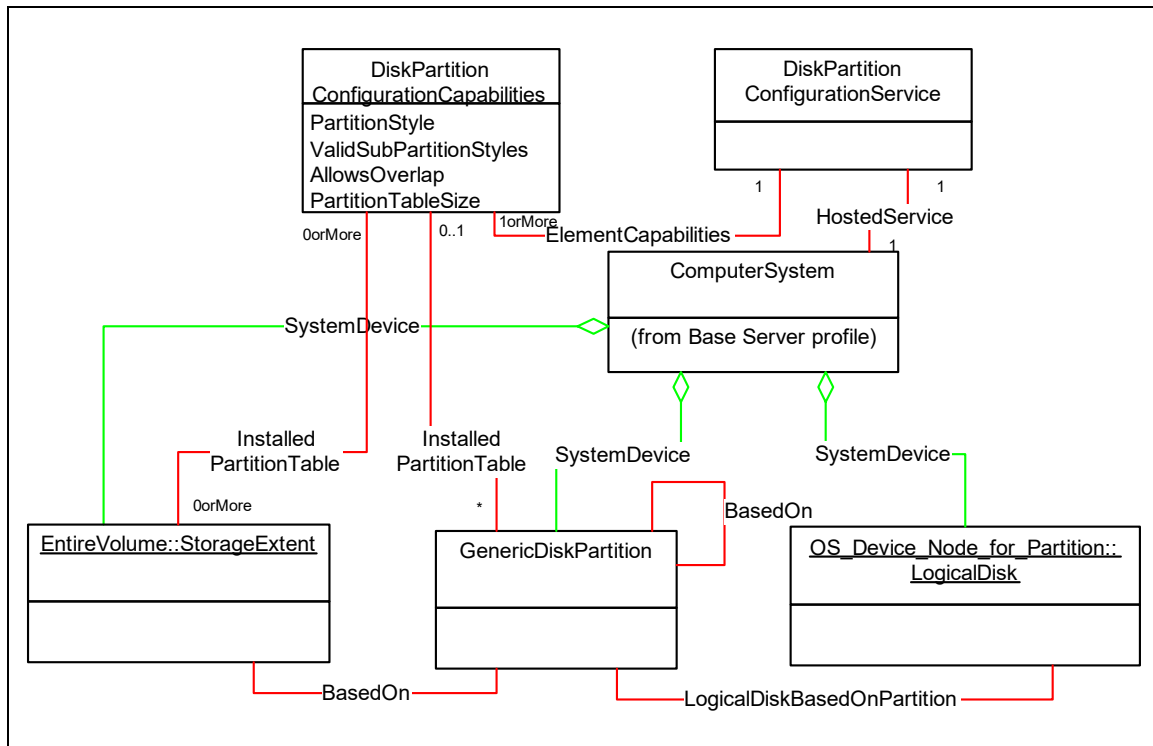


Figure 8 - Disk Partition Class Diagram

4.1.3 Background on X86 MBR Partitions

The terminology used in X86 partition applications is somewhat confusing and masks the actual configurations. The MBR style supports two tiers of partitions; up to four partitions at the entire disk volume tier and up to four partitions within each of these top-tier partitions. An MBR *primary* partition is a top-tier partition that is not sub-partitioned. An MBR *extended* partition is a top-tier partition that is sub-partitioned. An MBR *logical* partition is a sub-partition of an extended partition.

Figure 9: "Disk MBR Partition Example" represents the actual layout of an MBR drive with three usable partitions – with Windows/DOS driver letter names.

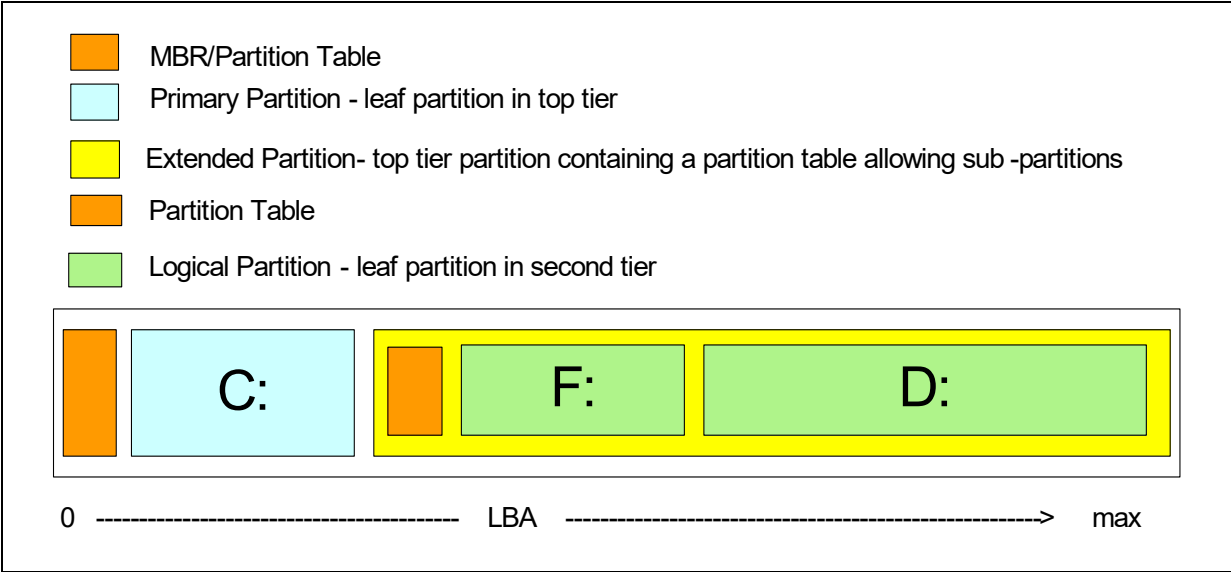


Figure 9 - Disk MBR Partition Example

C: is a primary partition and F: and D: are logical partitions that share an extended partition. Note that the partitions drive letters (C:, F:, and D:) are not in alphabetical order; the assignment of drive letters under Windows/DOS is decoupled from the partitioning logic.

Figure 10: "MBR Partition Instance Diagram" is an instance diagram of the SMI-S classes describing this configuration. Technically, the MBR/Partition tables could be considered to be small partitions. operating systems generally hide these sectors and treat the effective disk volume as starting just after the MBR. Rather than complicate the SMI-S model, these MBR areas are just ignored and the consumable block size is reduced by the appropriate value (the PartitionTableSize property of DiskPartitionConfigurationCapabilities). In the SMI-S model, the InstalledPartitionTable association to the containing extent indicates the presence of a disk label and/or partition table. In Figure 10: "MBR Partition Instance Diagram", the extent representing the entire disk volume (on the lower left) and the top-tier partition to the right each contain a partition table and are each associated to DiskPartitionConfigurationCapabilities via an InstalledPartitionTable association.

In Figure 10: "MBR Partition Instance Diagram" the StorageExtent at the lower left represents the entire disk volume and the two "top-tier" partitions are based on this extent. The LogicalDisk instances at the top represent the consumable partitions C:, F:, and D:.

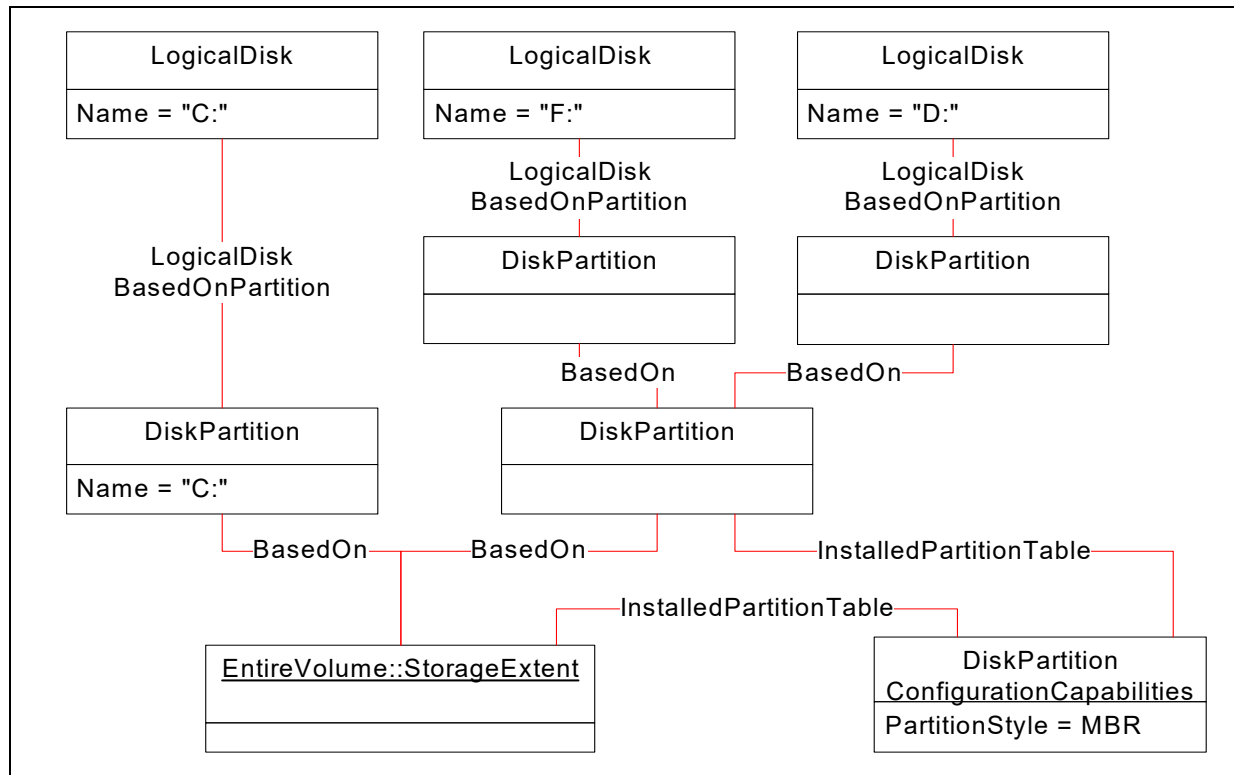


Figure 10 - MBR Partition Instance Diagram

Figure 11: "MBR and VTOC Partition Instance Diagram" models a similar configuration where the one top-tier partition contains a Solaris X86 installation. In this case, the instrumentation instantiates two instances of DiskPartitionConfigurationCapabilities, one for the top-tier MBR partition table and one for the vtoc partition table.

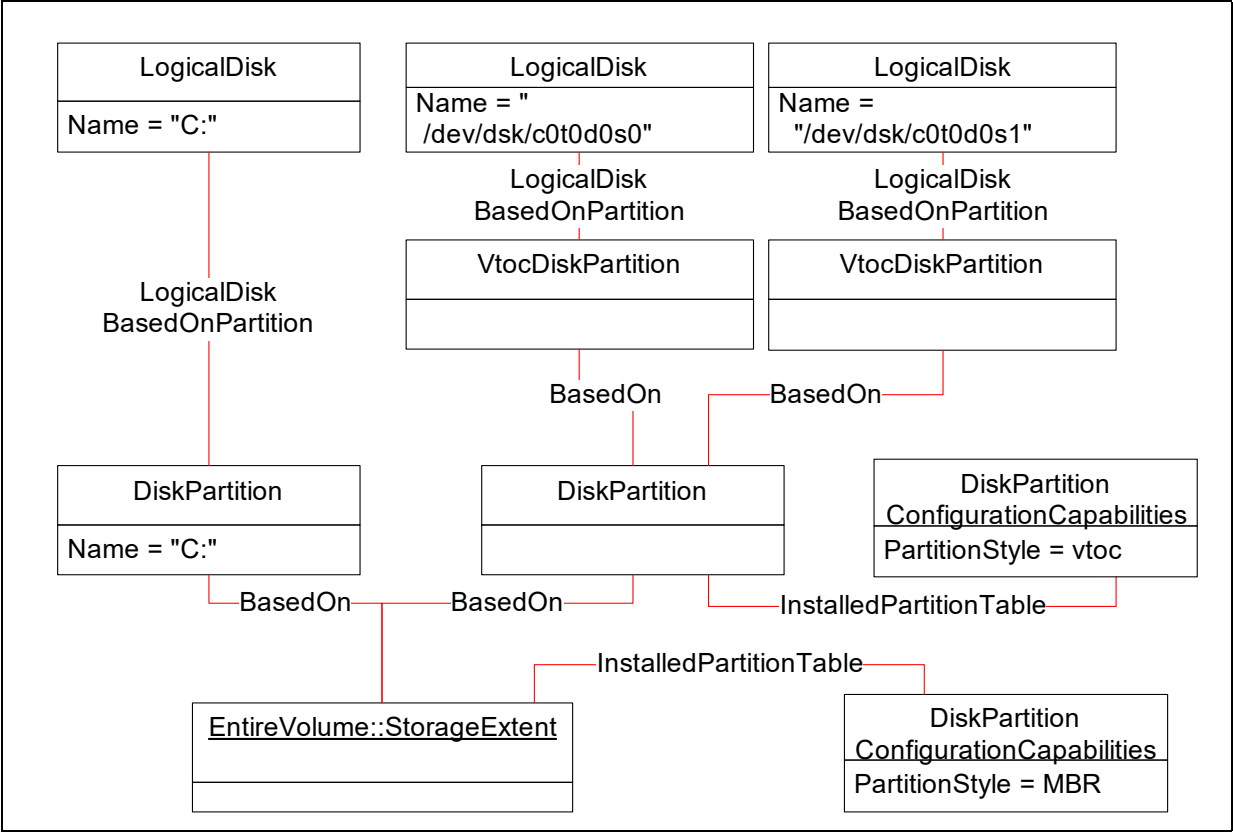


Figure 11 - MBR and VTOC Partition Instance Diagram

Table 1 summarizes likely values for capabilities properties and suggested Name properties on various operating systems

Table 1 - Capabilities Properties

Property	X86 MBR	vtoc	GPT			
			Win	Linux	Solaris SPARC	Solaris X86
Overlap Allowed	Depends on applications	true	false			
MaxCapacity	2 terabytes (2^32 blocks)	2 terabytes	2^64 blocks			
MaxNumberOfPartitions	4	8	128	15	127	127

The sizes and starting/ending addresses shall be consistent between the associated LogicalDisk, DiskPartition, and LogicalDisk instances. Figure 12: "Partition Instance Diagram for Size/Address Rules" shows the classes with size information.

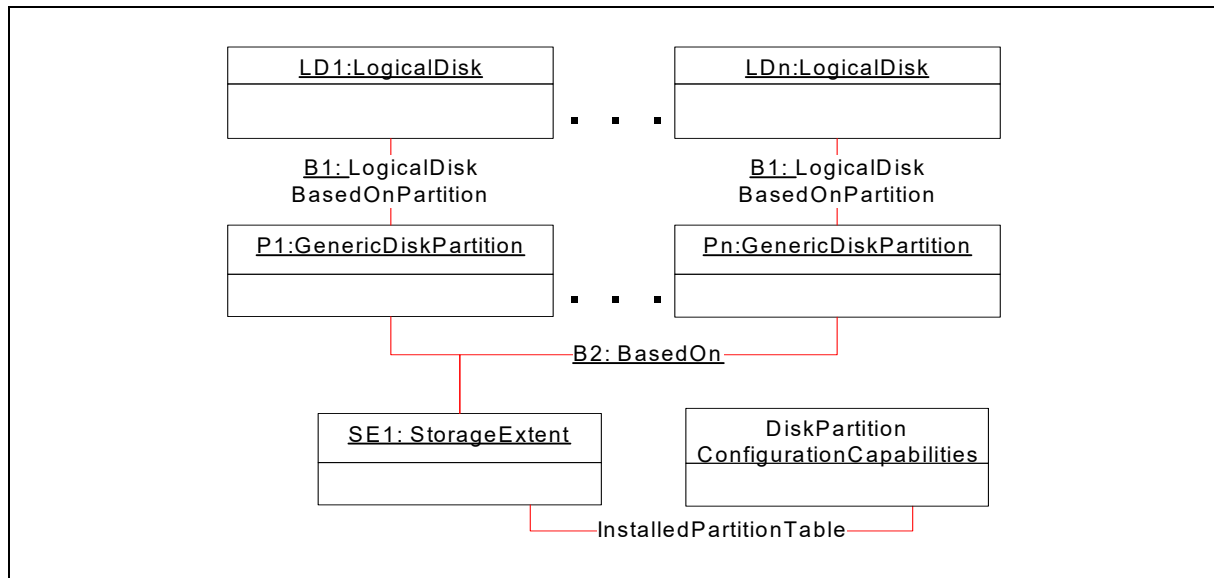


Figure 12 - Partition Instance Diagram for Size/Address Rules

In this diagram, partitions P1,... Pn are all based on the same underlying disk volume (or partition) SE1.

- The NumberOfBlocks shall be the same for a LogicalDisk and its underlying partition (for example, LD1 and P1 in the diagram).
- The StartingAddress in the LogicalDiskBasedOnPartition associations (B1 in the diagram) between a LogicalDisk and its underlying partition shall be 0. The EndingAddress in this association shall be one less than NumberOfBlocks from either the LogicalDisk or partition.
- The NumberOfBlocks for each partition (P1, ... Pn) shall be equal to the values of EndingAddress-StartingAddress+1 of the underlying BasedOn association (B2 in the diagram).
- DiskPartitionConfigurationCapabilities.PartitionTableSize shall hold the total number of blocks consumed by metadata (volume label, boot record, and partition tables) for the associated StorageExtent. For MBR and VTOC styles, this is a fixed value. For GPT, this value could in theory be larger for large extents. Separate instances of DiskPartitionConfigurationCapabilities shall be instantiated as needed to allow different values of PartitionTableSize.
- The size of maintenance tracks or cylinders shall not be included StorageExtent.NumberOfBlocks. This size may be included in DiskPartitionConfigurationCapabilities.PartitionTableSize.
- If DiskPartitionConfigurationCapabilities.OverlapAllowed is false, then the sum of the NumberOfBlocks properties for all partitions plus DiskPartitionConfigurationCapabilities.PartitionTableSize shall not exceed the value of NumberOfBlocks for the underlying StorageExtent. Other than that, there is no guaranteed relationship between StorageExtent.NumberOfBlocks and the sum of the NumberOfBlock values for partitions BasedOn the StorageExtent.

4.2 Health and Fault Management Considerations

No health information is required in LogicalDisk or partition instances. Clients should assume that the health-related properties of the underlying StorageExtent apply to all partitions and LogicalDisks based on that extent.

4.3 Supported Profiles and Packages

Not defined in this document

4.4 Methods of the Profile

4.4.1 SetPartitionStyle

This method installs a partition table on an extent of the specified partition style, creates DiskPartition instances if SettingStyleInstantiatedPartitions is non-zero, and BasedOn associations between the underlying extent and the new partition instances. As a side effect, the usable block size of the underlying extent is reduced by the block size of the metadata reserved by the partition table and associated metadata. This size is in the PartitionTableSize property of the associated DiskPartitionConfigurationCapabilities instance.

```
uint32 SetPartitionStyle (
    [IN, Description (
        "A reference to the extent (volume or partition) where "
        "this style (partition table) will be installed.")]
    CIM_StorageExtent REF Extent,

    [IN, Description (
        "A reference to the "
        "DiskPartitionConfigurationCapabilities instance "
        "describing the desired partition style.")]
    CIM_DiskPartitionConfigurationCapabilities REF PartitionStyle );
```

4.4.2 CreateOrModifyPartition

This method creates a new partition if the Partition parameter is null or modifies the partition specified. If the starting and ending address parameters are null, the resulting partition will occupy the entire underlying extent. If a the DeviceFileName parameter is non-null, a LogicalDisk instance is created and associated via LogicalDiskBasedOnPartition to the partition. The underlying extent shall be associated to a capabilities class describing the installed partition style (partition table); this association is established using

```
uint32 CreateOrModifyPartition (
    [IN, Description (
        "A reference to the underlying extent the partition is "
        "base on.")]
    CIM_StorageExtent REF extent,

    [IN, Description (
        "The starting block number.")]
    uint64 StartingAddress,

    [IN, Description (
        "The ending block number.")]
    uint64 EndingAddress,

    [IN, Description (
        "The platform-specific special file name to be assigned ")
```

```

        "to the LogicalDisk instance BasedOn the new "
        "DiskPartition instance.")]
string DeviceFileName,

[IN, OUT, Description (
    "A reference an existing partition instance to modify or "
    "null to request a new partition.")]
CIM_GenericDiskPartition REF Partition);

```

Intrinsic delete operation will delete the disk partition and all storage extents that are Dependent on the disk partition.

4.5 Client Considerations and Recipes

4.5.1 Client Considerations

A client discovers partition configuration support by looking for instances of DiskPartitionConfigurationService. If no service instances are available, then this operating system does not support disk partitions and the client can assume that any LogicalDisk instance is consumable by applications (such as volume managers or filesystems). For operating systems that do support partitioning, the client can discover whether a particular extent is partitioned by looking for a InstalledPartitionTable instance associated with the extent. The client can discover the existing partition configuration by following BasedOn associations between the extent and GenericDiskPartition instances.

For each discovered service, there shall be one or more instances of DiskPartitionConfigurationCapabilities. There is exactly one capabilities instance per Partition Style. If multiple capabilities instances are discovered, the client should look at the PartitionStyle property to determine the services that apply to entire disk volumes and those that apply to partitions.

4.5.2 Recipes

No recipes are included in this version of the standard.

4.6 CIM Elements

4.6.1 Overview

Table 2 describes the CIM elements for Disk Partition.

Table 2 - CIM Elements for Disk Partition

Element Name	Requirement	Description
4.6.2 CIM_BasedOn (Partition to Extent)	Mandatory	The disk partitions that are dependent on a storage extent.
4.6.3 CIM_BasedOn (Partition to Partition)	Mandatory	The disk partitions that are dependent on a disk partition.
4.6.4 CIM_DiskPartition	Optional	
4.6.5 CIM_DiskPartitionConfigurationCapabilities	Mandatory	
4.6.6 CIM_DiskPartitionConfigurationService	Mandatory	
4.6.7 CIM_ElementCapabilities	Mandatory	
4.6.8 CIM_ElementConformsToProfile (DiskPartitionConfigurationService to Disk Partition RegisteredProfile)	Mandatory	Ties the DiskPartitionConfigurationService to the registered profile for Disk Partition.

Table 2 - CIM Elements for Disk Partition

Element Name	Requirement	Description
4.6.9 CIM_GPTDiskPartition	Optional	
4.6.10 CIM_GenericDiskPartition	Mandatory	
4.6.11 CIM_HostedService	Mandatory	
4.6.12 CIM_InstalledPartitionTable (Capabilities to Extent)	Mandatory	The disk partition capabilities for disk partitions that are dependent on a storage extent.
4.6.13 CIM_InstalledPartitionTable (Capabilities to Partition)	Mandatory	The disk partition capabilities for disk partitions that are dependent on a storage extent.
4.6.14 CIM_LogicalDisk	Optional	
4.6.15 CIM_LogicalDiskBasedOnPartition (LogicalDisk to Partition)	Conditional	Conditional Requirement: Required if CIM_LogicalDisk is instantiated. The logical disks that are dependent on a disk partition.
4.6.16 CIM_StorageExtent	Mandatory	
4.6.17 CIM_SystemDevice (System to Extent)	Mandatory	The storage extent managed by a Base Server profile computer system.
4.6.18 CIM_SystemDevice (System to LogicalDisk)	Conditional	Conditional Requirement: Required if CIM_LogicalDisk is instantiated. The logical disk that is managed by a Base Server profile computer system.
4.6.19 CIM_SystemDevice (System to Partition)	Mandatory	The disk partition managed by a Base Server profile computer system.
4.6.20 CIM_VTOCDiskPartition	Optional	
SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_GenericDiskPartition	Mandatory	Partition Creation.
SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_GenericDiskPartition	Mandatory	Partition Deletion.

4.6.2 CIM_BasedOn (Partition to Extent)

Created By: Extrinsic: CreateOrModifyPartition

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 3 describes class CIM_BasedOn (Partition to Extent).

Table 3 - SMI Referenced Properties/Methods for CIM_BasedOn (Partition to Extent)

Properties	Flags	Requirement	Description & Notes
StartingAddress		Mandatory	
EndingAddress		Mandatory	
Dependent		Mandatory	Reference to a CIM_GenericDiskPartition that is based on the Antecedent
Antecedent		Mandatory	Reference to a CIM_StorageExtent

4.6.3 CIM_BasedOn (Partition to Partition)

Created By: Extrinsic: CreateOrModifyPartition

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 4 describes class CIM_BasedOn (Partition to Partition).

Table 4 - SMI Referenced Properties/Methods for CIM_BasedOn (Partition to Partition)

Properties	Flags	Requirement	Description & Notes
StartingAddress		Mandatory	
EndingAddress		Mandatory	
Antecedent		Mandatory	Reference to a CIM_GenericDiskPartition
Dependent		Mandatory	Reference to a CIM_GenericDiskPartition that is based on the Antecedent

4.6.4 CIM_DiskPartition

CIM_DiskPartition (MBR) is subclassed from CIM_GenericDiskPartition.

Created By: Extrinsic: CreateOrModifyPartition

Modified By: Static

Deleted By: Static

Requirement: Optional

Table 5 describes class CIM_DiskPartition.

Table 5 - SMI Referenced Properties/Methods for CIM_DiskPartition

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	
SystemName		Mandatory	
CreationClassName		Mandatory	
DeviceID		Mandatory	
Name		Mandatory	
NumberOfBlocks		Optional	
NameFormat		Optional	
NameNamespace		Optional	
PartitionSubtype		Optional	
PartitionType		Optional	
PrimaryPartition		Optional	

4.6.5 CIM_DiskPartitionConfigurationCapabilities

Created By: Static

Modified By: Static
 Deleted By: Static
 Requirement: Mandatory

Table 6 describes class CIM_DiskPartitionConfigurationCapabilities.

Table 6 - SMI Referenced Properties/Methods for CIM_DiskPartitionConfigurationCapabilities

Properties	Flags	Requirement	Description & Notes
InstanceID		Mandatory	Shall specify the unique identifier for an instance of this class within the Implementation namespace.
ElementName		Mandatory	User friendly name.
PartitionStyle		Mandatory	The partition style (i.e partition table type) associated with this capabilities instance.
ValidSubPartitionStyles		Optional	Some partitions can act as a container for other partitions.
MaxNumberOfPartitions		Mandatory	The maximum number of partitions that can be BasedOn the Underlying extent.
MaxCapacity		Mandatory	The largest partition size (in blocks) of this style supported on this platform.
OverlapAllowed		Mandatory	The platform supports partitions with overlapping address ranges.
PartitionTableSize		Mandatory	The number of blocks occupied by the partition table and other metadata.

4.6.6 CIM_DiskPartitionConfigurationService

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

Table 7 describes class CIM_DiskPartitionConfigurationService.

Table 7 - SMI Referenced Properties/Methods for CIM_DiskPartitionConfigurationService

Properties	Flags	Requirement	Description & Notes
CreationClassName		Mandatory	
Name		Mandatory	
SystemCreationClassName		Mandatory	
SystemName		Mandatory	
PartitioningSchemes		Optional	Describes the partitioning schemes supported by the platform.
SetPartitionStyle()		Mandatory	
CreateOrModifyPartition()		Mandatory	

4.6.7 CIM_ElementCapabilities

Created By: Static
 Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 8 describes class CIM_ElementCapabilities.

Table 8 - SMI Referenced Properties/Methods for CIM_ElementCapabilities

Properties	Flags	Requirement	Description & Notes
ManagedElement		Mandatory	Reference to the DiskPartitionConfigurationService.
Capabilities		Mandatory	Reference to the DiskPartitionConfigurationCapabilities.

4.6.8 CIM_ElementConformsToProfile (DiskPartitionConfigurationService to Disk Partition RegisteredProfile)

The CIM_ElementConformsToProfile ties DiskPartitionConfigurationService to the registered profile for Disk Partition.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 9 describes class CIM_ElementConformsToProfile (DiskPartitionConfigurationService to Disk Partition RegisteredProfile).

Table 9 - SMI Referenced Properties/Methods for CIM_ElementConformsToProfile (DiskPartitionConfigurationService to Disk Partition RegisteredProfile)

Properties	Flags	Requirement	Description & Notes
ManagedElement		Mandatory	A DiskPartitionConfigurationService instance that represents the Disk Partition.
ConformantStandard		Mandatory	RegisteredProfile instance describing the Disk Partition profile.

4.6.9 CIM_GPTDiskPartition

CIM_GPTDiskPartition (GPT) is subclassed from CIM_GenericDiskPartition.

Created By: Extrinsic: CreateOrModifyPartition

Modified By: Static

Deleted By: Static

Requirement: Optional

Table 10 describes class CIM_GPTDiskPartition.

Table 10 - SMI Referenced Properties/Methods for CIM_GPTDiskPartition

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

Table 10 - SMI Referenced Properties/Methods for CIM_GPTDiskPartition

Properties	Flags	Requirement	Description & Notes
DeviceID		Mandatory	
Name		Mandatory	
NumberOfBlocks		Optional	
PartitionType		Optional	

4.6.10 CIM_GenericDiskPartition

GenericDiskPartition is used for subclassing the various partition styles, including MBR, GPT, and VTOC.

Created By: Extrinsic: CreateOrModifyPartition

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 11 describes class CIM_GenericDiskPartition.

Table 11 - SMI Referenced Properties/Methods for CIM_GenericDiskPartition

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	
SystemName		Mandatory	
CreationClassName		Mandatory	
DeviceID		Mandatory	
Name		Mandatory	
NumberOfBlocks		Optional	

4.6.11 CIM_HostedService

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 12 describes class CIM_HostedService.

Table 12 - SMI Referenced Properties/Methods for CIM_HostedService

Properties	Flags	Requirement	Description & Notes
Antecedent		Mandatory	Reference to the ComputerSystem in Base Server.
Dependent		Mandatory	Reference to the DiskPartitionConfigurationService.

4.6.12 CIM_InstalledPartitionTable (Capabilities to Extent)

Created By: Static

Modified By: Static
 Deleted By: Static
 Requirement: Mandatory

Table 13 describes class CIM_InstalledPartitionTable (Capabilities to Extent).

Table 13 - SMI Referenced Properties/Methods for CIM_InstalledPartitionTable (Capabilities to Extent)

Properties	Flags	Requirement	Description & Notes
Dependent		Mandatory	Reference to a CIM_StorageExtent
Antecedent		Mandatory	Reference to a CIM_DiskPartitionConfigurationCapabilities

4.6.13 CIM_InstalledPartitionTable (Capabilities to Partition)

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

Table 14 describes class CIM_InstalledPartitionTable (Capabilities to Partition).

Table 14 - SMI Referenced Properties/Methods for CIM_InstalledPartitionTable (Capabilities to Partition)

Properties	Flags	Requirement	Description & Notes
Dependent		Mandatory	Reference to a CIM_GenericDiskPartition
Antecedent		Mandatory	Reference to a CIM_DiskPartitionConfigurationCapabilities

4.6.14 CIM_LogicalDisk

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

Table 15 describes class CIM_LogicalDisk.

Table 15 - SMI Referenced Properties/Methods for CIM_LogicalDisk

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	
SystemName		Mandatory	
CreationClassName		Mandatory	
DeviceID		Mandatory	
Name		Mandatory	
NameFormat		Mandatory	OS Device Name.
NameNamespace		Mandatory	OS Device Namespace.

Table 15 - SMI Referenced Properties/Methods for CIM_LogicalDisk

Properties	Flags	Requirement	Description & Notes
OperationalStatus		Mandatory	
NumberOfBlocks		Optional	

4.6.15 CIM_LogicalDiskBasedOnPartition (LogicalDisk to Partition)

Created By: Extrinsic: CreateOrModifyPartition

Modified By: Static

Deleted By: Static

Requirement: Required in CIM_LogicalDisk is instantiated.

Table 16 describes class CIM_LogicalDiskBasedOnPartition (LogicalDisk to Partition).

Table 16 - SMI Referenced Properties/Methods for CIM_LogicalDiskBasedOnPartition (LogicalDisk to Partition)

Properties	Flags	Requirement	Description & Notes
StartingAddress		Mandatory	Shall be 0.
EndingAddress		Mandatory	Shall be one less than NumberOfBlocks from either the LogicalDisk or Partition.
Dependent		Mandatory	Reference to CIM_LogicalDisk
Antecedent		Mandatory	Reference to CIM_GenericDiskPartition

4.6.16 CIM_StorageExtent

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 17 describes class CIM_StorageExtent.

Table 17 - SMI Referenced Properties/Methods for CIM_StorageExtent

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	
SystemName		Mandatory	
CreationClassName		Mandatory	
DeviceID		Mandatory	
Name		Mandatory	
OperationalStatus		Mandatory	
NumberOfBlocks		Optional	

4.6.17 CIM_SystemDevice (System to Extent)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 18 describes class CIM_SystemDevice (System to Extent).

Table 18 - SMI Referenced Properties/Methods for CIM_SystemDevice (System to Extent)

Properties	Flags	Requirement	Description & Notes
GroupComponent		Mandatory	Reference to the ComputerSystem in Base Server.
PartComponent		Mandatory	

4.6.18 CIM_SystemDevice (System to LogicalDisk)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required in CIM_LogicalDisk is instantiated.

Table 19 describes class CIM_SystemDevice (System to LogicalDisk).

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

Properties	Flags	Requirement	Description & Notes
GroupComponent		Mandatory	Reference to the ComputerSystem in Base Server.
PartComponent		Mandatory	Reference to CIM_LogicalDisk

4.6.19 CIM_SystemDevice (System to Partition)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 20 describes class CIM_SystemDevice (System to Partition).

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

Properties	Flags	Requirement	Description & Notes
GroupComponent		Mandatory	Reference to the ComputerSystem in Base Server.
PartComponent		Mandatory	

4.6.20 CIM_VTOCDiskPartition

CIM_VTOCDiskPartition (VTOC) is subclassed from CIM_GenericDiskPartition.

Created By: Extrinsic: CreateOrModifyPartition

Modified By: Static

Deleted By: Static

Requirement: Optional

Table 21 describes class CIM_VTOCDiskPartition.

Table 21 - SMI Referenced Properties/Methods for CIM_VTOCDiskPartition

Properties	Flags	Requirement	Description & Notes
SystemCreationClassName		Mandatory	
SystemName		Mandatory	
CreationClassName		Mandatory	
DeviceID		Mandatory	
Name		Mandatory	
NumberOfBlocks		Optional	
AsciiLabel		Optional	
Flags		Optional	
Tag		Optional	

EXPERIMENTAL

DEPRECATED

5 FC HBA Profile

The functionality of the FC HBA profile has been subsumed by 6 Storage HBA Profile.

See the latest version of this profile in SMI-S Version 1.6.1 Revision 5

DEPRECATED

EXPERIMENTAL
6 Storage HBA Profile**6.1 Synopsis****Profile Name:** Storage HBA (Component Profile)**Version:** 1.6.1**Organization:** SNIA**Central Class:** PortController**Scoping Class:** ComputerSystem**Related Profiles:** Table 22 describes the related profiles for Storage HBA.**Table 22 - Related Profiles for Storage HBA**

Profile Name	Organization	Version	Requirement	Description
Software Inventory	SNIA	1.0.0	Mandatory	
Software Update	DMTF	1.0.0	Optional	
Physical Asset	DMTF	1.0.2	Optional	
FC HBA Diagnostics	DMTF	1.1.0a	Optional	See DSP1104, version 1.1.0a
Diagnostic Job Control	DMTF	1.0.0b	Conditional	Conditional requirement: Required if the FC HBA Diagnostics profile is supported. See DSP1119, version 1.1.0b
Indications	DMTF	1.2.2	Mandatory	See DSP1054, version 1.2.2
SAS Initiator Ports	SNIA	1.7.0	Support for at least one is mandatory.	Conditional requirement: Required if the FC HBA Diagnostics profile is supported.
FC Initiator Ports	SNIA	1.7.0		Conditional requirement: Required if the FC HBA Diagnostics profile is supported.
FCoE Initiator Ports	SNIA	1.7.0		Conditional requirement: Required if the FC HBA Diagnostics profile is supported.

6.2 Description

The Storage HBA Profile represents the manageable elements of an HBA and optionally, the storage connected to it. An HBA can be connected to disks contained within a server's internal drive cage or an external drive enclosure or array. The profile does not include enclosure management of storage devices connected to the HBA. Storage device enclosure management is performed via the Storage Enclosure Profile.

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. Only the classes unique to a HBA are described by this profile. Other classes that are common to other profiles, reference to by the Storage HBA Profile, may be found in their respective section within this standard, DMTF specifications, or the DMTF CIM schema specification.

6.3 Implementation**6.3.1 Health and Fault Management Consideration**

Not defined in this document

6.3.2 Cascading Considerations

Not defined in this document

6.3.3 Storage HBA Model Overview

The PortController class is the central class of the Storage HBA Profile. It represents an instance of an HBA. The PortController shall be associated to one or more instances of LogicalPort (defined in initiator port profiles) using the ControlledBy association. The PortController shall be associated to the ComputerSystem (from a referencing profile) using the SystemDevice association. The PortController shall also be associated to Product using the ProductElementComponent association; properties of Product provide information about the HBAs manufacturer and model.

6.3.3.1 PhysicalPackage Requirements

When the instrumentation is representing a physical HBA (as opposed to virtual HBA in a guest virtual machine), PortController shall be associated to PhysicalPackage (or a subclass such as Card), as shown in Figure 13. Physical Package shall be associated to PortController via Realizes and Product via ProductPhysicalComponent. If the instrumentation is running in a guest virtual machine and representing a virtual HBA, PhysicalPackage, Realizes, and ProductPhysicalComponent shall not be instantiated. See 6.3.12 for additional virtual system considerations.

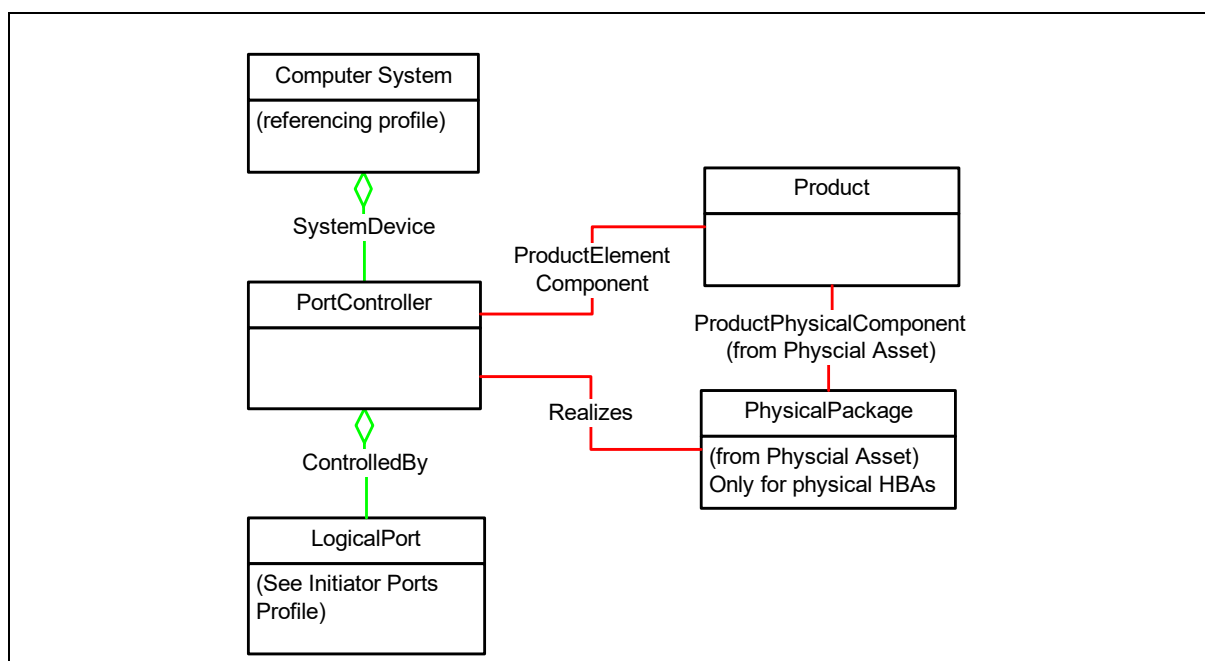


Figure 13 - Model Overview

6.3.4 CIM_ComputerSystem

In Storage HBA, the ComputerSystem Class in the diagrams represents the host containing the HBA and is not defined as part of this profile. Typically, this ComputerSystem will be defined as part of a shim profile or the Base Server Profile. Many of the other classes in the Storage HBA Profile are associated to the instance of ComputerSystem that represents the Host. This includes drives, logical ports, and physical cards.

6.3.5 Profile Registration Profile

For the Storage HBA Profile, the scoping class methodology of profile registration shall be used, as required by the Server Profile. The scoping class is the ComputerSystem in the referencing profile.