



# **Storage Management Technical Specification, Part 5 Fabric**

**Version 1.2.0, Revision 6**

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

***SNIA Technical Position***

***22 October, 2007***



## Errata/Change Log

20071022

No errata have been identified for 1.2.0.

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

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

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

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

Copyright © 2003-2007 Storage Networking Industry Association.

## INTENDED AUDIENCE

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

## DISCLAIMER

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

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

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

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

## CHANGES TO THE SPECIFICATION

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

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

## TYPOGRAPHICAL CONVENTIONS

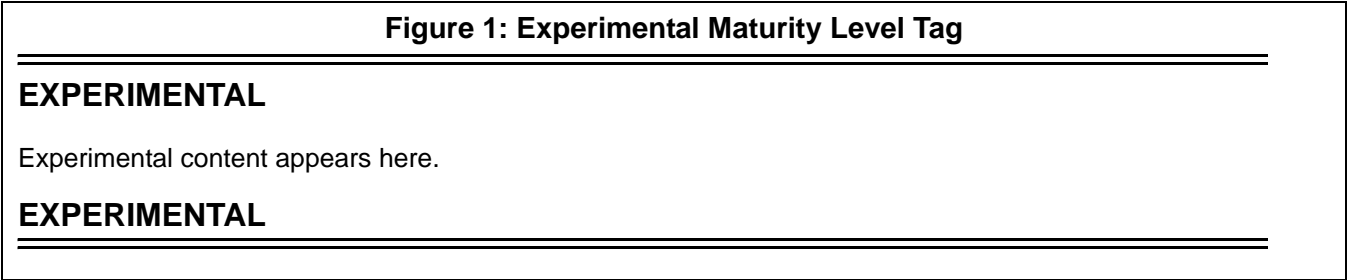
This specification has been structured to convey both the formal requirements and assumptions of the SMI-S API and its emerging implementation and deployment lifecycle. Over time, the intent is that all content in the specification will represent a mature and stable design, be verified by extensive implementation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

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

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

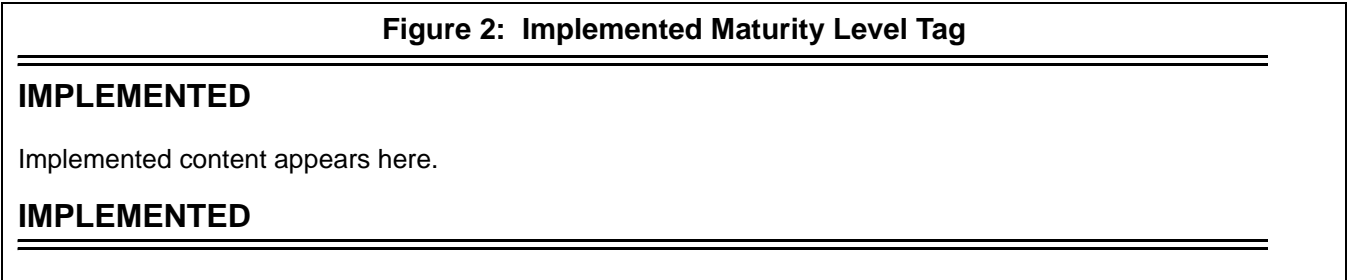
**Experimental Maturity Level**

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



**Implemented Maturity Level**

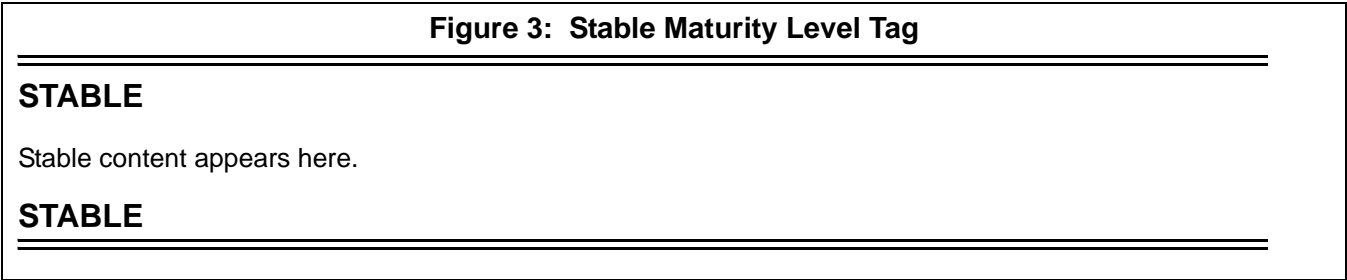
Profiles for which initial implementations have been completed are classified as “Implemented”. This indicates that at least two different vendors have implemented the profile, including at least one provider implementation. At this maturity level, the underlying architecture and modeling are stable,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



**Stable Maturity Level**

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

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



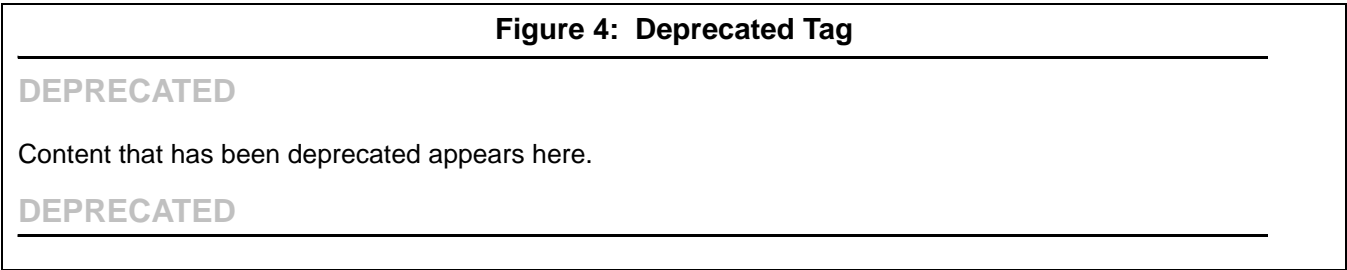
Finalized Maturity Level

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

Deprecated Material

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

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



## USAGE

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

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

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

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



# Contents

|  |           |
|--|-----------|
| Errata/Change Log .....  | iii       |
| List of Tables .....   | xiii      |
| List of Figures .....  | xix       |
| Foreword.....  | xxi       |
| <b>1. Scope .....</b>  | <b>1</b>  |
| <b>2. Normative References .....</b>                                   | <b>3</b>  |
| 2.1 Approved references .....  | 3         |
| 2.2 References under development .....                                 | 3         |
| 2.3 Other references.....  | 3         |
| <b>3. Terms and Definitions .....</b>                                  | <b>5</b>  |
| <b>4. Fabric Profile.....</b>  | <b>7</b>  |
| 4.1 Description.....   | 7         |
| 4.2 Health and Fault Management .....                                  | 12        |
| 4.3 Cascading Considerations.....                                      | 12        |
| 4.4 Supported Subprofiles and Package .....                            | 12        |
| 4.5 Methods of this Profile .....                                      | 12        |
| 4.6 Client Considerations and Recipes .....                            | 13        |
| 4.7 Registered Name and Version .....                                  | 18        |
| 4.8 CIM Elements .....   | 19        |
| <b>5. Enhanced Zoning and Enhanced Zoning Control Subprofile .....</b> | <b>39</b> |
| 5.1 Description.....   | 39        |
| 5.2 Health and Fault Management .....                                  | 39        |
| 5.3 Cascading Considerations.....                                      | 39        |
| 5.4 Dependencies on Profiles, Subprofiles, and Packages.....           | 39        |
| 5.5 Methods of this Profile .....                                      | 39        |
| 5.6 Client Considerations and Recipes .....                            | 39        |
| 5.7 Registered Name and Version .....                                  | 42        |
| 5.8 CIM Elements .....   | 43        |
| <b>6. Zone Control Subprofile.....</b>                                 | <b>47</b> |
| 6.1 Description.....   | 47        |
| 6.2 Durable Names and Correlatable IDs of the Profile .....            | 47        |
| 6.3 Instrumentation Requirements .....                                 | 47        |
| 6.4 Health and Fault Management .....                                  | 47        |
| 6.5 Cascading Considerations.....                                      | 47        |
| 6.6 Dependencies on Profiles, Subprofiles, and Packages.....           | 47        |
| 6.7 Methods of this Profile .....                                      | 47        |
| 6.8 Client Considerations and Recipes .....                            | 52        |
| 6.9 Registered Name and Version .....                                  | 63        |
| 6.10 CIM Elements .....  | 64        |
| <b>7. FDMI Subprofile .....</b>  | <b>67</b> |
| 7.1 Description.....   | 67        |
| 7.2 Health and Fault Management .....                                  | 67        |
| 7.3 Cascading Considerations.....                                      | 67        |
| 7.4 Dependencies on Profiles, Subprofiles, and Packages.....           | 68        |
| 7.5 Methods of this Profile .....                                      | 68        |
| 7.6 Client Considerations and Recipes .....                            | 68        |
| 7.7 Registered Name and Version .....                                  | 68        |
| 7.8 CIM Elements .....   | 69        |
| <b>8. Fabric Path Performance Subprofile .....</b>                     | <b>81</b> |
| 8.1 Description.....   | 81        |
| 8.2 Health and Fault Management .....                                  | 81        |
| 8.3 Dependencies on Profiles, Subprofiles, and Packages.....           | 81        |

|            |   |            |
|------------|---|------------|
| 8.4        | Methods of this Profile .....                             | 81         |
| 8.5        | Client Considerations and Recipes .....                   | 82         |
| 8.6        | Registered Name and Version .....                         | 82         |
| 8.7        | CIM Elements .....  | 82         |
| <b>9.</b>  | <b>Fibre Channel Security Subprofile .....</b>            | <b>89</b>  |
| 9.1        | Description.....  | 89         |
| 9.2        | Health and Fault Management Consideration .....           | 91         |
| 9.3        | Cascading Considerations.....                             | 91         |
| 9.4        | Supported Profiles, Subprofiles, and Packages .....       | 92         |
| 9.5        | Methods of the Profile.....                               | 92         |
| 9.6        | Client Considerations and Recipes .....                   | 92         |
| 9.7        | Registered Name and Version .....                         | 92         |
| 9.8        | CIM Elements .....  | 92         |
| <b>10.</b> | <b>Virtual Fabrics Subprofile .....</b>                   | <b>97</b>  |
| 10.1       | Description.....  | 97         |
| 10.2       | Health and Fault Management Consideration .....           | 99         |
| 10.3       | Cascading Considerations.....                             | 99         |
| 10.4       | Supported Profiles, Subprofiles, and Packages .....       | 100        |
| 10.5       | Methods of the Profile.....                               | 100        |
| 10.6       | Client Considerations and Recipes .....                   | 100        |
| 10.7       | Registered Name and Version .....                         | 100        |
| 10.8       | CIM Elements .....  | 100        |
| <b>11.</b> | <b>Switch Profile .....</b>                               | <b>105</b> |
| 11.1       | Description.....  | 105        |
| 11.2       | Health and Fault Management .....                         | 107        |
| 11.3       | Cascading Considerations.....                             | 108        |
| 11.4       | Dependencies on Profiles, Subprofiles, and Packages ..... | 108        |
| 11.5       | Methods of this Profile .....                             | 108        |
| 11.6       | Client Considerations and Recipes .....                   | 108        |
| 11.7       | Registered Name and Version .....                         | 125        |
| 11.8       | CIM Elements .....  | 126        |
| <b>12.</b> | <b>Switch Configuration Data Subprofile .....</b>         | <b>143</b> |
| 12.1       | Description.....  | 143        |
| 12.2       | Durable Names and Correlatable IDs of the Profile .....   | 143        |
| 12.3       | Instrumentation Requirements .....                        | 143        |
| 12.4       | Health and Fault Management .....                         | 143        |
| 12.5       | Cascading Considerations.....                             | 143        |
| 12.6       | Methods of this Profile .....                             | 143        |
| 12.7       | Client Considerations and Recipes .....                   | 144        |
| 12.8       | Registered Name and Version .....                         | 144        |
| 12.9       | CIM Elements .....  | 145        |
| <b>13.</b> | <b>Blades Subprofile .....</b>                            | <b>147</b> |
| 13.1       | Description.....  | 147        |
| 13.2       | Health and Fault Management .....                         | 147        |
| 13.3       | Cascading Considerations.....                             | 147        |
| 13.4       | Methods of this Profile .....                             | 148        |
| 13.5       | Client Considerations and Recipes .....                   | 148        |
| 13.6       | Registered Name and Version .....                         | 148        |
| 13.7       | CIM Elements .....  | 149        |
| <b>14.</b> | <b>Switch Partitioning Subprofile .....</b>               | <b>155</b> |
| 14.1       | Description.....  | 155        |
| 14.2       | Health and Fault Management Consideration .....           | 157        |
| 14.3       | Cascading Considerations.....                             | 157        |
| 14.4       | Supported Profiles, Subprofiles, and Packages .....       | 157        |

|            |   |            |
|------------|---|------------|
| 14.5       | Methods of the Profile.....                         | 157        |
| 14.6       | Client Considerations and Recipes .....             | 157        |
| 14.7       | Registered Name and Version .....                   | 157        |
| 14.8       | CIM Elements .....                                  | 158        |
| <b>15.</b> | <b>Extender Profile .....</b>                       | <b>163</b> |
| 15.1       | Description.....                                    | 163        |
| 15.2       | Health and Fault Management .....                   | 165        |
| 15.3       | Cascading Considerations.....                       | 165        |
| 15.4       | Supported Subprofiles and Packages .....            | 165        |
| 15.5       | Methods of this Profile .....                       | 165        |
| 15.6       | Client Considerations and Recipes .....             | 165        |
| 15.7       | Registered Name and Version .....                   | 176        |
| 15.8       | CIM Elements .....                                  | 177        |
| <b>16.</b> | <b>Router Profile .....</b>                         | <b>207</b> |
| <b>17.</b> | <b>iSCSI Gateway Profile .....</b>                  | <b>209</b> |
| 17.1       | Description.....                                    | 209        |
| 17.2       | Health and Fault Management Consideration .....     | 210        |
| 17.3       | Cascading Considerations.....                       | 210        |
| 17.4       | Supported Profiles, Subprofiles, and Packages ..... | 211        |
| 17.5       | Methods of the Profile.....                         | 211        |
| 17.6       | Client Considerations and Recipes .....             | 211        |
| 17.7       | Registered Name and Version .....                   | 211        |
| 17.8       | CIM Elements .....                                  | 212        |



## List of Tables

|           |  |    |
|-----------|--|----|
| Table 1.  | Supported Profiles for Fabric.....   | 12 |
| Table 2.  | Port OperationalStatus .....   | 13 |
| Table 3.  | OperationalStatus for ComputerSystem.....  | 13 |
| Table 4.  | CIM Elements for Fabric.....   | 19 |
| Table 5.  | SMI Referenced Properties/Methods for CIM_ActiveConnection.....  | 21 |
| Table 6.  | SMI Referenced Properties/Methods for CIM_AdminDomain (SAN) .....  | 22 |
| Table 7.  | SMI Referenced Properties/Methods for CIM_AdminDomain (Fabric).....  | 22 |
| Table 8.  | SMI Referenced Properties/Methods for CIM_Component .....  | 23 |
| Table 9.  | SMI Referenced Properties/Methods for CIM_ComputerSystem (Switch) .....                                      | 23 |
| Table 10. | SMI Referenced Properties/Methods for CIM_ComputerSystem (Platform) .....                                    | 24 |
| Table 11. | SMI Referenced Properties/Methods for CIM_ConnectivityCollection .....                                       | 24 |
| Table 12. | SMI Referenced Properties/Methods for CIM_ContainedDomain.....   | 25 |
| Table 13. | SMI Referenced Properties/Methods for CIM_DeviceSAPImplementation (Non-Switch to FCPort).....                | 25 |
| Table 14. | SMI Referenced Properties/Methods for CIM_DeviceSAPImplementation (Switch to FCPort) .....                   | 25 |
| Table 15. | SMI Referenced Properties/Methods for CIM_ElementCapabilities (ZoneCapabilities to fabric.) .....            | 26 |
| Table 16. | SMI Referenced Properties/Methods for CIM_ElementCapabilities (ZoneCapabilities to switch.).....             | 26 |
| Table 17. | SMI Referenced Properties/Methods for CIM_FCPort (Switch FCPort) .....                                       | 27 |
| Table 18. | SMI Referenced Properties/Methods for CIM_FCPort (Non-Switch FCPort) .....                                   | 28 |
| Table 19. | SMI Referenced Properties/Methods for CIM_HostedAccessPoint (ComputerSystem to ProtocolEndpoint) .....       | 29 |
| Table 20. | SMI Referenced Properties/Methods for CIM_HostedAccessPoint (AdminDomain to ProtocolEndpoint) .....          | 29 |
| Table 21. | SMI Referenced Properties/Methods for CIM_HostedCollection (Fabric to ConnectivityCollection).....           | 30 |
| Table 22. | SMI Referenced Properties/Methods for CIM_HostedCollection (Zones or ZoneSets to Fabric) .....               | 30 |
| Table 23. | SMI Referenced Properties/Methods for CIM_HostedCollection (Zones or ZoneSets to Switch) .....               | 30 |
| Table 24. | SMI Referenced Properties/Methods for CIM_LogicalPortGroup.....  | 31 |
| Table 25. | SMI Referenced Properties/Methods for CIM_HostedCollection (ComputerSystem to LogicalPortGroup) .....        | 31 |
| Table 26. | SMI Referenced Properties/Methods for CIM_MemberOfCollection (ConnectivityCollection to ProtocolEndpoint).32 |    |
| Table 27. | SMI Referenced Properties/Methods for CIM_MemberOfCollection (LogicalPortGroup to FCPort) .....              | 32 |
| Table 28. | SMI Referenced Properties/Methods for CIM_MemberOfCollection (ZoneSet to Zone).....                          | 33 |
| Table 29. | SMI Referenced Properties/Methods for CIM_ProtocolEndpoint .....   | 33 |
| Table 30. | SMI Referenced Properties/Methods for CIM_Zone (Active) .....  | 34 |
| Table 31. | SMI Referenced Properties/Methods for CIM_Zone (Inactive).....   | 34 |
| Table 32. | SMI Referenced Properties/Methods for CIM_ZoneCapabilities.....  | 35 |
| Table 33. | SMI Referenced Properties/Methods for CIM_ZoneMembershipSettingData.....                                     | 36 |
| Table 34. | SMI Referenced Properties/Methods for CIM_ZoneSet (Active).....  | 36 |
| Table 35. | SMI Referenced Properties/Methods for CIM_ZoneSet (Inactive) .....   | 37 |
| Table 36. | CIM Elements for Enhanced Zoning and Enhanced Zoning Control.....  | 43 |
| Table 37. | SMI Referenced Properties/Methods for CIM_HostedCollection .....   | 43 |
| Table 38. | SMI Referenced Properties/Methods for CIM_HostedCollection .....   | 44 |
| Table 39. | SMI Referenced Properties/Methods for CIM_ElementSettingData.....  | 44 |
| Table 40. | SMI Referenced Properties/Methods for CIM_MemberOfCollection.....  | 44 |
| Table 41. | SMI Referenced Properties/Methods for CIM_NamedAddressCollection .....                                       | 45 |
| Table 42. | SMI Referenced Properties/Methods for CIM_ZoneService .....  | 45 |
| Table 43. | CIM Elements for Zone Control .....  | 64 |
| Table 44. | SMI Referenced Properties/Methods for SNIA_ZoneService.....  | 64 |
| Table 45. | SMI Referenced Properties/Methods for CIM_HostedService .....  | 65 |

|           |   |     |
|-----------|---|-----|
| Table 46. | CIM Elements for FDMI .....   | 69  |
| Table 47. | SMI Referenced Properties/Methods for CIM_ComputerSystem .....  | 70  |
| Table 48. | SMI Referenced Properties/Methods for CIM_ControlledBy .....  | 70  |
| Table 49. | SMI Referenced Properties/Methods for CIM_ElementSoftwareIdentity .....                                   | 71  |
| Table 50. | SMI Referenced Properties/Methods for CIM_FCPort .....  | 71  |
| Table 51. | SMI Referenced Properties/Methods for CIM_HostedCollection .....  | 72  |
| Table 52. | SMI Referenced Properties/Methods for CIM_InstalledSoftwareIdentity .....                                 | 73  |
| Table 53. | SMI Referenced Properties/Methods for CIM_LogicalPortGroup .....  | 73  |
| Table 54. | SMI Referenced Properties/Methods for CIM_MemberOfCollection .....  | 74  |
| Table 55. | SMI Referenced Properties/Methods for CIM_PhysicalPackage .....   | 74  |
| Table 56. | SMI Referenced Properties/Methods for CIM_PortController .....  | 75  |
| Table 57. | SMI Referenced Properties/Methods for CIM_Product .....   | 75  |
| Table 58. | SMI Referenced Properties/Methods for CIM_ProductPhysicalComponent .....                                  | 76  |
| Table 59. | SMI Referenced Properties/Methods for CIM_Realizes .....  | 76  |
| Table 60. | SMI Referenced Properties/Methods for CIM_SoftwareIdentity (Firmware) .....                               | 77  |
| Table 61. | SMI Referenced Properties/Methods for CIM_SoftwareIdentity (Driver) .....                                 | 77  |
| Table 62. | SMI Referenced Properties/Methods for CIM_SoftwareIdentity (Option ROM) .....                             | 78  |
| Table 63. | SMI Referenced Properties/Methods for CIM_SystemDevice .....  | 78  |
| Table 64. | SMI Referenced Properties/Methods for CIM_SystemDevice (ComputerSystem to PortController) .....           | 79  |
| Table 65. | SMI Referenced Properties/Methods for CIM_SystemDevice (ComputerSystem to FCPort) .....                   | 79  |
| Table 66. | SMI Referenced Properties/Methods for CIM_SystemDevice (Switch to FCPort) .....                           | 79  |
| Table 67. | SMI Referenced Properties/Methods for CIM_SystemDevice (System to FCPort) .....                           | 80  |
| Table 68. | CIM Elements for FabricPathPerformance .....  | 82  |
| Table 69. | SMI Referenced Properties/Methods for CIM_Network .....   | 83  |
| Table 70. | SMI Referenced Properties/Methods for CIM_ElementStatisticalData .....                                    | 83  |
| Table 71. | SMI Referenced Properties/Methods for CIM_HostedNetworkPipe .....   | 83  |
| Table 72. | SMI Referenced Properties/Methods for CIM_EndpointOfNetworkPipe .....                                     | 84  |
| Table 73. | SMI Referenced Properties/Methods for CIM_NetworkPortStatistics .....                                     | 84  |
| Table 74. | SMI Referenced Properties/Methods for CIM_NetworkPipe .....   | 85  |
| Table 75. | SMI Referenced Properties/Methods for CIM_ProtocolEndpoint .....  | 85  |
| Table 76. | SMI Referenced Properties/Methods for CIM_MemberOfCollection .....  | 86  |
| Table 77. | SMI Referenced Properties/Methods for CIM_HostedCollection .....  | 86  |
| Table 78. | SMI Referenced Properties/Methods for CIM_StatisticsCollection .....                                      | 87  |
| Table 79. | CIM Elements for FabricSecurity .....   | 92  |
| Table 80. | SMI Referenced Properties/Methods for CIM_AuthorizationService .....                                      | 93  |
| Table 81. | SMI Referenced Properties/Methods for CIM_HostedService .....   | 93  |
| Table 82. | SMI Referenced Properties/Methods for CIM_AuthorizedPrivilege .....                                       | 94  |
| Table 83. | SMI Referenced Properties/Methods for CIM_AuthorizedTarget .....  | 94  |
| Table 84. | SMI Referenced Properties/Methods for CIM_AuthorizedSubject .....   | 94  |
| Table 85. | SMI Referenced Properties/Methods for CIM_StorageHardwareID .....   | 95  |
| Table 86. | SMI Referenced Properties/Methods for CIM_ServiceAvailableToElement (Fabric AdminDomain to Service) ..... | 95  |
| Table 87. | SMI Referenced Properties/Methods for CIM_ServiceAffectsElement (ManagedElement to Service) .....         | 96  |
| Table 88. | SMI Referenced Properties/Methods for CIM_ServiceAffectsElement (StorageHardwareID to Service) .....      | 96  |
| Table 89. | CIM Elements for FabricVirtualFabrics .....   | 100 |
| Table 90. | SMI Referenced Properties/Methods for CIM_AdminDomain (SAN) .....   | 101 |
| Table 91. | SMI Referenced Properties/Methods for SNIA_AdminDomain (Fabric) .....                                     | 101 |
| Table 92. | SMI Referenced Properties/Methods for CIM_HostedDependency (NetworkPort to FCPort) .....                  | 102 |
| Table 93. | SMI Referenced Properties/Methods for CIM_HostedDependency (Partitioning CS to Partitioned CS) .....      | 102 |
| Table 94. | SMI Referenced Properties/Methods for CIM_ComputerSystem (Partitioning) .....                             | 102 |

|            |  |     |
|------------|--|-----|
| Table 95.  | SMI Referenced Properties/Methods for CIM_ComputerSystem (Partitioned) .....   | 103 |
| Table 96.  | SMI Referenced Properties/Methods for CIM_SystemDevice (NetworkPort to ComputerSystem).....  | 103 |
| Table 97.  | SMI Referenced Properties/Methods for CIM_Component (AdminDomain to Partitioning CS).....  | 104 |
| Table 98.  | SMI Referenced Properties/Methods for CIM_NetworkPort.....   | 104 |
| Table 99.  | Supported Profiles for Switch .....  | 108 |
| Table 100. | CIM Elements for Switch .....  | 126 |
| Table 101. | SMI Referenced Properties/Methods for CIM_ComputerSystemPackage .....  | 128 |
| Table 102. | SMI Referenced Properties/Methods for CIM_ComputerSystem .....   | 128 |
| Table 103. | SMI Referenced Properties/Methods for CIM_FCSwitchCapabilities.....  | 129 |
| Table 104. | SMI Referenced Properties/Methods for CIM_FCSwitchSettings .....   | 130 |
| Table 105. | SMI Referenced Properties/Methods for CIM_ElementCapabilities (System to FCSwitchCapabilities) .....                                 | 130 |
| Table 106. | SMI Referenced Properties/Methods for CIM_ElementCapabilities (FCPort to FCPortCapabilities) .....                                   | 131 |
| Table 107. | SMI Referenced Properties/Methods for CIM_ElementSettingData (FCSwitchSettings to ComputerSystem) .....                              | 131 |
| Table 108. | SMI Referenced Properties/Methods for CIM_ElementSettingData (FCPortSettings to FCPort) .....  | 132 |
| Table 109. | SMI Referenced Properties/Methods for CIM_ElementStatisticalData (FCPortStatistics to FCPort) .....                                  | 132 |
| Table 110. | SMI Referenced Properties/Methods for CIM_ElementStatisticalData (FCPortRateStatistics to FCPort) .....                              | 132 |
| Table 111. | SMI Referenced Properties/Methods for CIM_FCPort .....   | 133 |
| Table 112. | SMI Referenced Properties/Methods for CIM_FCPortCapabilities.....  | 135 |
| Table 113. | SMI Referenced Properties/Methods for CIM_FCPortSettings .....   | 135 |
| Table 114. | SMI Referenced Properties/Methods for CIM_FCPortStatistics .....   | 136 |
| Table 115. | SMI Referenced Properties/Methods for CIM_FCPortRateStatistics .....   | 138 |
| Table 116. | SMI Referenced Properties/Methods for CIM_MemberOfCollection (NetworkPortStatistics to StatisticalCollection) ..<br>139              |     |
| Table 117. | SMI Referenced Properties/Methods for CIM_MemberOfCollection (FcPort to REdundancySet).....  | 139 |
| Table 118. | SMI Referenced Properties/Methods for CIM_HostedCollection .....   | 139 |
| Table 119. | SMI Referenced Properties/Methods for CIM_StatisticsCollection .....   | 140 |
| Table 120. | SMI Referenced Properties/Methods for CIM_SystemDevice.....  | 140 |
| Table 121. | SMI Referenced Properties/Methods for CIM_ProtocolEndpoint .....   | 141 |
| Table 122. | SMI Referenced Properties/Methods for CIM_RedundancySet .....  | 141 |
| Table 123. | CIM Elements for Switch Configuration Data .....   | 145 |
| Table 124. | SMI Referenced Properties/Methods for CIM_ComputerSystem.....  | 145 |
| Table 125. | SMI Referenced Properties/Methods for CIM_ElementSettingData.....  | 145 |
| Table 126. | SMI Referenced Properties/Methods for CIM_ConfigurationData.....   | 146 |
| Table 127. | CIM Elements for Blades.....   | 149 |
| Table 128. | SMI Referenced Properties/Methods for CIM_LogicalModule .....  | 150 |
| Table 129. | SMI Referenced Properties/Methods for CIM_ModulePort .....   | 150 |
| Table 130. | SMI Referenced Properties/Methods for CIM_PhysicalPackage .....  | 151 |
| Table 131. | SMI Referenced Properties/Methods for CIM_Product .....  | 151 |
| Table 132. | SMI Referenced Properties/Methods for CIM_ProductPhysicalComponent .....   | 152 |
| Table 133. | SMI Referenced Properties/Methods for CIM_Realizes.....  | 152 |
| Table 134. | SMI Referenced Properties/Methods for CIM_SystemDevice.....  | 153 |
| Table 135. | CIM Elements for FabricSwitchPartitioning .....  | 158 |
| Table 136. | SMI Referenced Properties/Methods for CIM_HostedDependency (NetworkPort to FCPort) .....   | 158 |
| Table 137. | SMI Referenced Properties/Methods for CIM_HostedDependency (Partitioning CS to Partitioned CS) .....                                 | 159 |
| Table 138. | SMI Referenced Properties/Methods for CIM_ComputerSystem (Partitioning) .....  | 159 |
| Table 139. | SMI Referenced Properties/Methods for CIM_SystemDevice (NetworkPort to ComputerSystem).....  | 160 |
| Table 140. | SMI Referenced Properties/Methods for CIM_NetworkPortCapabilities .....  | 160 |
| Table 141. | SMI Referenced Properties/Methods for CIM_NetworkPortSettings.....   | 161 |
| Table 142. | SMI Referenced Properties/Methods for CIM_ElementCapabilities (Association between NetworkPort and<br>NetworkPortCapabilities) ..... | 161 |

|  |     |
|--|-----|
| Table 143. SMI Referenced Properties/Methods for CIM_ElementSettingData (Association between NetworkPort and NetworkPortSettings)..... | 161 |
| Table 144. SMI Referenced Properties/Methods for CIM_NetworkPort.....  | 162 |
| Table 145. CIM Elements for Extender .....   | 177 |
| Table 146. SMI Referenced Properties/Methods for CIM_BindsTo (IPPE to REmoteSAP).....  | 180 |
| Table 147. SMI Referenced Properties/Methods for CIM_BindsTo (TCPPE to IPPE) .....   | 180 |
| Table 148. SMI Referenced Properties/Methods for CIM_BindsTo (IPPE to PE) .....  | 181 |
| Table 149. SMI Referenced Properties/Methods for CIM_BindsTo (PE to RemotePort) .....  | 181 |
| Table 150. SMI Referenced Properties/Methods for CIM_BindsTo (TCPPE to RemotePort).....  | 181 |
| Table 151. SMI Referenced Properties/Methods for CIM_BindsTo (IPPE to RemoteSAP) .....   | 182 |
| Table 152. SMI Referenced Properties/Methods for CIM_BindsTo (TCPPE to PE) .....   | 182 |
| Table 153. SMI Referenced Properties/Methods for CIM_Component .....   | 183 |
| Table 154. SMI Referenced Properties/Methods for CIM_ComputerSystem.....   | 183 |
| Table 155. SMI Referenced Properties/Methods for CIM_EndpointOfNetworkPipe (PE to NetworkPipe).....                                    | 184 |
| Table 156. SMI Referenced Properties/Methods for CIM_EndpointOfNetworkPipe (TCPPE to NetworkPipe) .....                                | 184 |
| Table 157. SMI Referenced Properties/Methods for CIM_ElementSettingData (System to FCIPSettings) .....                                 | 184 |
| Table 158. SMI Referenced Properties/Methods for CIM_ElementSettingData (System to TCPSettings) .....                                  | 185 |
| Table 159. SMI Referenced Properties/Methods for CIM_ElementSettingData (System to IPSettings).....                                    | 185 |
| Table 160. SMI Referenced Properties/Methods for CIM_ElementSettingData (IPPE to IPSettings).....                                      | 186 |
| Table 161. SMI Referenced Properties/Methods for CIM_ElementSettingData (PE to FCIPSettings) .....                                     | 186 |
| Table 162. SMI Referenced Properties/Methods for CIM_ElementSettingData (TCPPE to TCPSettings) .....                                   | 187 |
| Table 163. SMI Referenced Properties/Methods for CIM_ElementStatisticalData (EthernetPort to EthernetPortStatistics) ...               | 187 |
| Table 164. SMI Referenced Properties/Methods for CIM_ElementStatisticalData (FCPort to FCPortStatistics).....                          | 188 |
| Table 165. SMI Referenced Properties/Methods for CIM_ElementStatisticalData (IPPE to IPEndpointStatistics).....                        | 188 |
| Table 166. SMI Referenced Properties/Methods for CIM_ElementStatisticalData (TCPPE to TCPEndpointStatistics) .....                     | 189 |
| Table 167. SMI Referenced Properties/Methods for CIM_ElementStatisticalData (System to TCPStatisticalData).....                        | 189 |
| Table 168. SMI Referenced Properties/Methods for CIM_EthernetPort .....  | 189 |
| Table 169. SMI Referenced Properties/Methods for CIM_EthernetPortStatistics .....  | 190 |
| Table 170. SMI Referenced Properties/Methods for CIM_FCPort .....  | 191 |
| Table 171. SMI Referenced Properties/Methods for CIM_FCPortStatistics .....  | 193 |
| Table 172. SMI Referenced Properties/Methods for CIM_FCIPSettings .....  | 195 |
| Table 173. SMI Referenced Properties/Methods for CIM_HostedAccessPoint (ComputerSystem to ProtocolEndpoint) .....                      | 195 |
| Table 174. SMI Referenced Properties/Methods for CIM_HostedAccessPoint (ComputerSystem to TCPProtocolEndpoint) .....                   | 196 |
| Table 175. SMI Referenced Properties/Methods for CIM_HostedAccessPoint (ComputerSystem to IPProtocolEndpoint).....                     | 196 |
| Table 176. SMI Referenced Properties/Methods for CIM_HostedNetworkPipe.....  | 196 |
| Table 177. SMI Referenced Properties/Methods for CIM_IPEndpointStatistics.....   | 197 |
| Table 178. SMI Referenced Properties/Methods for CIM_IPProtocolEndpoint.....   | 198 |
| Table 179. SMI Referenced Properties/Methods for CIM_IPSettings .....  | 199 |
| Table 180. SMI Referenced Properties/Methods for CIM_Network .....   | 199 |
| Table 181. SMI Referenced Properties/Methods for CIM_NetworkPipe .....   | 200 |
| Table 182. SMI Referenced Properties/Methods for CIM_NetworkPipeComposition .....  | 200 |
| Table 183. SMI Referenced Properties/Methods for CIM_DeviceSAPImplementation.....  | 201 |
| Table 184. SMI Referenced Properties/Methods for CIM_ProtocolEndpoint .....  | 201 |
| Table 185. SMI Referenced Properties/Methods for CIM_LANEndpoint .....   | 202 |
| Table 186. SMI Referenced Properties/Methods for CIM_RemotePort .....  | 202 |
| Table 187. SMI Referenced Properties/Methods for CIM_RemoteServiceAccessPoint .....  | 203 |
| Table 188. SMI Referenced Properties/Methods for CIM_SystemDevice (System to EthernetPort) .....                                       | 203 |
| Table 189. SMI Referenced Properties/Methods for CIM_SystemDevice (System to FCPort) .....   | 204 |
| Table 190. SMI Referenced Properties/Methods for CIM_TCPEndpointStatistics.....  | 204 |



|   |     |
|---|-----|
| Table 191. SMI Referenced Properties/Methods for CIM_TCPProtocolEndpoint .....  | 205 |
| Table 192. SMI Referenced Properties/Methods for CIM_TCPSettings .....  | 205 |
| Table 193. SMI Referenced Properties/Methods for CIM_TCPStatisticalData .....   | 206 |
| Table 194. Supported Profiles for iSCSI to FC Gateway.....  | 211 |
| Table 195. CIM Elements for iSCSI to FC Gateway.....  | 212 |
| Table 196. SMI Referenced Properties/Methods for CIM_AuthorizedPrivilege.....   | 213 |
| Table 197. SMI Referenced Properties/Methods for CIM_AuthorizedSubject .....  | 213 |
| Table 198. SMI Referenced Properties/Methods for CIM_AuthorizedTarget .....   | 214 |
| Table 199. SMI Referenced Properties/Methods for CIM_Dependency (Associates ControllerConfigurationService and ProtocolController)..... | 214 |
| Table 200. SMI Referenced Properties/Methods for CIM_Dependency (Associates PrivilegeManagementService and AuthorizedPrivilege) .....   | 214 |
| Table 201. SMI Referenced Properties/Methods for CIM_ControllerConfigurationService.....  | 215 |
| Table 202. SMI Referenced Properties/Methods for CIM_HostedService (Associates ComputerSystem and ControllerConfigurationService) ..... | 216 |
| Table 203. SMI Referenced Properties/Methods for CIM_HostedService (Associates ComputerSystem and PrivilegeManagementService).....      | 216 |
| Table 204. SMI Referenced Properties/Methods for CIM_LogicalDevice .....  | 216 |
| Table 205. SMI Referenced Properties/Methods for CIM_PrivilegeManagementService .....   | 217 |
| Table 206. SMI Referenced Properties/Methods for CIM_SCSIProtocolController .....   | 218 |
| Table 207. SMI Referenced Properties/Methods for CIM_ProtocolControllerForUnit .....  | 218 |
| Table 208. SMI Referenced Properties/Methods for CIM_SAPAvailableForElement .....   | 219 |
| Table 209. SMI Referenced Properties/Methods for CIM_SCSIProtocolEndpoint.....  | 219 |
| Table 210. SMI Referenced Properties/Methods for CIM_StorageHardwareID.....   | 220 |
| Table 211. SMI Referenced Properties/Methods for CIM_ControllerConfigurationService.....  | 220 |
| Table 212. SMI Referenced Properties/Methods for CIM_StorageHardwareIDManagementService.....  | 221 |



## List of Figures

|            |   |     |
|------------|---|-----|
| Figure 1.  | Experimental Maturity Level Tag .....   | vi  |
| Figure 2.  | Implemented Maturity Level Tag.....   | vi  |
| Figure 3.  | Stable Maturity Level Tag .....   | vii |
| Figure 4.  | Deprecated Tag .....  | vii |
| Figure 5.  | Fabric Instance .....   | 9   |
| Figure 6.  | Zoning Instance (AdminDomain) .....   | 10  |
| Figure 7.  | Zoning Instance (ComputerSystem) .....  | 11  |
| Figure 8.  | FDMI Instance Diagram .....   | 67  |
| Figure 9.  | Instance Diagram.....   | 81  |
| Figure 10. | Specialization of Security Authorization Subprofile for Membership Policy .....   | 90  |
| Figure 11. | Specialization of Security Authorization Subprofile for Connectivity Policy ..... | 91  |
| Figure 12. | RegisteredProfile/Subprofile, AdminDomain, and ComputerSystem Relationships ..... | 97  |
| Figure 13. | Two Virtual Fabric and Two Partitioning Systems .....                             | 98  |
| Figure 14. | Two Virtual Fabrics and One Partitioning System .....                             | 99  |
| Figure 15. | Switch Instance Diagram .....   | 106 |
| Figure 16. | Trunking Instance Diagram.....  | 107 |
| Figure 17. | Switch Configuration Data Instance.....   | 143 |
| Figure 18. | Switch Blade Instance.....  | 147 |
| Figure 19. | Switch ComputerSystem and Partitioning System.....                                | 155 |
| Figure 20. | Switch and Partitioning System and Partitioning Ports .....                       | 156 |
| Figure 21. | Underlying System Port Settings and Capabilities.....                             | 156 |
| Figure 22. | FC Extender Node Instance.....  | 164 |
| Figure 23. | FC Extender Group Instance .....  | 165 |
| Figure 24. | iSCSI to FC Gateway Instance .....  | 210 |



## Foreword

The Fabric Part of the Storage Management Technical Specification defines management profiles for Autonomous (top level) profiles for programs and devices whose central function is providing support for storage networking. The Fabric Part includes fabric management including topology and device management for switches. The Fabric part also provides management of extenders that pass fibre channel frames over other protocols as well as a gateway that maps and translates iSCSI to Fibre Channel. As part of fabric management, this specification also has controls for fibre channel zoning and fibre channel security.

### Parts of this Standard

This standard is subdivided in the following parts:

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

### Acknowledgements

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

*Organization Represented .....Name of Representative*

Brocade Communications Systems .....John Crandall

EMC Corporation .....Kamesh Aiyer

.....Edgar St. Pierre

Hewlett-Packard.....Steve Peters

Hitachi Data Systems.....Steve Quinn

IBM.....Duane Baldwin

.....Jack Gelb

.....Mike Walker

iStor Networks, Inc. ....Scott Baker

Network Appliance .....Alan Yoder

Sun Microsystems.....Mark Carlson

Symantec .....Steve Hand

.....Paul von Behren

### SNIA Web Site

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

**SNIA Address**

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

## Clause 1: Scope

The Fabric Part of the Storage Management Technical Specification defines management profiles for Autonomous (top level) profiles for programs and devices whose central function is providing support for storage networking.

This version of the Fabric Part of the Storage Management Technical Specification includes four autonomous profiles:

- Fabric

This profile defines the model and functions of a storage network including topology and zoning control.

- Switch

This profile defines the model and functions of a Fibre Channel Switch including state, status, and control of the device and it's connections and product information,

- Extender

This profile defines the model and functions of a networking device that allows for fibre channel to be extended over other networks ,and specifically over IP (FCIP).

- iSCSI to FC Gateway

This profile defines the model and functions of a networking device that maps and switches iSCSI frames from a IP network to a fibre channel fabric.





## **Clause 2: Normative References**

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.

### **2.1 Approved references**

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

ISO/IEC 24775 Storage Management

### **2.2 References under development**

*Storage Management Technical Specification, Part 1 Common Architecture*

*Storage Management Technical Specification, Part 2 Common Profiles*

ISO/IEC 14776-452, SCSI Primary Commands - 3 (SPC-3) [ANSI INCITS 408-2005]

### **2.3 Other references**

DMTF DSP0214:2004 CIM Operations over HTTP

## Normative References

### **Clause 3: Terms and Definitions**

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



---

---

**STABLE****Clause 4: Fabric Profile****4.1 Description****4.1.1 SANS and Fabrics as AdminDomains**

A SAN and Fabric are represented in CIM by AdminDomain. A SAN contains one or more Fabrics, which are modeled as AdminDomains. The “containment” of Fabrics to SANs is through the association ContainedDomain. AdminDomain is sub-classed from System. This is significant because a SAN and a Fabric can be considered a group of components that operate together as a single system and should be/are managed as such. The relationship of the Fabrics in a SAN could be as redundant fabrics, interconnected (using the same or different transports/protocols), or not connected in any way. Even in the latter case where the Fabrics are disjoint, from an administrative perspective they may still be managed together applying common practices including naming across the Fabrics.

An AdminDomain in CIM is keyed by the property Name with an associated optional property NameFormat. Typically SANs are identified (“named”) administratively and precise naming conventions are left up to the implementation. The administrator is responsible for assuring that the names are unique within the discovery of known SANs that populate the same CIM Namespace.

For Fibre Channel Fabrics, the identifier (AdminDomain.Name) is the Fabric WWN that is the switch name of the principal switch. The AdminDomain for the Fibre Channel Fabric shall have a NameFormat of WWN.

**4.1.2 Fabrics and Topology**

A Fabric in CIM minimally contains a ConnectivityCollection and its component systems. They are associated to the Fabric by the association Component. For the purposes of this discussion, it is assumed one models both.

ConnectivityCollection represents the foundation necessary for routing (and is the reason it is defined in the Network model). A ConnectivityCollection groups a set of ProtocolEndpoints that are able to communicate with each other directly. The ProtocolEndpoint is associated to the ConnectivityCollection by MemberOfCollection. A link is represented by the association ActiveCollection, which associates two ProtocolEndpoints, defined as a connection that is currently carrying traffic or is configured to carry traffic.

It is important at this point to clarify the relationship (or use) of the ProtocolEndpoint versus the use of FCPort (discussed later). A NetworkPort (from which FCPort is subclassed) is the device that is used to represent the logical aspects of the link and data layers. The ProtocolEndpoint is used to represent the higher network layers for routing. This is best understood when thinking about Ethernet and IP, but applies to fibre channel also. When two ProtocolEndpoints are capable of communicating, the association ActiveConnection is used to represent the capability to communicate and completes the picture of the topology.

One can ultimately represent multiple ConnectivityCollections (e.g. FC, IP (over FC), and IP (FC encapsulated in IP)) for the same fibre channel fabric.

Note that in modeling SANs, Fabrics, and ConnectivityCollections, a ConnectivityCollection does not require a Fabric, and a Fabric does not require a SAN. But a SAN requires a Fabric, and a Fabric (for the purposes of this profile) requires a ConnectivityCollection.

The minimum set of requirements for this profile is based on ANSI T11 FC-GS.

**4.1.3 Systems and NetworkPorts**

As discussed in the previous section, a Port is associated to a device to represent the link layer. A NetworkPort is associated to the ProtocolEndpoint by DeviceSAPImplementation and “joins” the System and Device model to the Network model. Instantiation of DeviceSAPImplementation, ProtocolEndpoint, and ActiveConnection is not

necessary if the transceiver is not installed or the cable connecting the port to another port is not installed since the device is not capable of communicating.

Systems, or in this case ComputerSystem, represent the fabric elements that contain Ports. These are typically Hosts, Switches and Storage Systems. In Fibre Channel, these are called Platforms and Interconnect Elements. The property Dedicated in ComputerSystem allows these fabric elements to be identified. For a host, Dedicated is set to "Not Dedicated", for a switch, Dedicated is set to "Switch", and for a storage system, Dedicated is set to "Storage". The Ports on a System are associated by SystemDevice.

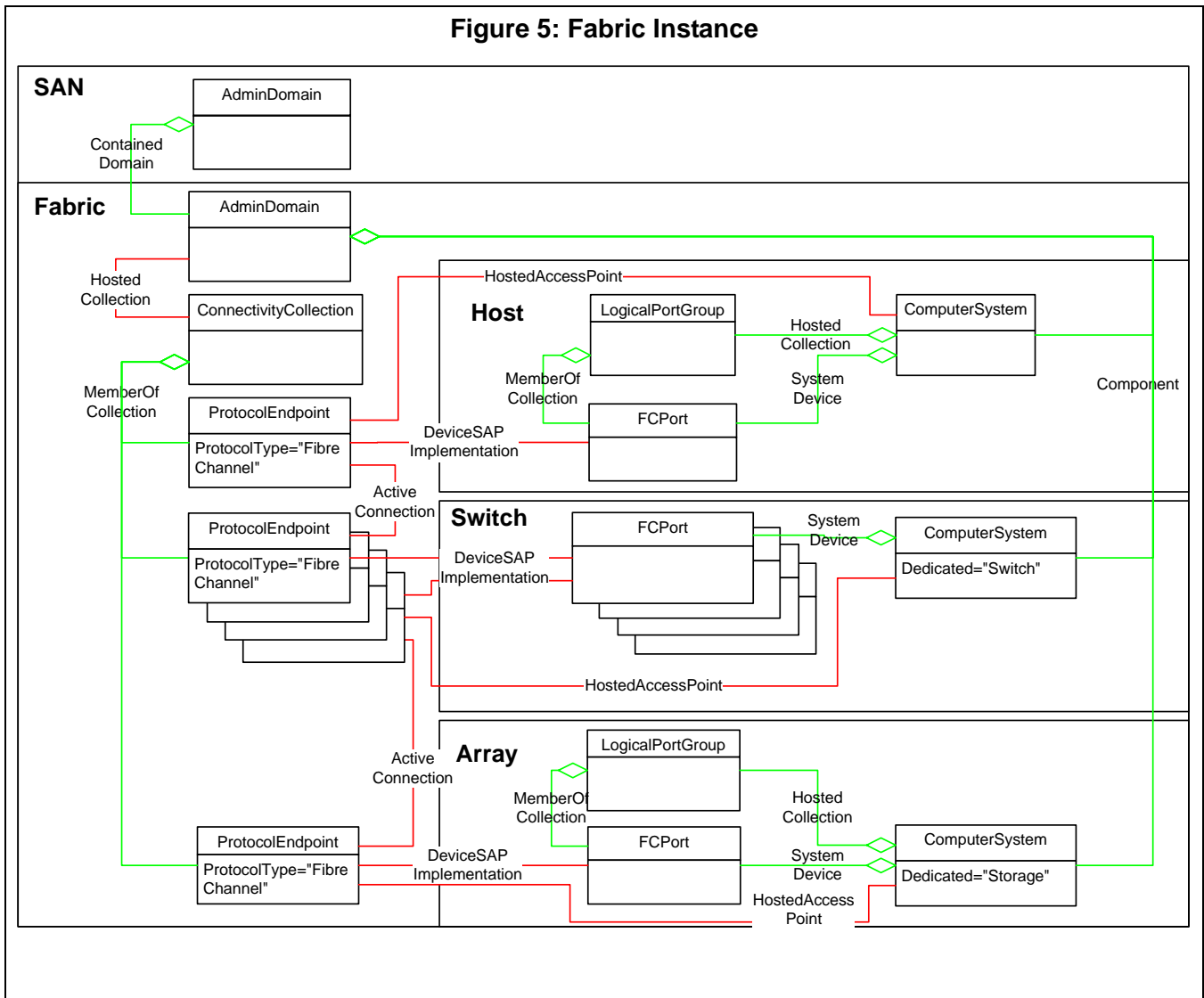
Discovery from the viewpoint of the fabric includes the end device, but often times the information available is minimal or not available. In the case of Fibre Channel, this occurs if the platform database is not populated. If this is the case, then discovery cannot tell whether a Fibre Channel Node is contained within the same platform or not. When this occurs, ComputerSystem is not instantiated and the LogicalPortGroup representing the Node and the FCPort are associated to the AdminDomain representing the Fabric.

The instrumentation needs to respond to physical fabric changes by adding or removing Logical elements to the AdminDomain. Adding an element to the fabric is straightforward, however it is not always clear when an element has been removed. The device may have been reset, or temporarily shut down, in which case it would be an element in the fabric with an "unknown" status. The lifetime of objects that can no longer be discovered is implementation specific.

If the instrumentation is unable to determine the type of platform discovered (defined in FC-GS), then the agent shall set the ComputerSystem.Dedicated property to "Unknown".

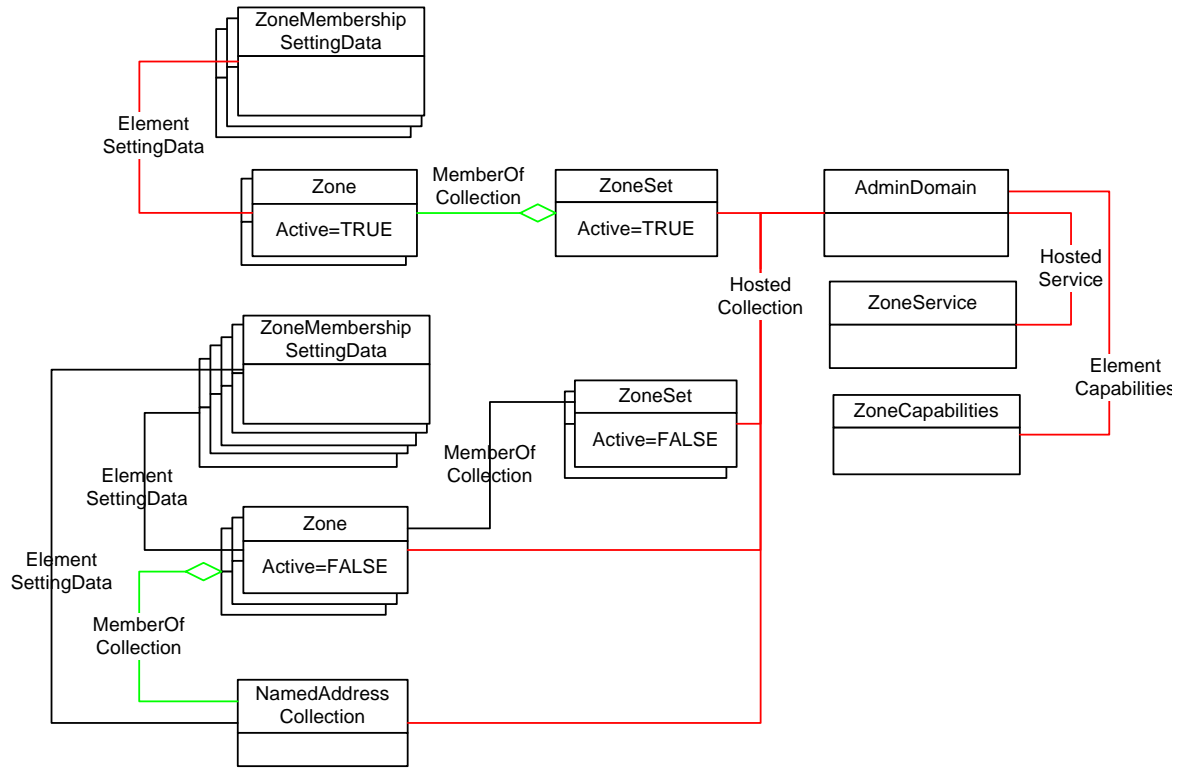
Additional identification information about ComputerSystem (e.g., DomainID) is placed in OtherIdentifyInfo property.

**Figure 5: Fabric Instance**

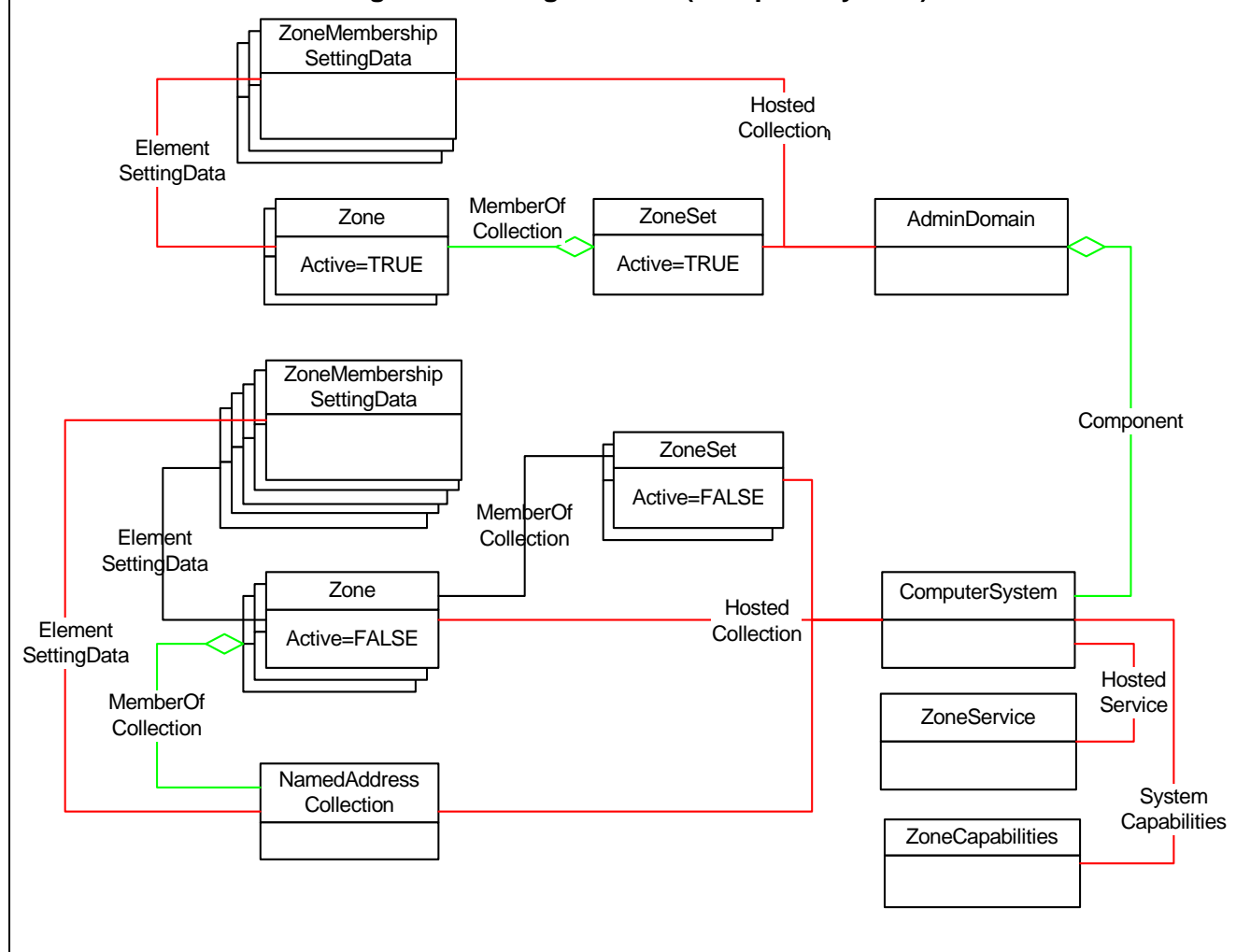


#### 4.1.4 Zoning

**Figure 6: Zoning Instance (AdminDomain)**





**Figure 7: Zoning Instance (ComputerSystem)**

The zoning model is based on ANSI FC-GS-4. This model represents the management model for defining Zone Sets, Zones, and Zone Members and “activation” of a Zone Set for a fabric. In the following discussion it may be helpful to also define the following:

- **Active ZoneSet:** the Zone Set currently enforced by the Fabric.
- **Zone Set Database:** The database of the Zone Sets not enforced by the Fabric. Referred to in this document as the Inactive Zone Sets.
- **Zoning Definitions:** a generic term used to indicate both the above concepts.

The zoning model refers to a Zone Set as ZoneSet, a Zone as Zone, ZoneAlias as a NamedAddressCollection, and Zone Member as ZoneMembershipSettingData. ZoneSets shall only contain Zones associated by MemberOfCollection. Zones shall only contain ZoneMembershipSettingData associated by ElementSettingData or NamedAddressCollections associated by MemberOfCollection. For more information with regards to NamedAddressCollection, see Clause 5: Enhanced Zoning and Enhanced Zoning Control Subprofile.

The class ZoneMembershipSettingData has two properties that indicate how the device was identified to be “zoned”. They are ConnectivityMemberType (e.g. PermanentAddress for WWN, NetworkAddress for FCID, etc.) and ConnectivityMemberID which contains the actual device identifier.

The Active Zone Set, defined by an instance of ZoneSet with the Active property set to TRUE, shall only be hosted on the AdminDomain representing the Fabric. The Inactive Zone Sets, defined by an instance of ZoneSet with the Active property set to FALSE, SHALL be hosted on either the AdminDomain representing the Fabric as shown in Figure 6 or the ComputerSystem representing the switch as shown in Figure 7. It is allowed to have no ZoneSets (active or inactive), only an active ZoneSet, only an inactive ZoneSet(s), or both an inactive ZoneSet(s) and an active ZoneSet.

The ZoneService and ZoneCapabilities are also associated to the same System (AdminDomain or ComputerSystem) as the Inactive Zone Sets using the association HostedService or ElementCapabilities, respectively.

ZoneService provides the configuration methods to control create ZoneSets, Zones, Zone Aliases, and Zone Members, as well as activation of the Zone Set. This service and its methods are described in the Clause 5: Enhanced Zoning and Enhanced Zoning Control Subprofile.

## 4.2 Health and Fault Management

The following classes report possible Health and Fault information through LifeCycle indications:

- ComputerSystem,
- FCPort

These LifeCycle indications are more fully described in 4.8, "CIM Elements".

Also in Table 4:, "CIM Elements for Fabric" is a list of AlertIndications which may also be indicators for Health and Fault Management.

## 4.3 Cascading Considerations

None

## 4.4 Supported Subprofiles and Package

**Table 1: Supported Profiles for Fabric**

| Registered Profile Names                    | Mandatory | Version |
|---|-----------|---------|
| Zone Control                                | No        | 1.2.0   |
| Enhanced Zoning and Enhanced Zoning Control | No        | 1.2.0   |
| FDMI  | No        | 1.2.0   |
| Fabric Path Performance                     | No        | 1.2.0   |

## 4.5 Methods of this Profile

None

## 4.6 Client Considerations and Recipes

### 4.6.1 Fabric Identifier

The client needs to consider that the fabric identifier is not durable but is correlatable and may change over time. See *Storage Management Technical Specification, Part 1 Common Architecture Clause 7: Correlatable and Durable Names*.

### 4.6.2 FCPort OperationalStatus

OperationalStatus is the property to indicate status and state for the FCPort. The FCPort instance has one of the following Operational Statuses.

**Table 2: Port OperationalStatus**

| OperationalStatus | Description          |
|-------------------|----------------------|
| OK                | Port is online       |
| Error             | Port has a failure   |
| Stopped           | Port is disabled     |
| InService         | Port is in Self Test |
| Unknown           |                      |

### 4.6.3 ComputerSystem OperationalStatus

OperationalStatus is the property to indicate status and state for the ComputerSystem. The ComputerSystem instance has one of the following Operational Statuses and possibly one of the Subsidiary statuses.

**Table 3: OperationalStatus for ComputerSystem**

| Operational Status | Possible Subsidiary Operational Status | Description   |
|--------------------|--|---|
| OK                 |  | The system has a good status  |
| OK                 | Stressed                               | The system is stressed, for example the temperature is over limit or there is too much IO in progress                 |
| OK                 | Predictive Failure                     | The system will probably will fail sometime soon  |
| Degraded           |  | The system is operational but not at 100% redundancy. A component has suffered a failure or something is running slow |
| Error              |  | An error has occurred causing the system to stop. This error may be recoverable with operator intervention.           |
| Error              | Non-recoverable error                  | A severe error has occurred. Operator intervention is unlikely to fix it  |
| Error              | Supporting entity in error             | A modeled element has failed  |
| InService          |  | Switch is in Self Test.   |

**Table 3: OperationalStatus for ComputerSystem**

| Operational Status | Possible Subsidiary Operational Status | Description   |
|--------------------|--|---|
| No contact         |  | The provider knows about the array but has not talked to it since last reboot         |
| Lost communication |  | The provider used to be able to communicate with the array, but has now lost contact. |
| Starting           |  | The system is starting up   |
| Stopping           |  | The system is shutting down.  |
| Stopped            |  | The data path is OK but shut down, the management channel is still working.           |

#### 4.6.4 Discover The Fabric Topology

```

// This recipe describes how to build a topology graph of a fabric.
//
// 1. Identifies all the Switches and adds their objects paths and the
// object paths of the FC Ports belonging to these Switches to the $nodes
// array
//
// 2. Creates a suitable Association instance (e.g. a SystemDevice
// Association instance between a Switch and a FC Port), setting its
// GroupComponent and PartComponent. Adds the object path of the
// Association to the $links array
//
// 3. Creates a map of all connected FC Ports (i.e., belonging to Switches
// that are ISL'd together and to Host HBAs and Storage System Front End
// Controllers)
//
// In this map, the FC Ports (i.e., the ones that are connected) are
// cross-connected.
//
// e.g., For a pair of FC Ports, one belonging to a Switch and the other
// belonging to a Host (HBA), the map indexed by the Switch Port WWN returns
// the Host (HBA) FC Port object path and the map indexed by the Host (HBA)
// FC Port WWN returns the Switch FC Port object path.
//
// Similar relationship exists between the pairs of FC Ports where one
// belongs to a Switch and the other belonging belongs to a Storage System
// Front End Controller and for FC Ports each of which belongs to a Switch.
//
// 4. Identifies all the Hosts and adds their objects paths to the $nodes
// array. Note that the object paths of the FC Ports (HBA Ports) belonging
// to these Hosts are already added to the $nodes array in step-3.
//
// 5. Creates a suitable Association instance (e.g. a SystemDevice

```

## Fabric Profile

```
// Association instance between a Host and a FC Port), setting its
// GroupComponent and PartComponent. Adds the object path of the Association
// to the $links array.
//
// 6. Identifies all the Storage Systems and adds their objects paths to the
// $nodes array.
// Note that the object paths of the FC Ports (i.e., Front End Controller
// FC Ports) belonging to these Storage Systems are already added to the
// $nodes array in step-3.
//
// 7. Creates a suitable Association instance (e.g. a SystemDevice
// Association instance between a Storage System and a FC Port), setting
// its GroupComponent and PartComponent. Adds the object path of the
// Association to the $links array.

// DESCRIPTION
// Create a map of how elements in a SAN are connected together via
// Fibre-ChannelFC ports.
//
// The map is built in array $attachedFcPorts->[], where the index is a
// WWN of any device port on the SAN, and the value at that index is
// the object path of the connected Switch or HBA or Storage System FC port.
//
// First find all the switches in a SAN. Get all the FC Ports for each
// switch and get the Attached FC Ports for each Switch FC Port. Save these
// device FC ports in the map described above.

// PREEXISTING CONDITIONS AND ASSUMPTIONS
// 1. All agents/namespaces supporting Fabric Profile previously identified
// using SLP. Do this for each CIMOM supporting Fabric Profile

switches[] = enumerateInstances("CIM_ComputerSystem", true, false, true, true,
                               null)

for #i in $switches[]
{
    if (!contains(5, $switches[#i].Dedicated))
        continue

    // only process switches, not other computer systems

    // Add the switch to the $nodes array

    $nodes.addIfNotAlreadyAdded ($switches[#i].getObjectPath());

    // Get all the SystemDevice associations between this switch and its
    // FC Ports

    $sysDevAssoc[] = ReferenceNames($switches[#i],
```

## Fabric Profile

```
        "CIM_FCPort",
        "GroupComponent");

// Add these associations to the $links array

for #a in $sysDevAssoc->[]
$links.addIfNotAlreadyAdded ($sysDevAssoc->[#a];

$fcPorts->[] = AssociatorNames(
    $switches[#i].getObjectPath(),
    "CIM_SystemDevice",
    "CIM_FCPort",
    "GroupComponent",
    "PartComponent")
for #j in $fcPorts->[]
{

    // Add the FC Port in $nodes array

    $nodes.addIfNotAlreadyAdded (fcPorts->[#j];

    $protocolEndpoints->[] = AssociatorNames(
        fcPorts->[#j],
        "CIM_DeviceSAPImplementation",
        "CIM_ProtocolEndpoint",
        "Antecedent",
        "Dependent");

    // NOTE - It is possible for this collection to be empty (i.e., ports
    // that are not connected). It is possible for this collection to
    // have more than one element (loops attached to a switch port is the
    // most common example).

    if ($protocolEndpoints->[].length == 0)
        continue

    // Add the Protocol End Point to the nodes array.
    // Currently this recipe is designed to only save one
    // ProtocolEndpoint.

    $nodes.addIfNotAlreadyAdded (protocolEndpoints[0]);

    // Add the associations between the fcPort and the Protocol end point
    // to the links array

    $devSAPImplassoc[] = ReferenceNames($fcPorts->[#j],
        "CIM_ProtocolEndpoint",
```

## Fabric Profile

```

                                null);
for #a in $devSAPImplassoc->[]
    $links.addIfNotAlreadyAdded ($devSAPImplassoc->[#a];

$attachedProtocolEndpoints->[] = AssociatorNames(
    $protocolEndpoints->[0],
    "CIM_ActiveConnection",
    "CIM_ProtocolEndpoint",
    null, null)

// Add the Attached Protocol End Point to the nodes array

$nodes.addIfNotAlreadyAdded (attachedProtocolEndpoints->[0]);

// Add the associations between the Protocol end point and the
// Attached protocol endpoint to the links array

$actConnassoc[] = ReferenceNames($protocolEndpoint->[#0],
                                "CIM_ActiveConnection",
                                null);
for #a in $actConnassoc->[]
    $links.addIfNotAlreadyAdded ($actConnassoc->[#a];

// NOTE: role & resultRole are null as the direction of the
// association is not dictated by the specification

// $attachedFcPort is either a device FC port or an ISL'd switch FC
// port from another switch. We store this result is stored (i.e.
// which device FC Port is connected // to which switch FC Port) in
// a suitable data structure for subsequent correlation to ports
// discovered on devices.

for #k in $attachedProtocolEndpoints->[]
{
    $attachedFcPorts->[] = Associators(
        $attachedProtocolEndpoints->[#k],
        "CIM_DeviceSAPImplementation",
        "CIM_FCPort",
        "Dependent",
        "Antecedent",
        false,
        false,
        ["PermanentAddress"])
    $attachedFcPort = $attachedFcPorts[0] // Exactly one member guaranteed
                                        by model

    // Add the attached FC Port to the $nodes array
    if $attachedFcPort != null

```

## Fabric Profile

```
        $nodes.addIfNotAlreadyAdded ($attachedFcPort);
    }
}

Determine the active Zone Set in a SAN
// DESCRIPTION
// Traverse from the fabric to all zone sets, looking for
// the active zone set
//
// PREEXISTING CONDITIONS AND ASSUMPTIONS
//
// 1. The fabric of interest (an AdminDomain) has been previously
//    identified and defined in the $Fabric-> variable

$ZoneSets[] = Associators($Fabric->, "CIM_HostedCollection", "CIM_ZoneSet", null,
    null, false, false, null)

for #i in $ZoneSets[] {
    if ($ZoneSet[#i].Active) {
        // <found active ZoneSet>
        // NOTE - there can be only one active ZoneSet in a fabric, though there
        //          may be none
        break
    }
}
```

## 4.7 Registered Name and Version

Fabric version 1.2.0



## 4.8 CIM Elements

**Table 4: CIM Elements for Fabric**

| Element Name  | Requirement | Description  |
|---|-------------|--|
| CIM_ActiveConnection (4.8.1)  | Mandatory   | The association between ProtocolEndpoints representing the links between devices.  |
| CIM_AdminDomain (SAN) (4.8.2)                                       | Mandatory   | AdminDomain representing the SAN.  |
| CIM_AdminDomain (Fabric) (4.8.3)                                    | Mandatory   | AdminDomain representing the Fabric.   |
| CIM_Component (4.8.4)   | Mandatory   | Aggregates Hosts, Arrays and Switches in the AdminDomain that represents the Fabric  |
| CIM_ComputerSystem (Switch) (4.8.5)                                 | Mandatory   | The ComputerSystem representing the Switch.  |
| CIM_ComputerSystem (Platform) (4.8.6)                               | Mandatory   | The ComputerSystem representing the Platform (e.g. Host and Array).  |
| CIM_ConnectivityCollection (4.8.7)                                  | Mandatory   | Collects the ProtocolEndpoints of the fabric.  |
| CIM_ContainedDomain (4.8.8)   | Mandatory   | Associates a Fabric to a SAN   |
| CIM_DeviceSAPImplementation (Non-Switch to FCPort) (4.8.9)          | Mandatory   | Associates the FCPort to the ProtocolEndpoint  |
| CIM_DeviceSAPImplementation (Switch to FCPort) (4.8.10)             | Mandatory   | Associates the FCPort to the ProtocolEndpoint  |
| CIM_ElementCapabilities (ZoneCapabilities to fabric.) (4.8.11)      | Mandatory   | Associates ZoneCapabilities to a System  |
| CIM_ElementCapabilities (ZoneCapabilities to switch.) (4.8.12)      | Mandatory   | Associates ZoneCapabilities to a switch system   |
| CIM_FCPort (Switch FCPort) (4.8.13)                                 | Mandatory   | Fibre Channel Port for Switch  |
| CIM_FCPort (Non-Switch FCPort) (4.8.14)                             | Mandatory   | Fibre Channel Port for Devices   |
| CIM_HostedAccessPoint (ComputerSystem to ProtocolEndpoint) (4.8.15) | Mandatory   | Associates the ProtocolEndpoint to the hosting ComputerSystem  |
| CIM_HostedAccessPoint (AdminDomain to ProtocolEndpoint) (4.8.16)    | Optional    | Associates the ProtocolEndpoint to AdminDomain.  |
| CIM_HostedCollection (Fabric to ConnectivityCollection) (4.8.17)    | Mandatory   | Associates the ConnectivityCollection to the AdminDomain representing the Fabric.  |
| CIM_HostedCollection (Zones or ZoneSets to Fabric) (4.8.18)         | Mandatory   | Associates the ZoneSets and Zones to the AdminDomain representing the Fabric.  |
| CIM_HostedCollection (Zones or ZoneSets to Switch) (4.8.19)         | Mandatory   | Associates the ZoneSets, Zones, and NamedAddressCollections representing the ZoneAliases to the hosting System (the ComputerSystem representing the switch). |
| CIM_LogicalPortGroup (4.8.20)                                       | Mandatory   | Fibre Channel Node   |

**Table 4: CIM Elements for Fabric**

| Element Name  | Requirement | Description  |
|---|-------------|--|
| CIM_HostedCollection (ComputerSystem to LogicalPortGroup) (4.8.21)  | Mandatory   | Associates the LogicalPortGroup to the ComputerSystem representing the platform (host or array). |
| CIM_MemberOfCollection (ConnectivityCollection to ProtocolEndpoint) (4.8.22)  | Mandatory   | Associates ConnectivityCollection to ProtocolEndpoint  |
| CIM_MemberOfCollection (LogicalPortGroup to FCPort) (4.8.23)  | Mandatory   |  |
| CIM_MemberOfCollection (ZoneSet to Zone) (4.8.24)   | Mandatory   |  |
| CIM_ProtocolEndpoint (4.8.25)   | Mandatory   | The endpoint of a link (ActiveConnection).   |
| CIM_Zone (Active) (4.8.26)  | Mandatory   | The active Zones being enforced by the Fabric.   |
| CIM_Zone (Inactive) (4.8.27)  | Mandatory   | The inactive Zones being enforced by the Fabric.   |
| CIM_ZoneCapabilities (4.8.28)   | Mandatory   | The Zoning Capabilities of the associated Fabric (or Switch).                                    |
| CIM_ZoneMembershipSettingData (4.8.29)  | Mandatory   | Defines the zone member  |
| CIM_ZoneSet (Active) (4.8.30)   | Mandatory   | The active ZoneSets being enforced by the Fabric.  |
| CIM_ZoneSet (Inactive) (4.8.31)   | Mandatory   | The inactive ZoneSet   |
| SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_ComputerSystem  | Mandatory   | Creation of a ComputerSystem instance.   |
| SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_ComputerSystem  | Mandatory   | Deletion of a ComputerSystem instance.   |
| SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_FCPort  | Mandatory   | Creation of a FC Port instance   |
| SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_FCPort  | Mandatory   | Deletion of a FC Port instance   |
| SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_FCPort AND SourceInstance.OperationalStatus <> PreviousInstance.OperationalStatus                         | Mandatory   | Deprecated WQL - Modification of OperationalStatus of a FC Port instance                         |
| SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_FCPort AND SourceInstance.CIM_FCPort::OperationalStatus <> PreviousInstance.CIM_FCPort::OperationalStatus | Optional    | Experimental CQL - Modification of OperationalStatus of a FC Port instance                       |

**Table 4: CIM Elements for Fabric**

| Element Name  | Requirement | Description   |
|---|-------------|---|
| SELECT * FROM CIM_InstModification<br>WHERE SourceInstance ISA<br>CIM_ComputerSystem AND<br>SourceInstance.OperationalStatus <><br>PreviousInstance.OperationalStatus   | Mandatory   | Deprecated WQL - Modification of<br>OperationalStatus of a ComputerSystem<br>instance   |
| SELECT * FROM CIM_InstModification<br>WHERE SourceInstance ISA<br>CIM_ComputerSystem AND<br>SourceInstance.CIM_ComputerSystem::Oper<br>ationalStatus <><br>PreviousInstance.CIM_ComputerSystem::Op<br>erationalstatus | Optional    | Experimental CQL - Modification of<br>OperationalStatus of a ComputerSystem<br>instance |
| SELECT * FROM CIM_AlertIndication<br>WHERE OwningEntity='SNIA' and<br>MessageID='FC1'   | Optional    | Experimental CQL - Modification of Zone<br>Database                                     |
| SELECT * FROM CIM_AlertIndication<br>WHERE OwningEntity=SNIA and<br>MessageID='FC1'   | Mandatory   | Deprecated WQL - Modification of Zone<br>Database                                       |
| SELECT * FROM CIM_AlertIndication<br>WHERE OwningEntity='SNIA' and<br>MessageID='FC2'   | Optional    | Experimental CQL - ZoneSet Activated  |
| SELECT * FROM CIM_AlertIndication<br>WHERE OwningEntity=SNIA and<br>MessageID='FC2'   | Mandatory   | Deprecated WQL - ZoneSet Activated  |

**4.8.1 CIM\_ActiveConnection**

The association between ProtocolEndpoints representing the links between devices (including ISLs). For loops, multiple ActiveConnections are instantiated as one to many relationships.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 5 describes class CIM\_ActiveConnection.

**Table 5: SMI Referenced Properties/Methods for CIM\_ActiveConnection**

| Properties | Flags | Requirement | Description & Notes   |
|------------|-------|-------------|---|
| Antecedent |       | Mandatory   | The reference to the ProtocolEndpoint for one end of the link       |
| Dependent  |       | Mandatory   | The reference to the ProtocolEndpoint for the other end of the link |

#### 4.8.2 CIM\_AdminDomain (SAN)

AdminDomain representing the SAN.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 6 describes class CIM\_AdminDomain (SAN).

**Table 6: SMI Referenced Properties/Methods for CIM\_AdminDomain (SAN)**

| Properties        | Flags | Requirement | Description & Notes                          |
|-------------------|-------|-------------|--|
| CreationClassName |       | Mandatory   | Name of Class                                |
| Name              |       | Mandatory   | An arbitrary name (implementation dependent) |
| NameFormat        |       | Mandatory   | Dependent on the arbitrary name chosen.      |

#### 4.8.3 CIM\_AdminDomain (Fabric)

AdminDomain representing the fabric. This is a logical entity and can represent virtual fabrics.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 7 describes class CIM\_AdminDomain (Fabric).

**Table 7: SMI Referenced Properties/Methods for CIM\_AdminDomain (Fabric)**

| Properties        | Flags | Requirement | Description & Notes |
|-------------------|-------|-------------|---------------------|
| CreationClassName |       | Mandatory   | Name of Class       |
| Name              | C     | Mandatory   | WWN of Fabric       |
| NameFormat        |       | Mandatory   | Shall be 'WWN'.     |

#### 4.8.4 CIM\_Component

Aggregates Hosts, Arrays and Switches in the AdminDomain that represents the Fabric

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 8 describes class CIM\_Component.

**Table 8: SMI Referenced Properties/Methods for CIM\_Component**

| Properties     | Flags | Requirement | Description & Notes                                  |
|----------------|-------|-------------|--|
| PartComponent  |       | Mandatory   | Reference to Switch                                  |
| GroupComponent |       | Mandatory   | Reference to the AdminDomain representing the Fabric |

#### 4.8.5 CIM\_ComputerSystem (Switch)

The ComputerSystem representing the Switch.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 9 describes class CIM\_ComputerSystem (Switch).

**Table 9: SMI Referenced Properties/Methods for CIM\_ComputerSystem (Switch)**

| Properties              | Flags | Requirement | Description & Notes   |
|-------------------------|-------|-------------|---|
| CreationClassName       |       | Mandatory   | Name of Class   |
| Name                    | C     | Mandatory   | The Switch WWN.   |
| ElementName             |       | Mandatory   | The Switch Symbolic Name.   |
| NameFormat              |       | Mandatory   | Shall be 'WWN'.   |
| OperationalStatus       |       | Mandatory   | One of the defined values (2 3 6 8 9 10 11 12 13) shall be present in the array value.  |
| OtherIdentifyingInfo    |       | Mandatory   | DomainID stored in decimal format   |
| Dedicated               |       | Mandatory   | Shall be 5 (Switch)   |
| IdentifyingDescriptions |       | Mandatory   | Identifying descriptor for OtherIdentifyingInfo. The value 'DomainID' is in IdentifyingDescriptions and in the corresponding index for OtherIdentifyingInfo the DomainID is placed. |

#### 4.8.6 CIM\_ComputerSystem (Platform)

The ComputerSystem representing the Platform (e.g. Host and Array). This class is typically instantiated if the end device has populated the Fibre Channel Platform Database or FDMI.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 10 describes class CIM\_ComputerSystem (Platform).

**Table 10: SMI Referenced Properties/Methods for CIM\_ComputerSystem (Platform)**

| Properties        | Flags | Requirement | Description & Notes  |
|-------------------|-------|-------------|--|
| CreationClassName |       | Mandatory   | Name of Class  |
| Name              |       | Mandatory   | The Platform Name or FDMI Host Name.   |
| ElementName       |       | Mandatory   | The Platform Label.  |
| NameFormat        |       | Mandatory   |  |
| Dedicated         |       | Mandatory   | For a FC-GS Platform Type of Host, 'Not Dedicated' (0); for storage subsystems, 'Storage' (3); for Gateway, 'Gateway' (20); for Router, 'Router' (4); for Bridge, 'Bridge/Extender' (19); for Platform Type of Other, 'Other' (2). |

#### 4.8.7 CIM\_ConnectivityCollection

Collects the ProtocolEndpoints of the fabric.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 11 describes class CIM\_ConnectivityCollection.

**Table 11: SMI Referenced Properties/Methods for CIM\_ConnectivityCollection**

| Properties  | Flags | Requirement | Description & Notes                  |
|-------------|-------|-------------|--------------------------------------|
| InstanceID  |       | Mandatory   | Opaque                               |
| ElementName |       | Optional    | Not required, can be the Fabric WWN. |

#### 4.8.8 CIM\_ContainedDomain

Associates one or more Fabrics to a SAN.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 12 describes class CIM\_ContainedDomain.

**Table 12: SMI Referenced Properties/Methods for CIM\_ContainedDomain**

| Properties     | Flags | Requirement | Description & Notes                                      |
|----------------|-------|-------------|--|
| PartComponent  |       | Mandatory   | The reference to the AdminDomain representing the Fabric |
| GroupComponent |       | Mandatory   | The reference to the AdminDomain representing the SAN    |

#### 4.8.9 CIM\_DeviceSAPImplementation (Non-Switch to FCPort)

Associates the FCPort to the ProtocolEndpoint

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 13 describes class CIM\_DeviceSAPImplementation (Non-Switch to FCPort).

**Table 13: SMI Referenced Properties/Methods for CIM\_DeviceSAPImplementation (Non-Switch to FCPort)**

| Properties | Flags | Requirement | Description & Notes               |
|------------|-------|-------------|-----------------------------------|
| Dependent  |       | Mandatory   | Reference to the ProtocolEndpoint |
| Antecedent |       | Mandatory   | Reference to the FCPort           |

#### 4.8.10 CIM\_DeviceSAPImplementation (Switch to FCPort)

Associates the FCPort to the ProtocolEndpoint

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 14 describes class CIM\_DeviceSAPImplementation (Switch to FCPort).

**Table 14: SMI Referenced Properties/Methods for CIM\_DeviceSAPImplementation (Switch to FCPort)**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Dependent  |       | Mandatory   |                     |
| Antecedent |       | Mandatory   |                     |

**4.8.11 CIM\_ElementCapabilities (ZoneCapabilities to fabric.)**

Associates the ZoneCapabilities to a System. The system is the AdminDomain representing the Fabric.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 15 describes class CIM\_ElementCapabilities (ZoneCapabilities to fabric.).

**Table 15: SMI Referenced Properties/Methods for CIM\_ElementCapabilities (ZoneCapabilities to fabric.)**

| Properties     | Flags | Requirement | Description & Notes           |
|----------------|-------|-------------|-------------------------------|
| Capabilities   |       | Mandatory   | Reference to ZoneCapabilities |
| ManagedElement |       | Mandatory   | Reference to AdminDomian      |

**4.8.12 CIM\_ElementCapabilities (ZoneCapabilities to switch.)**

Associates the ZoneCapabilities to a System. The system normally is the AdminDomain representing the Fabric, but in some cases where the Zone Database is not a fabric entity, it maybe hosted on a ComputerSystem representing the Switch.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 16 describes class CIM\_ElementCapabilities (ZoneCapabilities to switch.).

**Table 16: SMI Referenced Properties/Methods for CIM\_ElementCapabilities (ZoneCapabilities to switch.)**

| Properties     | Flags | Requirement | Description & Notes           |
|----------------|-------|-------------|-------------------------------|
| Capabilities   |       | Mandatory   | Reference to ZoneCapabilities |
| ManagedElement |       | Mandatory   | Reference to ComputerSystem   |

**4.8.13 CIM\_FCPort (Switch FCPort)**

Fibre Channel Port for Switch

Created By: Static

Modified By: Static

Deleted By: Static



Class Mandatory: Mandatory

Table 17 describes class CIM\_FCPort (Switch FCPort).

**Table 17: SMI Referenced Properties/Methods for CIM\_FCPort (Switch FCPort)**

| Properties              | Flags | Requirement | Description & Notes   |
|-------------------------|-------|-------------|---|
| SystemCreationClassName |       | Mandatory   | The scoping System's CreationClassName.   |
| SystemName              |       | Mandatory   | The scoping System's Name.  |
| CreationClassName       |       | Mandatory   | Name of Class   |
| DeviceID                |       | Mandatory   | Opaque  |
| ElementName             |       | Mandatory   | Port Symbolic Name if available. Otherwise NULL. If the underlying implementation includes characters that are illegal in CIM strings, then truncate before the first of those characters.  |
| PermanentAddress        | CD    | Mandatory   | Fibre Channel Port WWN  |
| Speed                   |       | Optional    | Speed of zero represents a link not established.<br>1Gb is 1 062 500 000 bps<br>2Gb is 2 125 000 000 bps<br>4Gb is 4 250 000 000 bps<br>10Gb single channel variants are 10 518 750 000 bps<br>10Gb four channel variants are 12 750 000 000 bps<br>This is the raw bit rate. |
| OperationalStatus       |       | Mandatory   | One of the defined values shall be present in the array value.  |
| PortType                |       | Mandatory   | The specific port type currently enabled (from FC-GS Port.Type)   |
| LinkTechnology          |       | Mandatory   | 'FC'  |

#### 4.8.14 CIM\_FCPort (Non-Switch FCPort)

Fibre Channel Port for non-Switches (Storage, Router, Bridge/Extender)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 18 describes class CIM\_FCPort (Non-Switch FCPort).

**Table 18: SMI Referenced Properties/Methods for CIM\_FCPort (Non-Switch FCPort)**

| Properties              | Flags | Requirement | Description & Notes  |
|-------------------------|-------|-------------|--|
| SystemCreationClassName |       | Mandatory   | The scoping System's CreationClassName.  |
| SystemName              |       | Mandatory   | The scoping System's Name.   |
| CreationClassName       |       | Mandatory   | Name of Class  |
| DeviceID                |       | Mandatory   | Opaque   |
| ElementName             |       | Optional    | Port Symbolic Name if available. Otherwise NULL. If the underlying implementation includes characters that are illegal in CIM strings, then truncate before the first of those characters.   |
| PermanentAddress        | CD    | Optional    | Fibre Channel Port WWN. Expressed as 16 un-separated upper case hex digits (see Table 4 for more information about formats).   |
| NetworkAddresses        | C     | Optional    | Fibre Channel ID (FCID). Expressed as 8 un-separated upper case hex digits (see Table 4 for more information about formats).   |
| OperationalStatus       |       | Mandatory   | One of the defined values shall be present in the array value.   |
| PortType                |       | Mandatory   | The specific port type currently enabled (from FC-GS Port.Type)  |
| LinkTechnology          |       | Mandatory   | Shall be 4 ('FC').   |
| SupportedFC4Types       |       | Optional    | An array of integers indicating the Fibre Channel FC-4 protocols supported.<br><br><b>EXPERIMENTAL</b>   When supporting FC-SB, this property shall be either '27' (FC-SB-2 Channel) for Host Ports or '28' (FC-SB-2 Control Unit) for targets.   <b>EXPERIMENTAL</b>        |
| SupportedCOS            |       | Optional    | An array of integers indicating the Fibre Channel Classes of Service that are supported.<br><br><b>EXPERIMENTAL</b>   When supporting FC-SB, this property shall be either '2' or '3' for FC-SB-2 Channel or FC-SB-2 Control Unit ports, respectively.   <b>EXPERIMENTAL</b> |

#### 4.8.15 CIM\_HostedAccessPoint (ComputerSystem to ProtocolEndpoint)

Associates the ProtocolEndpoint to the hosting System. The hosting System is a ComputerSystem for the Switch or Platform.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 19 describes class CIM\_HostedAccessPoint (ComputerSystem to ProtocolEndpoint).

**Table 19: SMI Referenced Properties/Methods for CIM\_HostedAccessPoint (ComputerSystem to ProtocolEndpoint)**

| Properties | Flags | Requirement | Description & Notes               |
|------------|-------|-------------|-----------------------------------|
| Dependent  |       | Mandatory   | Reference to the ProtocolEndpoint |
| Antecedent |       | Mandatory   | Reference to the ComputerSystem   |

#### 4.8.16 CIM\_HostedAccessPoint (AdminDomain to ProtocolEndpoint)

Associates the ProtocolEndpoint to the AdminDomain for those systems not registered in the Platform Database or discovered through FDMI.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Optional

Table 20 describes class CIM\_HostedAccessPoint (AdminDomain to ProtocolEndpoint).

**Table 20: SMI Referenced Properties/Methods for CIM\_HostedAccessPoint (AdminDomain to ProtocolEndpoint)**

| Properties | Flags | Requirement | Description & Notes               |
|------------|-------|-------------|-----------------------------------|
| Dependent  |       | Mandatory   | Reference to the ProtocolEndpoint |
| Antecedent |       | Mandatory   | Reference to the AdminDomain      |

#### 4.8.17 CIM\_HostedCollection (Fabric to ConnectivityCollection)

Associates the ConnectivityCollection to the AdminDomain representing the Fabric.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 21 describes class CIM\_HostedCollection (Fabric to ConnectivityCollection).

**Table 21: SMI Referenced Properties/Methods for CIM\_HostedCollection (Fabric to ConnectivityCollection)**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Dependent  |       | Mandatory   |                     |
| Antecedent |       | Mandatory   |                     |

#### 4.8.18 CIM\_HostedCollection (Zones or ZoneSets to Fabric)

Associates the ZoneSets and Zones to the AdminDomain representing the Fabric.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 22 describes class CIM\_HostedCollection (Zones or ZoneSets to Fabric).

**Table 22: SMI Referenced Properties/Methods for CIM\_HostedCollection (Zones or ZoneSets to Fabric)**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Dependent  |       | Mandatory   |                     |
| Antecedent |       | Mandatory   |                     |

#### 4.8.19 CIM\_HostedCollection (Zones or ZoneSets to Switch)

Associates the ZoneSets, Zones, and NamedAddressCollections representing the ZoneAliases to the hosting System (the ComputerSystem representing the switch).

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 23 describes class CIM\_HostedCollection (Zones or ZoneSets to Switch).

**Table 23: SMI Referenced Properties/Methods for CIM\_HostedCollection (Zones or ZoneSets to Switch)**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Dependent  |       | Mandatory   |                     |
| Antecedent |       | Mandatory   |                     |

#### 4.8.20 CIM\_LogicalPortGroup

Represents the Fibre Channel Node. Associated to the host system by the HostedCollection Association. The hosting System is either a ComputerSystem representing the Platform or the AdminDomain representing the fabric in the case for those systems not registered in the Platform Database or discovered through FDMI (but available through the Name Server/Management Server).

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 24 describes class CIM\_LogicalPortGroup.

**Table 24: SMI Referenced Properties/Methods for CIM\_LogicalPortGroup**

| Properties  | Flags | Requirement | Description & Notes  |
|-------------|-------|-------------|--|
| InstanceID  |       | Mandatory   | Opaque   |
| Name        | CD    | Mandatory   | Fibre Channel Node WWN   |
| NameFormat  |       | Mandatory   | Shall be 'WWN'.  |
| ElementName | N     | Mandatory   | Node Symbolic Name if available. Otherwise NULL. If the underlying implementation includes characters that are illegal in CIM strings, then truncate before the first of those characters. |

#### 4.8.21 CIM\_HostedCollection (ComputerSystem to LogicalPortGroup)

Associates the LogicalPortGroup to the ComputerSystem representing the platform (host or array).

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 25 describes class CIM\_HostedCollection (ComputerSystem to LogicalPortGroup).

**Table 25: SMI Referenced Properties/Methods for CIM\_HostedCollection (ComputerSystem to LogicalPortGroup)**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Dependent  |       | Mandatory   |                     |
| Antecedent |       | Mandatory   |                     |

#### 4.8.22 CIM\_MemberOfCollection (ConnectivityCollection to ProtocolEndpoint)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 26 describes class CIM\_MemberOfCollection (ConnectivityCollection to ProtocolEndpoint).

**Table 26: SMI Referenced Properties/Methods for CIM\_MemberOfCollection (ConnectivityCollection to ProtocolEndpoint)**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Collection |       | Mandatory   |                     |
| Member     |       | Mandatory   |                     |

#### 4.8.23 CIM\_MemberOfCollection (LogicalPortGroup to FCPort)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 27 describes class CIM\_MemberOfCollection (LogicalPortGroup to FCPort).

**Table 27: SMI Referenced Properties/Methods for CIM\_MemberOfCollection (LogicalPortGroup to FCPort)**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Collection |       | Mandatory   |                     |
| Member     |       | Mandatory   |                     |

#### 4.8.24 CIM\_MemberOfCollection (ZoneSet to Zone)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 28 describes class CIM\_MemberOfCollection (ZoneSet to Zone).

**Table 28: SMI Referenced Properties/Methods for CIM\_MemberOfCollection (ZoneSet to Zone)**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Collection |       | Mandatory   |                     |
| Member     |       | Mandatory   |                     |

#### 4.8.25 CIM\_ProtocolEndpoint

The endpoint of a link (ActiveConnection). ProtocolEndpoint shall be implemented when an ActiveConnection exists. It may be implemented if no ActiveConnections exist.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 29 describes class CIM\_ProtocolEndpoint.

**Table 29: SMI Referenced Properties/Methods for CIM\_ProtocolEndpoint**

| Properties              | Flags | Requirement | Description & Notes                         |
|-------------------------|-------|-------------|---|
| SystemCreationClassName |       | Mandatory   | The scoping System's CreationClassName.     |
| SystemName              |       | Mandatory   | The scoping System's Name.                  |
| CreationClassName       |       | Mandatory   | Name of Class                               |
| Name                    | CD    | Mandatory   | The Fibre Channel Port WWN.                 |
| NameFormat              |       | Mandatory   | 'WWN'                                       |
| ProtocolIFType          |       | Mandatory   | Shall be 56(Fibre channel). 'Fibre Channel' |

#### 4.8.26 CIM\_Zone (Active)

The active Zones being enforced by the Fabric.

Created By: Extrinsic: ActivateZoneSet

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 30 describes class CIM\_Zone (Active).

**Table 30: SMI Referenced Properties/Methods for CIM\_Zone (Active)**

| Properties  | Flags | Requirement | Description & Notes                                   |
|-------------|-------|-------------|---|
| InstanceID  |       | Mandatory   | Opaque  |
| ElementName |       | Mandatory   | The Zone Name   |
| ZoneType    |       | Mandatory   | The Zone Type   |
| Active      |       | Mandatory   | Shall be TRUE. Indicates that this ZoneSet is active. |

#### 4.8.27 CIM\_Zone (Inactive)

The inactive Zones being enforced by the Fabric.

Created By: Extrinsic: CreateZone

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 31 describes class CIM\_Zone (Inactive).

**Table 31: SMI Referenced Properties/Methods for CIM\_Zone (Inactive)**

| Properties  | Flags | Requirement | Description & Notes                                      |
|-------------|-------|-------------|--|
| InstanceID  |       | Mandatory   | Opaque   |
| ElementName |       | Mandatory   | The Zone Name  |
| ZoneType    |       | Mandatory   | The Zone Type.   |
| Active      |       | Mandatory   | Shall be FALSE. Indicates that this ZoneSet is inactive. |

#### 4.8.28 CIM\_ZoneCapabilities

The Zoning Capabilities of the associated Fabric (or Switch).

ZoneCapabilities exposes the capabilities of the AdminDomain representing the Fabric for active zoning and the capabilities of the ComputerSystem representing the Switch or AdminDomain representing the Fabric for Zone Set Database.

If a ZoneCapability property is not applicable or does not explicitly exists (e.g. the capability is limited only by a memory size), the property is NULL.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory



Table 32 describes class CIM\_ZoneCapabilities.

**Table 32: SMI Referenced Properties/Methods for CIM\_ZoneCapabilities**

| Properties                       | Flags | Requirement | Description & Notes  |
|----------------------------------|-------|-------------|--|
| InstanceID                       |       | Mandatory   | Opaque   |
| MaxNumZoneSets                   |       | Optional    | The maximum number of ZoneSets in the Zone Set Database.<br><br>NULL should be returned in such cases when the property is not applicable or the number is not limited explicitly.   |
| MaxNumZone                       |       | Optional    | The maximum number of Zones in the Zone Set Database.<br><br>NULL should be returned in such cases when the property is not applicable or the number is not limited explicitly.  |
| MaxNumZoneMembers                |       | Optional    | The maximum number of ZoneMembers in the Zone Set Database . All ZoneMembers included in both Zones and ZoneAliases are counted, while the same ZoneMember included in multiple Zones or ZoneAliases is counted only once.<br><br>NULL should be returned in such cases when the property is not applicable or the number is not limited explicitly. |
| MaxNumZoneAliases                |       | Optional    | The maximum number of ZoneAliases in the Zone Set Database<br><br>NULL should be returned in such cases when the property is not applicable or the number is not limited explicitly.   |
| ZoneNameMaxLen                   |       | Mandatory   | The maximum length for the name of a ZoneAlias (NamedAddressCollection.ElementName), Zone (Zone.ElementName) or ZoneSet (ZoneSet.ElementName) the Fabric (or Switch) are capable of supporting.  |
| ZoneNameFormat                   |       | Mandatory   | The name format of a ZoneAlias *NamedAddressCollection.ElementName), Zone (Zone.ElementName) or ZoneSet (ZoneSet.ElementName) supported by either the Fabric (or the Switch)   |
| MaxNumZonesPerZoneSet            |       | Optional    | The maximum number of Zones per ZoneSet.<br><br>NULL should be returned in such cases when the property is not applicable or the number is not limited explicitly.   |
| SupportedConnectivityMemberTypes |       | Mandatory   | An array containing the supported connectivity member types supported which include Permanent Address (WWN), Switch Port ID (Domain:Port in base10), Network Address (FCID), Logical Port Group (Node WWN).  |

#### 4.8.29 CIM\_ZoneMembershipSettingData

Defines the zone member

Created By: Extrinsic: AddZoneMemberSettingData

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 33 describes class CIM\_ZoneMembershipSettingData.

**Table 33: SMI Referenced Properties/Methods for CIM\_ZoneMembershipSettingData**

| Properties             | Flags | Requirement | Description & Notes  |
|------------------------|-------|-------------|--|
| InstanceID             |       | Mandatory   | Opaque   |
| ConnectivityMemberType |       | Mandatory   | Permanent Address (WWN), Switch Port ID (Domain:Port in base10), Network Address (FCID). |
| ConnectivityMemberID   | C     | Mandatory   | The value of the WWN, Domain/Port, or FCID.  |

#### 4.8.30 CIM\_ZoneSet (Active)

The active ZoneSet being enforced by the Fabric.

Created By: Extrinsic: ActivateZoneSet

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 34 describes class CIM\_ZoneSet (Active).

**Table 34: SMI Referenced Properties/Methods for CIM\_ZoneSet (Active)**

| Properties  | Flags | Requirement | Description & Notes   |
|-------------|-------|-------------|---|
| InstanceID  |       | Mandatory   | Opaque  |
| ElementName |       | Mandatory   | The ZoneSet name.   |
| Active      |       | Mandatory   | shall be TRUE. Indicates that this ZoneSet is active and members cannot be changed. |

#### 4.8.31 CIM\_ZoneSet (Inactive)

The inactive ZoneSets.

Created By: Extrinsic: CreateZoneSet

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 35 describes class CIM\_ZoneSet (Inactive).

**Table 35: SMI Referenced Properties/Methods for CIM\_ZoneSet (Inactive)**

| Properties  | Flags | Requirement | Description & Notes                                      |
|-------------|-------|-------------|--|
| InstanceID  |       | Mandatory   | Opaque   |
| ElementName |       | Mandatory   | The ZoneSet name.  |
| Active      |       | Mandatory   | Shall be FALSE. Indicates that this ZoneSet is inactive. |

**STABLE**

---



---

---

**STABLE****Clause 5: Enhanced Zoning and Enhanced Zoning Control Subprofile****5.1 Description**

This profile describes the additional zoning functions for enhanced zoning. Note that Sessions are normally part of enhanced zoning, but are included in the base fabric profile to address the various types of zoning operations into a single object model. In this subprofile, then only Zone Alias is added.

**5.2 Health and Fault Management**

None

**5.3 Cascading Considerations**

None

**5.4 Dependencies on Profiles, Subprofiles, and Packages**

Support for the Clause 6: Zone Control Subprofile is mandatory for the Enhanced Zoning and Enhanced Zoning Control subprofile.

**5.5 Methods of this Profile****CreateZoneAlias**

The method creates a ZoneAlias and the association HostedCollection. The newly created association, HostedCollection, associates the ZoneAlias to the same AdminDomain the ZoneService is hosted to. For the newly created ZoneAlias, the Active property is always set to false.

```
CreateZoneAlias(
    [IN] string CollectionAlias,
    [OUT] CIM_NamedAddressCollection ref ZoneAlias);
```

**AddZoneAlias**

Adds to the Zone the specified ZoneAlias.

```
AddZoneAlias(
    [IN] CIM_Zone ref Zone,
    [IN] CIM_NamedAddressCollection ref ZoneAlias);
```

**5.6 Client Considerations and Recipes****5.6.1 Create a ZoneAlias**

```
// DESCRIPTION
// Create zone alias and add new zone member based on
// the parameters collected by the CIM Client.
// Before any operations can be imposed on the zoning
// service, a session is requested and obtained from the zone
// service. Create a new ZoneAlias. The session may not be ended if
// the ZoneAlias is empty, so add a zone member to the new ZoneAlias.
```

```

// The session is released when the operations are
// completed.
//
// PRE-EXISTING CONDITIONS AND ASSUMPTION
// 1.The system of interest,either the fabric (AdminDomain)
//   or the switch (ComputerSystem), has been
//   previously identified and defined in the
//   $System-> variable
// 2.The name of the new zone alias is defined in the
//   #ZoneAliasName variable
// 3.   The zone member type is defined in the #ConnectivityMemberType
//   variable
// 4.   The zone member Id of the new zone member is defined in the
//   #ConnectiivityMemberID variable

// 1. Get the ZoneService and start a session
$ZoneServices->[] = AssociatorNames(
    $System->,
    "CIM_HostedService",
    "CIM_ZoneService", null, null)

// Assumption 1 above guarantees there is a zone service for this
// system. the fabric and switch profiles that there is no more than
// one ZoneService for this system
$ZoneService-> = $ZoneServices[0]

if(!&startSession($ZoneService->))
{
    return
}

// 2. Create the ZoneAlias
%InArguments["CollectionAlias"] = #ZoneAliasName
#status = InvokeMethod(
    $ZoneService->,
    "CreateZoneAlias",
    %InArguments[],
    %OutArguments[])

$ZoneAlias-> = %OutArguments["ZoneAlias"]
if(#status != 0)
    // ERROR!

// 3. Create or locate a ZoneMembershipSettingData
%InArguments["ConnectivityMemberType"] = #ConnectivityMemberType
%InArguments["ConnectivityMemberID"] = #ConnectivityMemberID

```

```

%InArguments["SystemSpecificCollection"] = $ZoneAlias->
#status = InvokeMethod($ZoneService->, "CreateZoneMembershipSettingData",
                        %InArguments[], %OutArguments[])

// 4. Add to zone alias if not created as a member of the zone alias
//     Zone member reference is set accordingly in the output arguments.

$ZoneMember-> = %OutArguments["ZoneMembershipSettingData"]

if (#status != 0)
    // ERROR!

// 5. End the session gracefully
&endSession($ZoneService->)
// 6. Verify that the ZoneAlias exists in the database
try{
    GetInstance($ZoneAlias->)
}catch(CIM_ERR_NOT_FOUND){
    // error
}

```

### 5.6.2 Delete a ZoneAlias

```

// DESCRIPTION
// Delete a zone alias.
// Before any operations can be imposed on the zoning service, a
// session is requested and obtained from the zone service.
// The session is released when the operations are completed.
//
// if the deletion fails, it may be because the Zone Alias is not empty.
// In this case, remove all members from the alias by deleting the
// ElementSettingData associations, and try the deletion again.
//
// PRE-EXISTING CONDITIONS AND ASSUMPTION
// 1.The system of interest,either the fabric (AdminDomain)
//    or the switch (ComputerSystem), has been
//    previously identified and defined in the
//    $System-> variable
// 2.The object name of the zone alias to be deleted is
//    defined in the $ZoneAlias-> variable

// 1. Get the zone service and start a session
$ZoneServices->[] = AssociatorNames(
    $System->,
    "CIM_HostedService",
    "CIM_ZoneService",

```

```

    null,
    null)

// Assumption 1 above guarantees there is a zone service for this
// system. the fabric and switch profiles that there is no more than
// one ZoneService for this system
$ZoneService-> = $ZoneServices[0]

if(!&startSession($ZoneService->))
{
    return
}

// 2. Attempt to delete the alias
try{
    DeleteInstance($ZoneAlias->)
}catch(CIM_ERR_FAILED){
    // Try to remove any zone members in the alias
    // via the ElementSettingData association
    $ZoneMembers->[] = referenceNames($ZoneAlias->,
        "CIM_ElementSettingData",
        null)
    for #j in $ZoneMembers->[] {
        DeleteInstance($ZoneMembers[#j])
    }
    // Try again
    DeleteInstance($ZoneAlias->)
}

// 3. End Session
&endSession($ZoneService->)
// verify that the deletion occurred
try{
    GetInstance($ZoneAlias->)
}catch(CIM_ERR_NOT_FOUND){
    //expect exception
    return
}
// error!!

```

## 5.7 Registered Name and Version

Enhanced Zoning and Enhanced Zoning Control version 1.2.0



## 5.8 CIM Elements

**Table 36: CIM Elements for Enhanced Zoning and Enhanced Zoning Control**

| Element Name                       | Requirement | Description  |
|------------------------------------|-------------|--|
| CIM_HostedCollection (5.8.1)       | Mandatory   | Associates the NameAddressCollection representing the Zone Alias to the AdminDomain            |
| CIM_HostedCollection (5.8.2)       | Mandatory   | Associates the NameAddressCollection representing the Zone Alias to the System                 |
| CIM_ElementSettingData (5.8.3)     | Optional    | Associates ZoneMembershipSettingData to the NamedAddressCollection representing the ZoneAlias. |
| CIM_MemberOfCollection (5.8.4)     | Mandatory   | Associates NamedAddressCollection with Zone.   |
| CIM_NamedAddressCollection (5.8.5) | Mandatory   | The Zone Alias.  |
| CIM_ZoneService (5.8.6)            | Mandatory   | The service that allows for all of the zoning configuration changes.                           |

### 5.8.1 CIM\_HostedCollection

Associates the NamedAddressCollection representing the Zone Alias to the AdminDomain representing the Fabric.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 37 describes class CIM\_HostedCollection.

**Table 37: SMI Referenced Properties/Methods for CIM\_HostedCollection**

| Properties | Flags | Requirement | Description & Notes   |
|------------|-------|-------------|---|
| Antecedent |       | Mandatory   |   |
| Dependent  |       | Mandatory   | Reference to the NamedAddressCollection representing the Zone Alias |

### 5.8.2 CIM\_HostedCollection

Associates the NamedAddressCollection representing the Zone Alias to the ComputerSystem representing the switch.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 38 describes class CIM\_HostedCollection.

**Table 38: SMI Referenced Properties/Methods for CIM\_HostedCollection**

| Properties | Flags | Requirement | Description & Notes   |
|------------|-------|-------------|---|
| Antecedent |       | Mandatory   |   |
| Dependent  |       | Mandatory   | Reference to the NamedAddressCollection representing the Zone Alias |

### 5.8.3 CIM\_ElementSettingData

Associates ZoneMembershipSettingData to the NamedAddressCollection representing the ZoneAlias.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Optional

Table 39 describes class CIM\_ElementSettingData.

**Table 39: SMI Referenced Properties/Methods for CIM\_ElementSettingData**

| Properties     | Flags | Requirement | Description & Notes |
|----------------|-------|-------------|---------------------|
| SettingData    |       | Mandatory   |                     |
| ManagedElement |       | Mandatory   |                     |

### 5.8.4 CIM\_MemberOfCollection

Associates NamedAddressCollection with Zone.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 40 describes class CIM\_MemberOfCollection.

**Table 40: SMI Referenced Properties/Methods for CIM\_MemberOfCollection**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Collection |       | Mandatory   |                     |
| Member     |       | Mandatory   |                     |

### 5.8.5 CIM\_NamedAddressCollection

The Zone Alias.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 41 describes class CIM\_NamedAddressCollection.

**Table 41: SMI Referenced Properties/Methods for CIM\_NamedAddressCollection**

| Properties      | Flags | Requirement | Description & Notes |
|-----------------|-------|-------------|---------------------|
| InstanceID      |       | Mandatory   | Opaque              |
| CollectionAlias |       | Mandatory   | The Zone Alias Name |

### 5.8.6 CIM\_ZoneService

The service that allows for all of the zoning configuration changes. The definition in this profile adds additional methods to the same service defined in Zone Control.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 42 describes class CIM\_ZoneService.

**Table 42: SMI Referenced Properties/Methods for CIM\_ZoneService**

| Properties                  | Flags | Requirement | Description & Notes                     |
|-----------------------------|-------|-------------|---|
| SystemCreationClass<br>Name |       | Mandatory   | The scoping System's CreationClassName. |
| SystemName                  |       | Mandatory   | The scoping System's Name.              |
| CreationClassName           |       | Mandatory   | The Class Name                          |
| Name                        |       | Mandatory   | Opaque                                  |
| CreateZoneAlias()           |       | Mandatory   |   |
| AddZoneAlias()              |       | Mandatory   |   |

## **STABLE**

---

---

---

**STABLE****Clause 6: Zone Control Subprofile****6.1 Description**

This profile includes extrinsic methods for creating Zone Sets, Zones, and Zone Members and adding Zones to Zone Sets and Zone Members to Zones. Additionally SMI-S defines intrinsic methods for the removing of Zone Members from Zones and Zone Aliases, Zones from Zone Sets, and deleting Zone Members, Zones, and Zone Sets. The profile also includes methods to allow a client to request or release a lock on the fabric for zoning configuration changes.

When an Inactive ZoneSet is “Activated”, new instances representing the Active Zone Set and Active Zones are generated from the Inactive Zone Set definition (where a switch may prune the referenced Zone Set collapsing aliases, removes empty zones, etc.).

When a new Zone Set is “Activated”, the instances representing the previous active Zone Set no longer exists.

This profile also includes the property ConfigurationID on the ZoneService. The ConfigurationID is used to identify a unique point in time identifier of the zoning configuration. It may be a timestamp, a generation identifier, or a checksum. The ConfigurationID shall change whenever the zoning configuration changes. The value of the property is used as an input to ActivateZoneSet() and SessionControl() to allow the client to confirm that the change requested are consistent with the last discovery the client made of the zoning configuration.

In the case where the Inactive Zone Sets are hosted on a switch, the client cannot know which Inactive Zone Set was used to define the current Active Zone Set. Also if two Inactive Zone Sets with the same name are hosted on two different switches, the definitions maybe completely different.

**6.2 Durable Names and Correlatable IDs of the Profile**

None

**6.3 Instrumentation Requirements**

The agent shall support the use case defined in 6.8.

**6.4 Health and Fault Management**

None

**6.5 Cascading Considerations**

None

**6.6 Dependencies on Profiles, Subprofiles, and Packages**

None

**6.7 Methods of this Profile****6.7.1 CreateZoneSet**

The method creates a ZoneSet and associates it to the System (AdminDomain representing the Fabric or the ComputerSystem representing the Switch) that the ZoneService is hosted on.

```
CreateZoneSet (
    [IN] string ZoneSetName,
    [OUT] CIM_ZoneSet ref ZoneSet);
```

### 6.7.2 CreateZone

The method creates a Zone and associates it to System (AdminDomain representing the Fabric or the ComputerSystem representing the Switch) that the ZoneService is hosted on.

```
CreateZone (
    [IN] string ZoneName,
    [IN] uint16 ZoneType,
    [IN] uint16 ZoneSubType,
    [OUT] CIM_Zone ref Zone);
```

### 6.7.3 CreateZoneMembershipSettingData

The method creates a ZoneMembershipSettingData (a zone member) and adds it to the specified Zone or NamedAddressCollection representing a Zone Alias. The ConnectivityMemberID is dependent upon the ConnectivityMemberType.

For Fibre Channel, the ConnectivityMemberType of "PermanentAddress", the ConnectivityMemberID is the NxPort WWN; for ConnectivityMemberType of "NetworkAddress", the ConnectivityMemberID is the NXPort Address ID; for ConnectivityMemberType of "SwitchPortID", the ConnectivityMemberID is "Domain:PortNumber".

```
CreateZoneMembershipSettingData (
    [IN] uint16 ConnectivityMemberType,
    [IN] string ConnectivityMemberID,
    [IN] CIM_SystemSpecificCollection ref SystemSpecificCollection,
    [OUT] CIM_ZoneMembershipSettingData ref ZoneMembershipSettingData);
```

### 6.7.4 AddZone

The method adds to the specified ZoneSet the specified Zone. Adding a Zone to a ZoneSet, extends the zone enforcement definition of the ZoneSet to include the members of that Zone. If adding the Zone is, successful, the Zone should be associated to the ZoneSet by MemberOfCollection.

```
AddZone (
    [IN] CIM_ZoneSet ref ZoneSet,
    [IN] CIM_Zone ref Zone);
```

### 6.7.5 AddZoneMembershipSettingData

The method adds to the specified Zone or NamedAddressCollection representing the Zone Alias the specified ZoneMembershipSettingData (a zone member).

```
AddZoneMembershipSettingData (
    [IN] CIM_SystemSpecificCollection ref SystemSpecificCollection,
    [IN] CIM_ZoneMembershipSettingData ref ZoneMembershipSettingData);
```

---

---

## EXPERIMENTAL

### 6.7.6 ActivateZoneSetWithJob

This method activates the specified ZoneSet with the expectation that a job will be created. It has an optional input, ConfigurationID, which is used to confirm that the ZoneSet being activated matches the definition the application previously discovered or changed. If the ConfigurationID passed in matches the ConfigurationID associated to the current zone definitions, then the fabric attempts to activate the ZoneSet. If the ConfigurationIDs do not match, the method exits without activating the ZoneSet. Once a ZoneSet is activated, a ZoneSet with the property Active set to true, its associated Zones with the property Active set to true, and the Zone's associated ZoneMembershipSettingData are instantiated.

This method may run asynchronously due to the length of the operation. This typically occurs when the session is ended and the changes are committed. When the method is run asynchronously, the value "Method parameters checked - job started" is returned and the output parameter Job is populated with a reference to the job instance.

ActivateZoneSet shall be supported outside of a session. ActivateZoneSet being called within a session is implementation specific.

Calling ActivateZoneSet outside of a session while a session is open is implementation specific.

```
uint32 ActivateZoneSetWithJob (
    [IN] CIM_ZoneSet ref ZoneSet,
    [IN] boolean Activate,
    [IN,OUT] string ConfigurationID,
    [OUT] CIM_ConcreteJob REF Job,
    [IN] datetime TimeoutPeriod)
```

---

---

## EXPERIMENTAL

### 6.7.7 ActivateZoneSet

This method activates the specified ZoneSet without a Job. ActivateZoneSetWithJob replaces this method. It is supported for backwards compatibility and should not be used in future implementations.

Once a ZoneSet is activated, a ZoneSet with the property Active set to true, its associated Zones with the property Active set to true, and the Zone's associated ZoneMembershipSettingData are instantiated.

ActivateZoneSet shall be supported outside of a session. ActivateZoneSet being called within a session is implementation specific.

Calling ActivateZoneSet outside of a session while a session is open is implementation specific.

```
uint32 ActivateZoneSet (
    [IN] CIM_ZoneSet ref ZoneSet,
    [IN] boolean Activate )
```

---

---

## EXPERIMENTAL

### 6.7.8 SessionControlWithJob

This method allows a client to request or release a lock on the fabric for zoning configuration changes. The method has an optional input, ConfigurationID, which is used to confirm that the ZoneSet being activated matches the definition the application previously discovered or changed. If the ConfigurationID passed in matches the ConfigurationID associated to the current zone definitions, then the fabric attempts to grant the lock. If the ConfigurationIDs do not match, the method exits without attempting to have the fabric grant the lock.

As described in FC-GS, in the context of Enhanced Zoning Management, management actions to a Zone Server (e.g. write access to the Zoning Database) shall occur only inside a GS session. Clients executing zoning management operations shall use fabric sessions cooperatively if the SMI-S agent supports it. (If the value of SessionState is 4 ("Not Applicable") then no cooperative session usage is possible).

Before a client executes zoning management operations (intrinsic or extrinsic methods), the client shall request a new session and wait for the request to be granted. To request a new session, first wait until the property "SessionState" of the fabric's ZoneService is 3 ("Ended") and the property "RequestedSessionState" is 5 "No Change". Then call SessionControl with RequestedSessionState = 2 ("Started"). Once zoning management operations are completed, the client shall release the session to enable the provider to propagate changes to the fabric, and to allow other clients to perform management operations. To end a session and commit the changes, call SessionControl with RequestedSessionState = 3 ("Ended"). To abort a sequence of zoning management operations without updating the fabric, call SessionControl with RequestedSessionState = 4 ("Terminated").

SMI-S agents shall block on calls to SessionControl until the request is fulfilled. For example, an error may occur while committing changes to a fabric, i.e. after a call to SessionControl with RequestedSessionState = 3 ("Ended"). The method cannot return until the session has ended, so that a CIM error can be returned if a problem occurs. While the method is in progress, another client may read the value of the RequestedSessionState property and see the value set by the method currently in progress. Once the request is fulfilled, the RequestedSessionState property is set to value 5 "No Change", regardless of the value in the setInstance operation.

Sessions can timeout. The session timeout behavior and settings are defined by FC-SW in the section discussing mapping GS sessions for Enhanced Zoning Management.

This method may run asynchronously due to the length of the operation. This shall occur when the session is "Ended" to commit the changes. It shall not be used for any other case. When the method is run asynchronously, the value "Method parameters checked - job started" is returned and the output parameter Job is populated with a reference to the job instance.

A SMIS agent may raise an error if these client cooperation rules are not followed. For the purposes of a SMIS agent, a series of requests from the same authenticated entity are considered to be from a single client. An agent may verify that such a series corresponds to the sequence described above and raise the error CIM\_ERR\_FAILED at any time if the sequence is violated.

```
uint32 SessionControlWithJob (
    [IN,
        ValueMap {"2", "3", "4"},
        Values {"Started", "Ended", "Terminated"}]
    uint16 RequestedSessionState,
    [IN,OUT] string ConfigurationID,
    [OUT] CIM_ConcreteJob REF Job,
    [IN] datetime TimeoutPeriod)
```

## EXPERIMENTAL

---

### 6.7.9 SessionControl

The method enables a client to request a lock of the fabric to begin zoning configuration changes. It is supported for legacy implementations. SessionControlWithJob replaces this method. It is supported for backwards compatibility only and should not be used in future implementations.

This method allows a client to request or release a lock on the fabric for zoning configuration changes. As described in FC-GS, in the context of Enhanced Zoning Management, management actions to a Zone Server (e.g. write access to the Zoning Database) shall occur only inside a GS session. Clients executing zoning management operations shall use fabric sessions cooperatively if the SMI-S agent supports it. (If the value of SessionState is 4 ("Not Applicable") then no cooperative session usage is possible).



Before a client executes zoning management operations (intrinsic or extrinsic methods), the client shall request a new session and wait for the request to be granted. To request a new session, first wait until the property "SessionState" of the fabric's ZoneService is 3 ("Ended") and the property "RequestedSessionState" is 5 "No Change". Then call SessionControl with RequestedSessionState = 2 ("Started"). Once zoning management operations are completed, the client shall release the session to enable the provider to propagate changes to the fabric, and to allow other clients to perform management operations. To end a session and commit the changes, call SessionControl with RequestedSessionState = 3 ("Ended"). To abort a sequence of zoning management operations without updating the fabric, call SessionControl with RequestedSessionState = 4 ("Terminated").

SMIS agents shall block on calls to SessionControl until the request is fulfilled. For example, an error may occur while committing changes to a fabric, i.e. after a call to SessionControl with RequestedSessionState = 3 ("Ended"). The method cannot return until the session has ended, so that a CIM error can be returned if a problem occurs. While the method is in progress, another client may read the value of the RequestedSessionState property and see the value set by the method currently in progress. Once the request is fulfilled, the RequestedSessionState property is set to value 5 "No Change", regardless of the value in the setInstance operation.

Sessions can timeout. The session timeout behavior and settings are defined by INCITS FC-SW in the section discussing mapping GS sessions for Enhanced Zoning Management.

A SMIS agent may raise an error if these client cooperation rules are not followed. For the purposes of a SMIS agent, a series of requests from the same authenticated entity are considered to be from a single client. An agent may verify that such a series corresponds to the sequence described above and raise the error CIM\_ERR\_FAILED at any time if the sequence is violated.

```
uint32 SessionControl (
    [IN,
    ValueMap {"2", "3", "4"},
    Values {"Started", "Ended", "Terminated"}]
    uint16 RequestedSessionState);
```

#### 6.7.10 Intrinsic for removing a zone from a zone set

As seen in the instance diagram, a zone is a member of a zone set if there is a "CIM\_MemberOfCollection" association from the zone set to the zone. To remove a zone from a zone set, delete the instance of the association "CIM\_MemberOfCollection" using the intrinsic operation deleteInstance.

#### 6.7.11 Intrinsic for removing a zone alias from a zone

A zone alias is a member of a zone if there is a "CIM\_MemberOfCollection" association from the zone to the zone alias. To remove a zone alias from a zone set, delete the instance of the association "CIM\_MemberOfCollection" using the intrinsic operation deleteInstance.

#### 6.7.12 Intrinsic for removing a zone member from a zone or zone alias

Zone members are represented by CIM\_ZoneMembershipSettingData instances. No instance of CIM\_ZoneMembershipSettingData exists unless it is associated to a zone or zone alias by a CIM\_ElementSettingData association. However, an instance of CIM\_ZoneMembershipSettingData may be associated to more than one zone or zone alias.

Removing a zone member from a zone or zone alias is equivalent to deleting the instance of the CIM\_ElementSettingData association. Delete the instance using the intrinsic operation deleteInstance.

If this is the last instance of a CIM\_ElementSettingData association for a particular CIM\_ZoneMembershipSettingData, do not delete the instance of CIM\_ZoneMembershipSettingData; it is the provider's responsibility to clean up these structures.

### 6.7.13 Intrinsic for deleting a zone member

Zone members are represented by CIM\_ZoneMembershipSettingData instances associated to zones or zone aliases via CIM\_ElementSettingData associations. To delete a zone member (and remove it from any zones or zone aliases from which it is a member) use the CIM operation deleteInstance to delete the instance of CIM\_ZoneMembershipSettingData.

Do not delete the corresponding instances of the CIM\_ElementSettingData; it is the provider's responsibility to clean up these structures.

### 6.7.14 Intrinsic for deleting a zone, zone alias, or zone set

Use the intrinsic operation deleteInstance to delete a zone, zone alias or zone set. Client are allowed to delete zones or zone aliases that are members of collections (zones or zone sets). Clients are allowed to delete the last member of a zone or zone set, leaving the collection empty.

A zone set or zone cannot be deleted if it is currently active (the error would be CIM\_ERR\_FAILED). Some implementations may prohibit deleting zonesets, zones or zone aliases that still have members (the error would be CIM\_ERR\_FAILED). When a zone, zone alias or zone set is deleted, the client does not have to delete the corresponding instances of CIM\_MemberOfCollection or CIM\_HostedCollection; it is the provider's responsibility to clean up these structures.

## 6.8 Client Considerations and Recipes

Many agent implementations do not allow Zone, a ZoneAlias or a Zone Set to be defined empty. Since the methods defined in SMI-S do not support creating a Zone Set with a Zone and a Zone with a Zone Member, the SessionControl method should be used to build a Zone Definition that is interoperable. This is done by calling ZoneSession() to “Start” defining or updating the Zone Definition. The client then calls the appropriate methods as necessary to build the desired Zone Definition. For example, calling CreateZoneSet() to create a new Zone Set, CreateZone() to create a new Zone, AddZoneToZoneSet() to add the newly created Zone to the newly created Zone Set, and CreateZoneMembershipSettingData() to create and add a new Zone Member to the newly created Zone. Upon completion of the new zoning definition, ZoneControl is called again to “End” the session. The changes to the Zone Definition would then be applied to the Zone Set Database. This set of calls would create a Zone Definition where the Zone and ZoneSet are not empty and would be interoperable across all agent implementations.

### 6.8.1 Create or delete zones Common Functions

```
// DESCRIPTION
//
// Common functions used by the recipes below.
//
// startSession: attempt to start fabric session if required;
//   returns false if attempt fails; returns true if attempt succeeds
//   or if session control is unnecessary
//
// endSession: finalize fabric session if required; returns false
//   if attempt fails; returns true if attempt succeeds or if session
//   control is unnecessary
//
//
// findActiveZoneSet: routine to find the active zoneset
// on a fabric, and return the reference to it
//
```

## Zone Control Subprofile

```
// PREEXISTING CONDITIONS AND ASSUMPTIONS
//
// None

sub boolean startSession ($ZoneService->)
{
    $ZoneService = GetInstance($ZoneService->, false, false, false, null)

    // session statuses
    #Ended = 3
    #NotApplicable = 4

    // requested session statuses
    #Started = 2
    #NoChange = 5

    if ($ZoneService.SessionState == #NotApplicable)
        return true // no session control implemented by this agent

    if ($ZoneService.SessionState != #Ended)
        return false // fabric session is in use by another client or agent

    if ($ZoneService.RequestedSessionState != #NoChange)
        return false // another client has already requested session

    %InArguments["RequestedSessionState"] = #Started

    #status = InvokeMethod($ZoneService->, "SessionControl", %InArguments,
                           %OutArguments)
    if (#status != 0) // e.g. "Failed"
        return false

    $ZoneService = GetInstance($ZoneService->, false, false, false, null)
    if ($ZoneService.SessionState != #Started)
        return false

    return true
}

// PREEXISTING CONDITIONS AND ASSUMPTIONS
//
// None

sub boolean endSession ($ZoneService->) {
    $ZoneService = GetInstance($ZoneService->, false, false, false, null)
```

## Zone Control Subprofile

```
// session statuses
#Started = 2
#NotApplicable = 4

// requested session statuses
#End = 3

if ($ZoneService.SessionStatus == #NotApplicable){
    return true        // no need for session control

if ($ZoneService.SessionStatus != #Started)
    return false       // no session started by this client

%InArguments["RequestedSessionState"] = #End
#status = InvokeMethod($ZoneService, "SessionControl", %InArguments,
                        %OutArguments)
if (#status != 0)      // e.g. "Failed"
    return false

// Do not wait, or even check, for SessionState to have value "Ended" as
// a) InvokeMethod will block till done (or failed) anyway
// b) Before the check can be made, session may already be started
//    by another client

return true
}

// PREEXISTING CONDITIONS AND ASSUMPTIONS
// The reference to the fabric on which the active
// zoneset it to be sought is already known in
// the input variable $Fabric. Calling code
// should verify that the returned reference is non-null
//
sub Ref findActiveZoneSet($Fabric->){
    $ActiveZoneSet->=null
    $ZoneSets[] = Associators(
        $Fabric->,
        "CIM_HostedCollection",
        "CIM_ZoneSet",
        null,
        null,
        false,
        false,
        {"Active"} )
    // there may be no active zoneset
    if(0 < ZoneSets[].size()){
        for(#i in $ZoneSets[]){
```

```

        if(true==$ZoneSets[#i].Active){
            $ActiveZoneSet->=nameof($ZoneSets[#i])
            break
        }
    }
}
return $ActiveZoneSet->
}

```

### 6.8.2 Add new Zone Member to Existing Zone

```

// DESCRIPTION
// Add new Zone Member to Existing Zone
//
// Assume the client has already invoked some logic to determine which
// System (fabric or switch) will host the zone database and zone
// service to be used. Request and obtain a fabric session from the
// zone service. Use an extrinsic method to attempt to create a new
// instance of ZoneMembershipSettingData, associated to a zone. If
// the creation fails because an instance already exists for the
// desired zone member id, simply create an association between the
// pre-existing ZoneMembershipSettingData instance and the zone
// instance. Then close the fabric session.
//
// PREEXISTING CONDITIONS AND ASSUMPTIONS
//
// 1. The System hosting the zone database (ComputerSystem or
//     AdminDomain) has been previously identified and defined in the
//     $System-> variable
//
// 2. The zone member type is defined in the #ConnectivityMemberType variable
//
// 3. The zone member id of the new zone member is defined in the
//     #ConnectivityMemberID variable
//
// 4. An existing zone is defined in the $Zone-> variable
//
// FUNCTIONS

// 1. Get the Zone Service and start the session

$ZoneServices->[] = AssociatorNames($System->, "CIM_HostedService",
                                   "CIM_ZoneService", null, null)

// Assumption 1 (above) guarantees there is a zone service for this
// System, Fabric Profile mandates there is no more than one zone
// service for this System

```

## Zone Control Subprofile

```
$ZoneService-> = $ZoneService->[0]

// Start the session
if (!&startSession($ZoneService->)) {
    <ERROR! Failed to start zone session>
}

// 2. Create a ZoneMembershipSettingData
%InArguments["ConnectivityMemberType"] = #ConnectivityMemberType
%InArguments["ConnectivityMemberID"] = #ConnectivityMemberID
%InArguments["SystemSpecificCollection"] = $Zone->
#status = InvokeMethod($ZoneService->, "CreateZoneMembershipSettingData",
    %InArguments[], %OutArguments[])

if (#status != 0){
    <ERROR! call to method CreateZoneMembershipSettingData failed #status>
}

// 3. Store the returned object path for verification
$ZoneMember-> = %OutArguments["ZoneMembershipSettingData"]

// 4. End session successfully
if(!&endSession($ZoneService->)){
    <ERROR! Failed to end session, changes may not have been committed>
}

// 5. Verify that the zonemember exist within the specified zone

$ZoneMembers->[]=associatorNames(
    $Zone->,
    "CIM_ElementSettingData",
    "CIM_ZoneMembershipSettingData",
    "ManagedElement",
    "SettingData" )
if(!contains($ZoneMember->,$ZoneMembers[])){
    <ERROR! Failed to verify zone member created>
}
```

### 6.8.3 Create new Zone, add new Zone Member, and add to existing ZoneSet

```
// DESCRIPTION
// Create new Zone, add new Zone Member, and add to existing ZoneSet
//
// Assume the client has already invoked some logic to determine which
// System (fabric or switch) will host the zone database and zone
// service to be used. Request and obtain a fabric session from the
// zone service. Create a new Zone using an extrinsic method. The
// session may not be ended if any zone is empty, so add a zone member
```

## Zone Control Subprofile

```
// to the new zone. The session also may not be ended unless every
// zone is a member of at least one zone set, so add the new zone to
// an existing zone set. Then close the fabric session.
//
//
// PREEXISTING CONDITIONS AND ASSUMPTIONS
//
// 1. The System hosting the zone database (ComputerSystem or
//     AdminDomain) has been previously identified and defined in the
//     $System-> variable
//
// 2. The name for a new zone is defined in the #ZoneName variable
//
// 3. The type for the new zone is defined in the #ZoneType variable
//
// 4. The sub type for the new zone is defined in the #ZoneSubType
//     variable
//
// 5. The zone member type is defined in the #ConnectivityMemberType variable
//
// 6. The zone member id of the new zone member is defined in the
//     #ConnectivityMemberID variable
//
// 7. An existing zoneSet is defined in the $ZoneSet-> variable
//
// FUNCTIONS

// 1. Get the Zone Service and start the session
$ZoneServices->[] = AssociatorNames($System->, "CIM_HostedService",
                                   "CIM_ZoneService", null, null)

// Assumption 1 (above) guarantees there is a zone service for this
// System, Fabric Profile mandates there is no more than one zone
// service for this System
$ZoneService-> = $ZoneServices->[0]

    if (!&startSession($ZoneService->)) {
        <ERROR! Failed to start zone session>
    }

// 2. Create a zone
%InArguments["ZoneName"] = #ZoneName
%InArguments["ZoneType"] = #ZoneType
%InArguments["ZoneSubType"] = #ZoneSubType
InvokeMethod($ZoneService->, "CreateZone", %InArguments[], %OutArguments[])
$Zone-> = $OutArguments["Zone"]
```

## Zone Control Subprofile

```
// 3. Create a ZoneMembershipSettingData
%InArguments["ConnectivityMemberType"] = #ConnectivityMemberType
%InArguments["ConnectivityMemberID"] = #ConnectivityMemberID
%InArguments["SystemSpecificCollection"] = $Zone->
#status = InvokeMethod($ZoneService->, "CreateZoneMembershipSettingData",
                        %InArguments[], %OutArguments[])

if (#status != 0){
    <ERROR! Call to method CreateZoneMembershipSettingData failed #status>
}
// 4. Save the returned member objectpath for verification
$ZoneMember-> = %OutArguments["ZoneMembershipSettingData"]

// 5. Add the new zone to the existing zone set
%InArguments["ZoneSet"] = $ZoneSet->
%InArguments["Zone"] = $Zone->
#status = InvokeMethod($ZoneService->, "AddZone", %InArguments[], %OutArguments[])
if (#status != 0){
    <ERROR Call to method AddZone failed>
}
// 6. End Session
if(!&endSession($ZoneService->)){
    <ERROR! Failed to end session, changes may not have been committed>
}
// 7. Verify that the zone exists in the zone set
$Zones->[]=associatorNames(
    $ZoneSet->,
    "CIM_MemberOfCollection",
    "CIM_Zone",
    "Collection",
    "Member"
)
// see if the zone is in the returned array
if(!contains($Zone->,$Zones->[])){
    <ERROR! Failed to verify that Zone was added to ZoneSet>
}
```

### 6.8.4 Create new ZoneSet and add existing Zone

```
// DESCRIPTION
// Create new ZoneSet and add existing Zone
//
// Assume the client has already invoked some logic to determine which
// System (fabric or switch) will host the zone database and zone
```



## Zone Control Subprofile

```
// service to be used. Request and obtain a fabric session from the
// zone service. Create a new ZoneSet with a given name, using an
// extrinsic method. The session may not be ended if any ZoneSet is
// empty, so add an existing zone to the ZoneSet. Then close the
// fabric session.
//
// PREEXISTING CONDITIONS AND ASSUMPTIONS
//
// 1. The System hosting the zone database (ComputerSystem or
//     AdminDomain) has been previously identified and defined in the
//     $System-> variable
//
// 2. The name for the new zone set is defined in the #ZoneSetName
//     variable
//
// 3. An existing zone is defined in the $Zone-> variable
//
// FUNCTIONS

// 1. Get the Zone Service and start the session

$ZoneServices->[] = AssociatorNames($System->, "CIM_HostedService",
                                   "CIM_ZoneService", null, null)
// Assumption 1 (above) guarantees there is a zone service for this
// System, Fabric Profile mandates there is no more than one zone
// service for this System
$ZoneService-> = $ZoneServices->[0]

if (!&startSession($ZoneService->)){
    <ERROR! Failed to start zone session>
}

// 2. Create a zone set
$InArguments["ZoneSetName"] = #ZoneSetName
#status = InvokeMethod($ZoneService->, "CreateZoneSet", $InArguments[],
                      %OutArguments[])
if (#status != 0){
    <ERROR! Call to method CreateZoneSet failed>
}

$ZoneSet-> = %OutArguments["ZoneSet"]

// 3. Add the existing zone to the new zone set
$InArguments["ZoneSet"] = $ZoneSet->
$InArguments["Zone"] = $Zone->
#status = InvokeMethod($ZoneService->, "AddZone", $InArguments[], %OutArguments[])
if (#status != 0){
    <ERROR! Call to method AddZone failed #status>
}
```

```

}

// 4. End Session
if(!&endSession($ZoneService->)){
    <ERROR! Failed to end zone session, changes may not be committed>
}

// 5. Verify that the new zone set exists in the zone database
try{
    GetInstance($ZoneSet->);
}catch(CIM_ERR_NOT_FOUND){
    <ERROR! Failed to verify ZoneSet created>
}

```

### 6.8.5 Delete zone

```

// DESCRIPTION
// Delete Zone
//
// Try to use intrinsic delete operation to delete a Zone instance.
// Before any operations can be imposed on the zoning service, a
// session is requested and obtained from the zone service. If the
// deletion fails, this may be because the zone is active, or because
// it is not empty. In the latter case, remove all members from the
// zone by deleting the ElementSettingData association instances, and
// try the deletion again.
//
// PRE-EXISTING CONDITIONS AND ASSUMPTION
// 1. The object name of the zone to be deleted is defined in the
//     $Zone-> variable
// 2. The object name of the zone service object for the System
//     hosting the zone database is defined in the $ZoneService->
//     variable

if(!&startSession($ZoneService->)){
    <ERROR! Failed to start session>
}

try {
    DeleteInstance($Zone->)
}
catch(CIM_ERR_FAILED) {
    // Verify that Zone is not active
    $Zone = GetInstance($Zone->, false, false, false, null)
    if ($Zone.Active) {
        // tell client of its logic problem

```

```

        <ERROR! May not delete Zone from active ZoneSet>
    }

    // Failure may be caused because zone has members
    // Try to delete all zone memberships (not zone members themselves)
    $ZoneElements->[] = ReferenceNames($Zone->, "CIM_ElementSettingData", null)
    for #i in $ZoneElements->[] {
        DeleteInstance($ZoneElements[#i])
    }

    // Try again
    DeleteInstance($Zone->)
}
if(!&endSession($ZoneService->)){
    <ERROR! Failed to end session, changes may not be committed>
}
// Verify that the zone no longer exists in the zone database
try{
    GetInstance($Zone->)
}catch(CIM_ERR_NOT_FOUND){
    // expect failure
    return
}
// error if no exception thrown
<ERROR! Found Zone that should have been deleted>

```

### 6.8.6 Delete ZoneSet

```

// DESCRIPTION
// Delete Zone Set
//
// Try to use intrinsic delete operation to delete a ZoneSet
// instance. Before any operations can be imposed on the zoning
// service, a session is requested and obtained from the zone service.
// The session is released when the operations are complete. If the
// deletion fails, this may be because the zone set is active, or
// because it is not empty. In the latter case, remove all zones from
// the zone set by deleting the MemberOfCollection association
// instances, and try the deletion again.
//
// PRE-EXISTING CONDITIONS AND ASSUMPTION
// 1. The object name of the zone set to be deleted is defined in the
//     $ZoneSet-> variable
// 2. The object name of the zone service object for the system
//     hosting the zone database is defined in the $ZoneService->
//     variable

```

```

if (!&startSession($ZoneService->))
    <ERROR! Failed to start session>
}

try {
    DeleteInstance($ZoneSet->)
}
catch(CIM_ERR_FAILED) {
    $ZoneSet = GetInstance($ZoneSet->, false, false, false, null)
    if ($ZoneSet.Active) {
        // tell client of logic problem
        <ERROR! May not delete an active ZoneSet>
    }

    // Failure may be because zoneset is not empty
    $ZoneMemberships->[] = ReferenceNames($ZoneSet->, "CIM_MemberOfCollection",
        null)
    for #i in $ZoneMemberships->[] {
        DeleteInstance($ZoneMemberships->[$i])
    }

    // Try again
    DeleteInstance($ZoneSet->)
}
if(!&endSession($ZoneService->)){
    <ERROR! Failed to end session, changes may not have been committed>
}
// Verify that the deletion did indeed occur
try{
    GetInstance($ZoneSet->)
}catch(CIM_ERR_NOT_FOUND){
    // expected, not a recipe error
    return
}
// error if no exception caught
<ERROR! Found ZoneSet that should have been deleted>

```

### 6.8.7 Delete ZoneMember

```

// DESCRIPTION
// Delete a zone member, removing it from any zones and aliases of
// which it is a member.
//
// Use the intrinsic delete operation to delete a
// ZoneMembershipSettingData instance. Before any operations can be

```

```

// imposed on the zoning service, a session is requested and obtained
// from the zone service. The session is released when the operations
// are complete.
//
// PRE-EXISTING CONDITIONS AND ASSUMPTION
// 1. The object name of the ZoneMembershipSettingData to be deleted is defined in
//     the
//     $ZoneMember-> variable
// 2. The object name of the zone service object for the system
//     hosting the zone database is defined in the $ZoneService->
//     variable

if(!&startSession($ZoneService->)){
    <ERROR! Failed to start session>
}

DeleteInstance($ZoneMember->)
if(!&endSession($ZoneService->)){
    <ERROR! Failed to end session, changes may not have been committed>
}
// verify that it is indeed deleted
try{
    GetInstance($ZoneMember->)
}catch(CIM_ERR_NOT_FOUND){
    // expect an exception,
    // not a recipe error
    return
}
// error if no exception caught
<ERROR! Found ZoneMember that should have been deleted>

```

## 6.9 Registered Name and Version

Zone Control version 1.2.0

## 6.10 CIM Elements

**Table 43: CIM Elements for Zone Control**

| Element Name               | Requirement | Description  |
|----------------------------|-------------|--|
| SNIA_ZoneService (6.10.1)  | Mandatory   | The service that allows for all of the zoning configuration changes.   |
| CIM_HostedService (6.10.2) | Mandatory   | Associates the ZoneService to the AdminDomain representing the fabric or the ComputerSystem representing the switch. |

### 6.10.1 SNIA\_ZoneService

The service that allows for all of the zoning configuration changes.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 44 describes class SNIA\_ZoneService.

**Table 44: SMI Referenced Properties/Methods for SNIA\_ZoneService**

| Properties              | Flags | Requirement | Description & Notes   |
|-------------------------|-------|-------------|---|
| SystemCreationClassName |       | Mandatory   | The scoping System's CreationClassName.   |
| SystemName              |       | Mandatory   | The scoping System's Name.  |
| CreationClassName       |       | Mandatory   | The Class Name  |
| Name                    |       | Mandatory   | Opaque  |
| OperationalStatus       |       | Mandatory   | Status of Zoning Service.   |
| SessionState            |       | Mandatory   | State of session. Valid values are "Starting", "Ended".   |
| RequestedSessionState   |       | Mandatory   | The requested session state from the client. The valid values that can be set are "Start", "End", and "Terminate".  |
| DefaultZoningState      |       | Mandatory   |   |
| CreateZoneSet()         |       | Mandatory   | The method creates a ZoneSet and associates it to the System (AdminDomain representing the Fabric or the ComputerSystem representing the Switch) that the ZoneService is hosted on. |
| CreateZone()            |       | Mandatory   | The method creates a Zone and associates it to System (AdminDomain representing the Fabric or the ComputerSystem representing the Switch) that the ZoneService is hosted on.        |

**Table 44: SMI Referenced Properties/Methods for SNIA\_ZoneService**

| Properties                            | Flags | Requirement | Description & Notes   |
|---------------------------------------|-------|-------------|---|
| CreateZoneMembers<br>hipSettingData() |       | Mandatory   | The method creates a ZoneMembershipSettingData (a zone member) and adds it to the specified Zone or NamedAddressCollection representing a Fibre Channel Node.   |
| AddZone()                             |       | Mandatory   | The method adds to the specified ZoneSet the specified Zone.  |
| AddZoneMembership<br>SettingData()    |       | Mandatory   | The method adds to the specified Zone or NamedAddressCollection representing the Fibre Channel Node the specified ZoneMembershipSettingData (a zone member).  |
| ActivateZoneSet()                     |       | Mandatory   | The method activates the specified ZoneSet.   |
| ActivateZoneSetWith<br>Job()          |       | Mandatory   | The method activates the specified ZoneSet. The method has an optional input, ConfigurationID, which is used to confirm that the ZoneSet being activated matches the definition the application previously discovered or changed. This method may run asynchronously due to the length of the operation.  |
| SessionControl()                      |       | Mandatory   | The method enables a client to request a lock of the fabric to begin zoning configuration changes.  |
| SessionControlWithJ<br>ob()           |       | Mandatory   | The method enables a client to request a lock of the fabric to begin zoning configuration changes. The method has an optional input, ConfigurationID, which is used to confirm that the ZoneSet being activated matches the definition the application previously discovered or changed. This method may run asynchronously due to the length of the operation. |

**6.10.2 CIM\_HostedService**

Associates the ZoneService to the AdminDomain representing the fabric or the ComputerSystem representing the switch.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 45 describes class CIM\_HostedService.

**Table 45: SMI Referenced Properties/Methods for CIM\_HostedService**

| Properties | Flags | Requirement | Description & Notes   |
|------------|-------|-------------|---|
| Antecedent |       | Mandatory   | The reference to the AdminDomain representing the fabric or the ComputerSystem representing the switch. |

**Table 45: SMI Referenced Properties/Methods for CIM\_HostedService**

| Properties | Flags | Requirement | Description & Notes               |
|------------|-------|-------------|-----------------------------------|
| Dependent  |       | Mandatory   | The reference to the ZoneService. |

**STABLE**

---

---



---

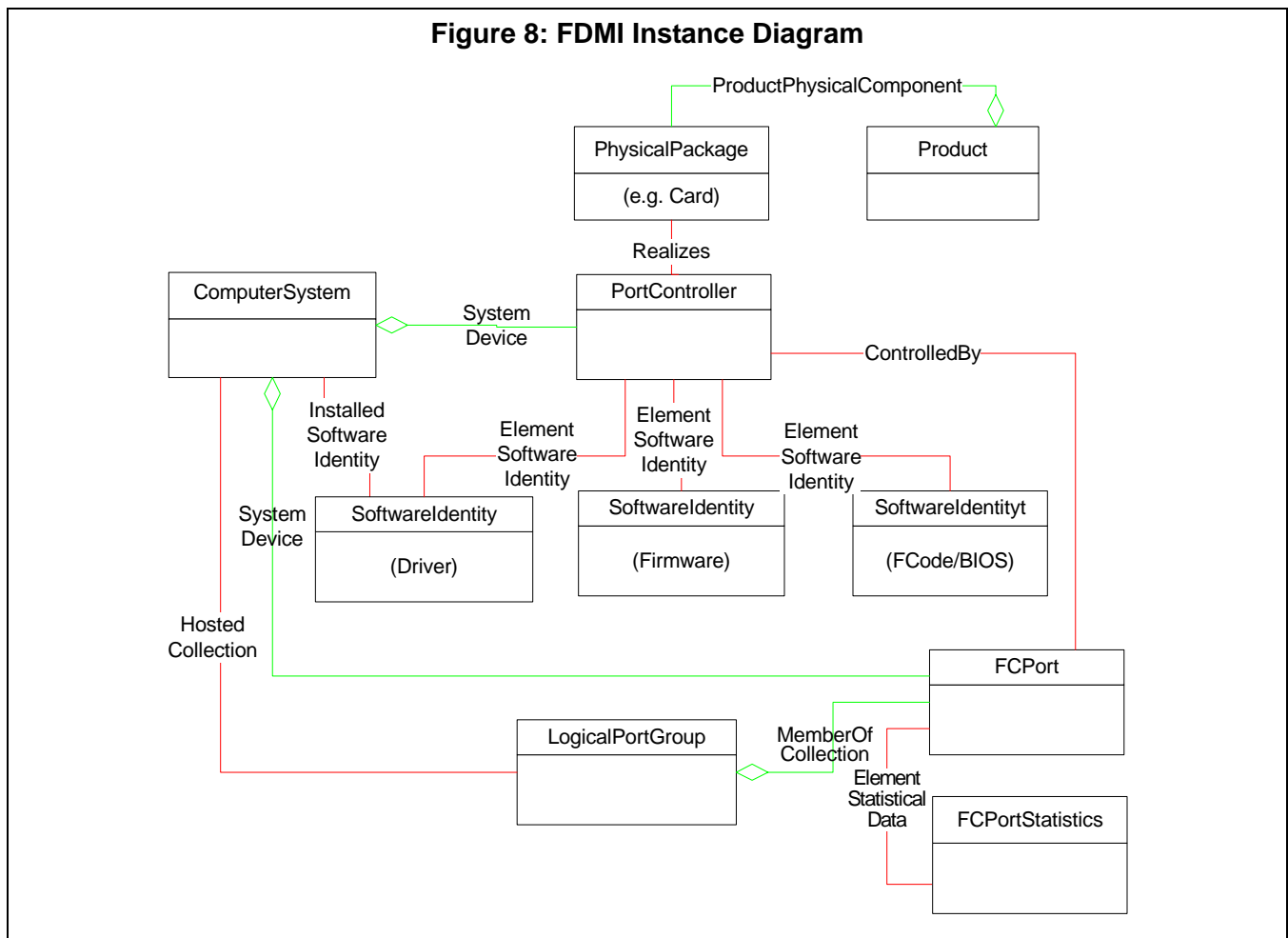


---

**IMPLEMENTED**
**Clause 7: FDMI Subprofile**
**7.1 Description**

The Fabric-Device Management Interface (FDMI) enables the management of devices such as HBAs through the Fabric. The FDMI complements data in the Fabric Profile. It allows for any entity in the Fabric to expose through SMI the HBA information without having an agent resident on the Host containing the HBA.

This profile only addresses HBA type devices. The HBA Management Interface defined by FDMI is a subset of interface defined by the Fibre Channel HBA API specification, as exposed by *Storage Management Technical Specification, Part 6 Host Elements* Clause 6: FC HBA Profile.

**Figure 8: FDMI Instance Diagram**

**7.2 Health and Fault Management**

None

**7.3 Cascading Considerations**

None

#### **7.4 Dependencies on Profiles, Subprofiles, and Packages**

None

#### **7.5 Methods of this Profile**

None

#### **7.6 Client Considerations and Recipes**

None

#### **7.7 Registered Name and Version**

FDMI version 1.2.0

## 7.8 CIM Elements

**Table 46: CIM Elements for FDMI**

| Element Name   | Requirement | Description   |
|--|-------------|---|
| CIM_ComputerSystem (7.8.1)                                   | Mandatory   | The System the HBA is within.   |
| CIM_ControlledBy (7.8.2)                                     | Mandatory   | Associates the ComputerSystem with the PortController                                     |
| CIM_ElementSoftwareIdentity (7.8.3)                          | Mandatory   | Associates the SoftwareIdentity to the HBA  |
| CIM_FCPort (7.8.4)   | Mandatory   | The HBA Fibre Channel Port  |
| CIM_HostedCollection (7.8.5)                                 | Mandatory   | Associates the LogicalPortGroup (Fibre Channel Node) to the hosting System.               |
| CIM_InstalledSoftwareIdentity (7.8.6)                        | Mandatory   | Associates the SoftwareIdentity representing the driver to the System it is installed on. |
| CIM_LogicalPortGroup (7.8.7)                                 | Mandatory   | The Fibre Channel Node  |
| CIM_MemberOfCollection (7.8.8)                               | Mandatory   | Associates FCPort to the LogicalPortGroup   |
| CIM_PhysicalPackage (7.8.9)                                  | Mandatory   | The physical package that the HBA is contained in   |
| CIM_PortController (7.8.10)                                  | Mandatory   | The HBA   |
| CIM_Product (7.8.11)   | Mandatory   | The product information for the HBA   |
| CIM_ProductPhysicalComponent (7.8.12)                        | Mandatory   | Associates the Product to the PhysicalPackage   |
| CIM_Realizes (7.8.13)  | Conditional | Associates the PhysicalPackage to the PortController                                      |
| CIM_SoftwareIdentity (Firmware) (7.8.14)                     | Mandatory   | The software for the firmware   |
| CIM_SoftwareIdentity (Driver) (7.8.15)                       | Mandatory   | The software for the driver   |
| CIM_SoftwareIdentity (Option ROM) (7.8.16)                   | Mandatory   | The software for the Option ROM   |
| CIM_SystemDevice (7.8.17)                                    | Mandatory   | Associates the ComputerSystem with the FCPort   |
| CIM_SystemDevice (ComputerSystem to PortController) (7.8.18) | Mandatory   | Associates the ComputerSystem with the PortController                                     |
| CIM_SystemDevice (ComputerSystem to FCPort) (7.8.19)         | Optional    | Associates the FCPort to the ComputerSystem   |
| CIM_SystemDevice (Switch to FCPort) (7.8.20)                 | Optional    | Associates the FCPort to the ComputerSystem   |
| CIM_SystemDevice (System to FCPort) (7.8.21)                 | Optional    | Associates the FCPort to the ComputerSystem   |

### 7.8.1 CIM\_ComputerSystem

The system the HBA is within. It is identified using Host Name from the FDMI interface.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 47 describes class CIM\_ComputerSystem.

**Table 47: SMI Referenced Properties/Methods for CIM\_ComputerSystem**

| Properties        | Flags | Requirement | Description & Notes   |
|-------------------|-------|-------------|---|
| CreationClassName |       | Mandatory   |   |
| Name              |       | Mandatory   | The name of the host containing the Device. The key identifier helping in discovery to determine which HBAs are in the same host. |
| NameFormat        |       | Mandatory   |   |

### 7.8.2 CIM\_ControlledBy

Associates the ComputerSystem with the PortController.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 48 describes class CIM\_ControlledBy.

**Table 48: SMI Referenced Properties/Methods for CIM\_ControlledBy**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Dependent  |       | Mandatory   |                     |
| Antecedent |       | Mandatory   |                     |

### 7.8.3 CIM\_ElementSoftwareIdentity

Associates the SoftwareIdentities representing the various software for the HBA to the PortController representing the HBA.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 49 describes class CIM\_ElementSoftwareIdentity.

**Table 49: SMI Referenced Properties/Methods for CIM\_ElementSoftwareIdentity**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Antecedent |       | Mandatory   |                     |
| Dependent  |       | Mandatory   |                     |

#### 7.8.4 CIM\_FCPort

The HBA Fibre Channel Port.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 50 describes class CIM\_FCPort.

**Table 50: SMI Referenced Properties/Methods for CIM\_FCPort**

| Properties              | Flags | Requirement | Description & Notes  |
|-------------------------|-------|-------------|--|
| SystemCreationClassName |       | Mandatory   | The scoping System's CreationClassName.  |
| SystemName              |       | Mandatory   | The scoping System's Name.   |
| CreationClassName       |       | Mandatory   | Name of Class  |
| DeviceID                |       | Mandatory   | Opaque   |
| ElementName             |       | Mandatory   | Port Symbolic Name if available. Otherwise NULL. If the underlying implementation includes characters that are illegal in CIM strings, then truncate before the first of those characters.   |
| LinkTechnology          |       | Mandatory   | 'FC'   |
| PermanentAddress        |       | Mandatory   | Fibre Channel Port WWN   |
| NetworkAddresses        | C     | Mandatory   | Fibre Channel ID (FCID). Expressed as 8 un-separated upper case hex digits (see Table 4 for more information about formats).   |
| ActiveFC4Types          |       | Mandatory   | The active Fibre Channel FC-4 protocol   |
| SupportedFC4Types       |       | Optional    | <p>An array of integers indicating the Fibre Channel FC-4 protocols supported</p> <p><b>EXPERIMENTAL</b></p> <p>When supporting FC-SB, this property shall be either '27' (FC-SB-2 Channel) for Host Ports or '28' (FC-SB-2 Control Unit) for targets.</p> |

**Table 50: SMI Referenced Properties/Methods for CIM\_FCPort**

| Properties   | Flags | Requirement | Description & Notes  |
|--------------|-------|-------------|--|
| SupportedCOS |       | Optional    | An array of integers indicating the Fibre Channel Classes of Service that are supported.   |
| PortType     |       | Mandatory   | The specific port type currently enabled (from FC-GS Port.Type)  |
| Speed        |       | Optional    | Speed of zero represents a link not established.<br>1Gb is 1062500000 bps<br>2Gb is 2125000000 bps<br>4Gb is 4250000000 bps<br>10Gb single channel variants are 10518750000 bps<br>10Gb four channel variants are 12750000000 bps<br>This is the raw bit rate. |

**7.8.5 CIM\_HostedCollection**

Associates the LogicalPortGroup (Fibre Channel Node) to the hosting System.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 51 describes class CIM\_HostedCollection.

**Table 51: SMI Referenced Properties/Methods for CIM\_HostedCollection**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Antecedent |       | Mandatory   |                     |
| Dependent  |       | Mandatory   |                     |

**7.8.6 CIM\_InstalledSoftwareIdentity**

Associates the SoftwareIdentity representing the driver to the System it is installed on.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 52 describes class CIM\_InstalledSoftwareIdentity.

**Table 52: SMI Referenced Properties/Methods for CIM\_InstalledSoftwareIdentity**

| Properties        | Flags | Requirement | Description & Notes |
|-------------------|-------|-------------|---------------------|
| InstalledSoftware |       | Mandatory   |                     |
| System            |       | Mandatory   |                     |

### 7.8.7 CIM\_LogicalPortGroup

The Fibre Channel Node

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 53 describes class CIM\_LogicalPortGroup.

**Table 53: SMI Referenced Properties/Methods for CIM\_LogicalPortGroup**

| Properties  | Flags | Requirement | Description & Notes  |
|-------------|-------|-------------|--|
| InstanceID  |       | Mandatory   | Opaque   |
| Name        | D     | Mandatory   | Fibre Channel Node WWN   |
| NameFormat  |       | Mandatory   | 'WWN'  |
| ElementName | N     | Mandatory   | Node Symbolic Name if available. Otherwise NULL. If the underlying implementation includes characters that are illegal in CIM strings, then truncate before the first of those characters. |

### 7.8.8 CIM\_MemberOfCollection

Associates FCPort to the LogicalPortGroup

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 54 describes class CIM\_MemberOfCollection.

**Table 54: SMI Referenced Properties/Methods for CIM\_MemberOfCollection**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Collection |       | Mandatory   |                     |
| Member     |       | Mandatory   |                     |

### 7.8.9 CIM\_PhysicalPackage

The physical package that the HBA is contained by. It can be simply a PhysicalPackage that the system and HBA is contained within. If it is known that the HBA is on a separate board, Card (a subclass of PhysicalPackage) can be used.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 55 describes class CIM\_PhysicalPackage.

**Table 55: SMI Referenced Properties/Methods for CIM\_PhysicalPackage**

| Properties        | Flags | Requirement | Description & Notes   |
|-------------------|-------|-------------|---|
| CreationClassName |       | Mandatory   | Name of Class   |
| Tag               |       | Mandatory   | An arbitrary string that uniquely identifies the PhysicalPackage. |
| ElementName       |       | Optional    | User Friendly name.<br>This property is OPTIONAL.                 |
| Name              |       | Optional    |   |
| Manufacturer      |       | Mandatory   |   |
| Model             |       | Mandatory   |   |
| SerialNumber      |       | Optional    |   |
| Version           |       | Optional    |   |
| PartNumber        |       | Optional    |   |

### 7.8.10 CIM\_PortController

The HBA. The HBA may have logical operations that can apply to it (e.g. OperationalStatus).

Created By: Static

Modified By: Static

Deleted By: Static



Class Mandatory: Mandatory

Table 56 describes class CIM\_PortController.

**Table 56: SMI Referenced Properties/Methods for CIM\_PortController**

| Properties              | Flags | Requirement | Description & Notes |
|-------------------------|-------|-------------|---------------------|
| SystemCreationClassName |       | Mandatory   |                     |
| SystemName              |       | Mandatory   |                     |
| CreationClassName       |       | Mandatory   |                     |
| DeviceID                |       | Mandatory   |                     |
| ControllerType          |       | Mandatory   |                     |

#### 7.8.11 CIM\_Product

The product information for the HBA

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 57 describes class CIM\_Product.

**Table 57: SMI Referenced Properties/Methods for CIM\_Product**

| Properties        | Flags | Requirement | Description & Notes  |
|-------------------|-------|-------------|--|
| Name              |       | Mandatory   | Commonly used Product name.  |
| IdentifyingNumber |       | Mandatory   | Product identification such as a serial number.                        |
| Vendor            |       | Mandatory   | The manufacturer or the OEM.   |
| Version           |       | Mandatory   | Product version information.   |
| ElementName       |       | Mandatory   | User Friendly name. Suggested use is Vendor, Version and product name. |

#### 7.8.12 CIM\_ProductPhysicalComponent

Associates the Product to the PhysicalPackage. This is necessary to link the Product information to the HBA.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 58 describes class CIM\_ProductPhysicalComponent.

**Table 58: SMI Referenced Properties/Methods for CIM\_ProductPhysicalComponent**

| Properties     | Flags | Requirement | Description & Notes |
|----------------|-------|-------------|---------------------|
| GroupComponent |       | Mandatory   |                     |
| PartComponent  |       | Mandatory   |                     |

#### 7.8.13 CIM\_Realizes

Associates the PhysicalPackage to the PortController.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: C2

Table 59 describes class CIM\_Realizes.

**Table 59: SMI Referenced Properties/Methods for CIM\_Realizes**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Antecedent |       | Mandatory   |                     |
| Dependent  |       | Mandatory   |                     |
| Antecedent |       | Mandatory   |                     |
| Dependent  |       | Mandatory   |                     |

#### 7.8.14 CIM\_SoftwareIdentity (Firmware)

The software for the firmware

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 60 describes class CIM\_SoftwareIdentity (Firmware).

**Table 60: SMI Referenced Properties/Methods for CIM\_SoftwareIdentity (Firmware)**

| Properties      | Flags | Requirement | Description & Notes |
|-----------------|-------|-------------|---------------------|
| InstanceID      |       | Mandatory   |                     |
| VersionString   |       | Mandatory   |                     |
| Manufacturer    |       | Mandatory   |                     |
| BuildNumber     |       | Optional    |                     |
| MajorVersion    |       | Optional    |                     |
| RevisionNumber  |       | Optional    |                     |
| MinorVersion    |       | Optional    |                     |
| Classifications |       | Mandatory   |                     |

#### 7.8.15 CIM\_SoftwareIdentity (Driver)

The software for the driver

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 61 describes class CIM\_SoftwareIdentity (Driver).

**Table 61: SMI Referenced Properties/Methods for CIM\_SoftwareIdentity (Driver)**

| Properties      | Flags | Requirement | Description & Notes |
|-----------------|-------|-------------|---------------------|
| InstanceID      |       | Mandatory   |                     |
| VersionString   |       | Mandatory   |                     |
| Manufacturer    |       | Mandatory   |                     |
| BuildNumber     |       | Optional    |                     |
| MajorVersion    |       | Optional    |                     |
| RevisionNumber  |       | Optional    |                     |
| MinorVersion    |       | Optional    |                     |
| Classifications |       | Mandatory   |                     |

#### 7.8.16 CIM\_SoftwareIdentity (Option ROM)

The software for the Option ROM

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 62 describes class CIM\_SoftwareIdentity (Option ROM).

**Table 62: SMI Referenced Properties/Methods for CIM\_SoftwareIdentity (Option ROM)**

| Properties      | Flags | Requirement | Description & Notes |
|-----------------|-------|-------------|---------------------|
| InstanceID      |       | Mandatory   |                     |
| VersionString   |       | Mandatory   |                     |
| Manufacturer    |       | Mandatory   |                     |
| BuildNumber     |       | Optional    |                     |
| MajorVersion    |       | Optional    |                     |
| RevisionNumber  |       | Optional    |                     |
| MinorVersion    |       | Optional    |                     |
| Classifications |       | Mandatory   |                     |

#### 7.8.17 CIM\_SystemDevice

Associates the ComputerSystem with the FCPort

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 63 describes class CIM\_SystemDevice.

**Table 63: SMI Referenced Properties/Methods for CIM\_SystemDevice**

| Properties     | Flags | Requirement | Description & Notes |
|----------------|-------|-------------|---------------------|
| PartComponent  |       | Mandatory   |                     |
| GroupComponent |       | Mandatory   |                     |

#### 7.8.18 CIM\_SystemDevice (ComputerSystem to PortController)

Associates the ComputerSystem with the PortController

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 64 describes class CIM\_SystemDevice (ComputerSystem to PortController).

**Table 64: SMI Referenced Properties/Methods for CIM\_SystemDevice (ComputerSystem to PortController)**

| Properties     | Flags | Requirement | Description & Notes |
|----------------|-------|-------------|---------------------|
| PartComponent  |       | Mandatory   |                     |
| GroupComponent |       | Mandatory   |                     |

#### 7.8.19 CIM\_SystemDevice (ComputerSystem to FCPort)

Associates the FCPort to the ComputerSystem

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Optional

Table 65 describes class CIM\_SystemDevice (ComputerSystem to FCPort).

**Table 65: SMI Referenced Properties/Methods for CIM\_SystemDevice (ComputerSystem to FCPort)**

| Properties     | Flags | Requirement | Description & Notes |
|----------------|-------|-------------|---------------------|
| PartComponent  |       | Mandatory   |                     |
| GroupComponent |       | Mandatory   |                     |

#### 7.8.20 CIM\_SystemDevice (Switch to FCPort)

Associates the FCPort to the ComputerSystem

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Optional

Table 66 describes class CIM\_SystemDevice (Switch to FCPort).

**Table 66: SMI Referenced Properties/Methods for CIM\_SystemDevice (Switch to FCPort)**

| Properties     | Flags | Requirement | Description & Notes |
|----------------|-------|-------------|---------------------|
| GroupComponent |       | Mandatory   |                     |

**Table 66: SMI Referenced Properties/Methods for CIM\_SystemDevice (Switch to FCPort)**

| Properties    | Flags | Requirement | Description & Notes |
|---------------|-------|-------------|---------------------|
| PartComponent |       | Mandatory   |                     |

**7.8.21 CIM\_SystemDevice (System to FCPort)**

Associates the FCPort to the ComputerSystem

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Optional

Table 67 describes class CIM\_SystemDevice (System to FCPort).

**Table 67: SMI Referenced Properties/Methods for CIM\_SystemDevice (System to FCPort)**

| Properties     | Flags | Requirement | Description & Notes |
|----------------|-------|-------------|---------------------|
| GroupComponent |       | Mandatory   |                     |
| PartComponent  |       | Mandatory   |                     |

**IMPLEMENTED**

---

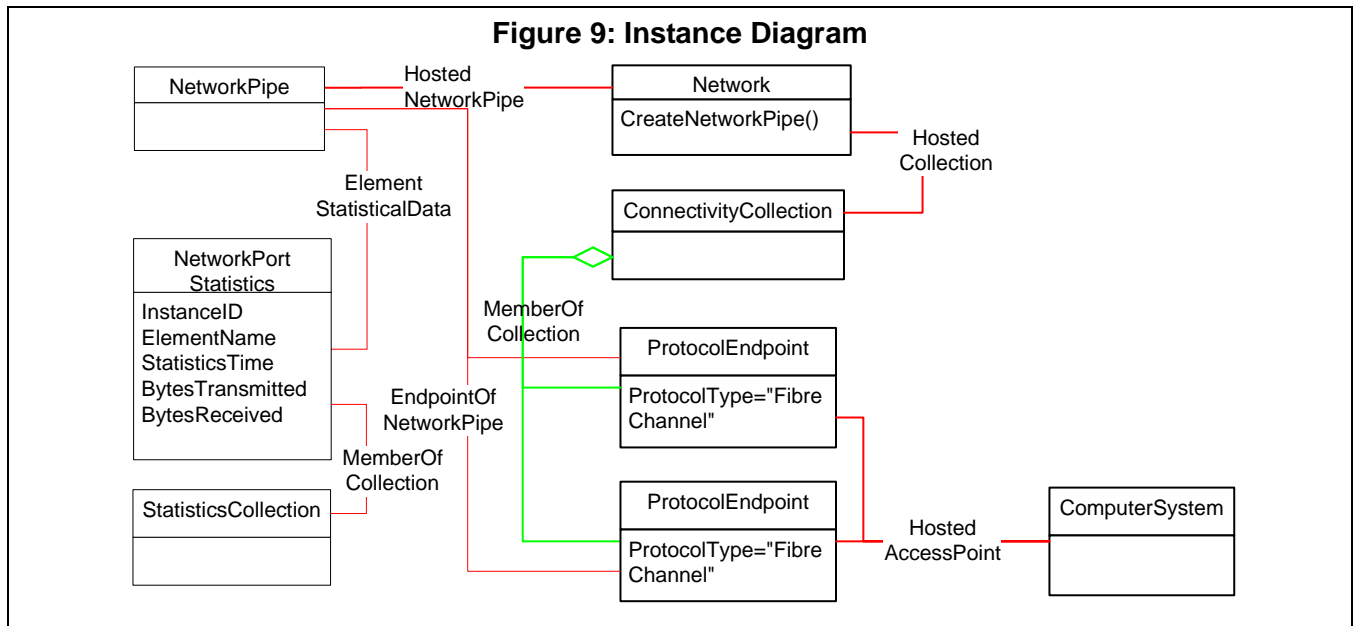


---

**EXPERIMENTAL**
**Clause 8: Fabric Path Performance Subprofile**
**8.1 Description**

The fabric path performance subprofile extends the standard capabilities of obtaining performance associated to a port to identify the performance in the path defined by an initiator and target ProtocolEndpoint. In the current networking model, the path through the “cloud” is defined by NetworkPipe which is a class that is associated to a ProtocolEndpoint by EndpointOfNetworkPipe. Since the statistics model is defined to allow an association to any LogicalElement, the statistics collected for an NetworkPort, NetworkPortStatistics, can be associated to the NetworkPipe also. When a device supports the Fabric Path Performance Subprofile, it will instantiate the NetworkPipe and as it collects statistics will instantiate the StatisticalData.

The class, StatisticsCollection, provides a mechanism to “collect” all the statistics associated to the NetworkPipes.


**8.2 Health and Fault Management**

None

**8.3 Dependencies on Profiles, Subprofiles, and Packages**

None

**8.4 Methods of this Profile**

The method `CreateNetworkPipe` is used to indicate to the underlying instrumentation that statistics shall be collected between the two `ServiceAccessPoints`. A `NetworkPipe` maybe created for each set of statistics needed, though the underlying hardware may have maximum limitations. If the creation of the `NetworkPipe` is successful, the client can then discover the `NetworkPipe` and also the associated statics. It is implementation specific how long the statuses are collected. For instance, the `NetworkPipes` may persist as long as the hardware is not rebooted.

```
CreateNetworkPipe (
```

```

[IN] Directionality=2,
[IN] CIM_ServiceAccessPoint REF SourceSAP,
[IN] CIM_ServiceAccessPoint REF SinkSAP,
[OUT] CIM_NetworkPipe REF NetworkPipe);

```

## 8.5 Client Considerations and Recipes

None

## 8.6 Registered Name and Version

FabricPathPerformance version 1.1.0

## 8.7 CIM Elements

**Table 68: CIM Elements for FabricPathPerformance**

| Element Name                       | Requirement | Description  |
|------------------------------------|-------------|--|
| CIM_Network (8.7.1)                | Mandatory   | Subclass of AdminDomain representing the fabric                              |
| CIM_ElementStatisticalData (8.7.2) | Mandatory   | Associates FCPortStatistics to the FCPort                                    |
| CIM_HostedNetworkPipe (8.7.3)      | Mandatory   | Associates NetworkPipe to the Network  |
| CIM_EndpointOfNetworkPipe (8.7.4)  | Mandatory   | Associates FCPortRateStatistics to the FCPort                                |
| CIM_NetworkPortStatistics (8.7.5)  | Mandatory   | NetworkPort Statistics of the NetworkPipe                                    |
| CIM_NetworkPipe (8.7.6)            | Mandatory   | Pipe through the cloud from an initiator to the target.                      |
| CIM_ProtocolEndpoint (8.7.7)       | Mandatory   | The initiator or target (ends of the NetworkPipe).                           |
| CIM_MemberOfCollection (8.7.8)     | Mandatory   | Associates the NetworkPortStatistics to the StatisticsCollection.            |
| CIM_HostedCollection (8.7.9)       | Mandatory   | Associates the Statistics Collection to the Network representing the fabric. |
| CIM_StatisticsCollection (8.7.10)  | Mandatory   | Collection to aggregate the NetworkPipe statistics                           |

### 8.7.1 CIM\_Network

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory



Table 69 describes class CIM\_Network.

**Table 69: SMI Referenced Properties/Methods for CIM\_Network**

| Properties          | Flags | Requirement | Description & Notes |
|---------------------|-------|-------------|---------------------|
| CreationClassName   |       | Mandatory   |                     |
| Name                |       | Mandatory   |                     |
| NameFormat          |       | Mandatory   |                     |
| CreateNetworkPipe() |       | Optional    |                     |

### 8.7.2 CIM\_ElementStatisticalData

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 70 describes class CIM\_ElementStatisticalData.

**Table 70: SMI Referenced Properties/Methods for CIM\_ElementStatisticalData**

| Properties     | Flags | Requirement | Description & Notes |
|----------------|-------|-------------|---------------------|
| ManagedElement |       | Mandatory   |                     |
| Stats          |       | Mandatory   |                     |

### 8.7.3 CIM\_HostedNetworkPipe

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 71 describes class CIM\_HostedNetworkPipe.

**Table 71: SMI Referenced Properties/Methods for CIM\_HostedNetworkPipe**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Antecedent |       | Mandatory   | NetworkPipe         |

**Table 71: SMI Referenced Properties/Methods for CIM\_HostedNetworkPipe**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Dependent  |       | Mandatory   | Network             |
| Antecedent |       | Mandatory   |                     |
| Dependent  |       | Mandatory   |                     |

**8.7.4 CIM\_EndpointOfNetworkPipe**

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 72 describes class CIM\_EndpointOfNetworkPipe.

**Table 72: SMI Referenced Properties/Methods for CIM\_EndpointOfNetworkPipe**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Antecedent |       | Mandatory   |                     |
| Dependent  |       | Mandatory   |                     |

**8.7.5 CIM\_NetworkPortStatistics**

Network Port Statistics represent a snapshots of counters for the NetworkPipe. An instance of this class can represent the statistics for the current statistics, archived and consolidated statistics, or both.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 73 describes class CIM\_NetworkPortStatistics.

**Table 73: SMI Referenced Properties/Methods for CIM\_NetworkPortStatistics**

| Properties    | Flags | Requirement | Description & Notes   |
|---------------|-------|-------------|---|
| InstanceID    |       | Mandatory   | Opaque  |
| ElementName   |       | Optional    |   |
| StatisticTime |       | Optional    | The time the statistics were collected. If historical data is instantiated (present), this property shall be set with the time representing the time the statistic was collected. |

**Table 73: SMI Referenced Properties/Methods for CIM\_NetworkPortStatistics**

| Properties       | Flags | Requirement | Description & Notes |
|------------------|-------|-------------|---------------------|
| BytesTransmitted |       | Mandatory   |                     |
| BytesReceived    |       | Mandatory   |                     |

**8.7.6 CIM\_NetworkPipe**

The NetworkPipe for this profile is instantiated to provide a mechanism to indicate monitors are in place in the network to collect statistical information. NetworkPortStatistics are associated to the pipe via the association ElementStatisticalData to NetworkPortStatistics and subclasses of NetworkPortStatistics (e.g. FCPortStatistics).

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 74 describes class CIM\_NetworkPipe.

**Table 74: SMI Referenced Properties/Methods for CIM\_NetworkPipe**

| Properties  | Flags | Requirement | Description & Notes |
|-------------|-------|-------------|---------------------|
| InstanceID  |       | Mandatory   |                     |
| ElementName |       | Optional    |                     |

**8.7.7 CIM\_ProtocolEndpoint**

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 75 describes class CIM\_ProtocolEndpoint.

**Table 75: SMI Referenced Properties/Methods for CIM\_ProtocolEndpoint**

| Properties              | Flags | Requirement | Description & Notes |
|-------------------------|-------|-------------|---------------------|
| SystemCreationClassName |       | Mandatory   |                     |
| SystemName              |       | Mandatory   |                     |
| CreationClassName       |       | Mandatory   |                     |
| Name                    |       | Mandatory   |                     |

**Table 75: SMI Referenced Properties/Methods for CIM\_ProtocolEndpoint**

| Properties     | Flags | Requirement | Description & Notes |
|----------------|-------|-------------|---------------------|
| NameFormat     |       | Mandatory   |                     |
| ProtocolIFType |       | Mandatory   |                     |

**8.7.8 CIM\_MemberOfCollection**

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 76 describes class CIM\_MemberOfCollection.

**Table 76: SMI Referenced Properties/Methods for CIM\_MemberOfCollection**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Collection |       | Mandatory   |                     |
| Member     |       | Mandatory   |                     |

**8.7.9 CIM\_HostedCollection**

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 77 describes class CIM\_HostedCollection.

**Table 77: SMI Referenced Properties/Methods for CIM\_HostedCollection**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Dependent  |       | Mandatory   |                     |
| Antecedent |       | Mandatory   |                     |

**8.7.10 CIM\_StatisticsCollection**

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 78 describes class CIM\_StatisticsCollection.

**Table 78: SMI Referenced Properties/Methods for CIM\_StatisticsCollection**

| Properties      | Flags | Requirement | Description & Notes |
|-----------------|-------|-------------|---------------------|
| InstanceID      |       | Mandatory   |                     |
| ElementName     |       | Mandatory   |                     |
| SampleInterval  |       | Mandatory   |                     |
| TimeLastSampled |       | Mandatory   |                     |

## **EXPERIMENTAL**

---

---



---

---

**EXPERIMENTAL****Clause 9: Fibre Channel Security Subprofile****9.1 Description**

Fibre Channel Security Subprofile describes the Fibre Channel Security Protocol (FC-SP) management policies. This profile is a specialization of the Authorization Subprofile.

The FC-SP Policies can be broken down into four parts:

- Membership
- Connectivity
- Management Access
- Attributes

FC-SP Membership policy is composed of two components, Switch Membership and Device Membership. Switch Membership is fabric-wide information which defines which switches are allowed to be part of a fabric, controls physical management access, and allows switch characteristics to be specified. Device Membership is fabric-wide information which defines which devices are allowed to be part of a fabric, controls in-band management access, and allows device characteristics to be specified. FC-SP Connectivity policy defines connectivity restrictions on a per-switch basis.

This subprofile describes exposing the FC-SP Membership and Connectivity policies. It does not define physical management and switch characteristics as part of the Membership policy. The specialization of the Authorization Subprofile primarily includes the subclassing of Identity to StorageHardwareID to identify the AuthorizedSubject and AuthorizedTargets.

For the Membership Policy, the AuthorizedTarget is always the Fabric identified by its Fabric Name and the two possible AuthorizedSubjects are the switch identified by its switch name (WWN) and a device by its node WWN. In Figure 10, the diagram shows both a switch having membership and a device having membership.

**Figure 10: Specialization of Security Authorization Subprofile for Membership Policy**

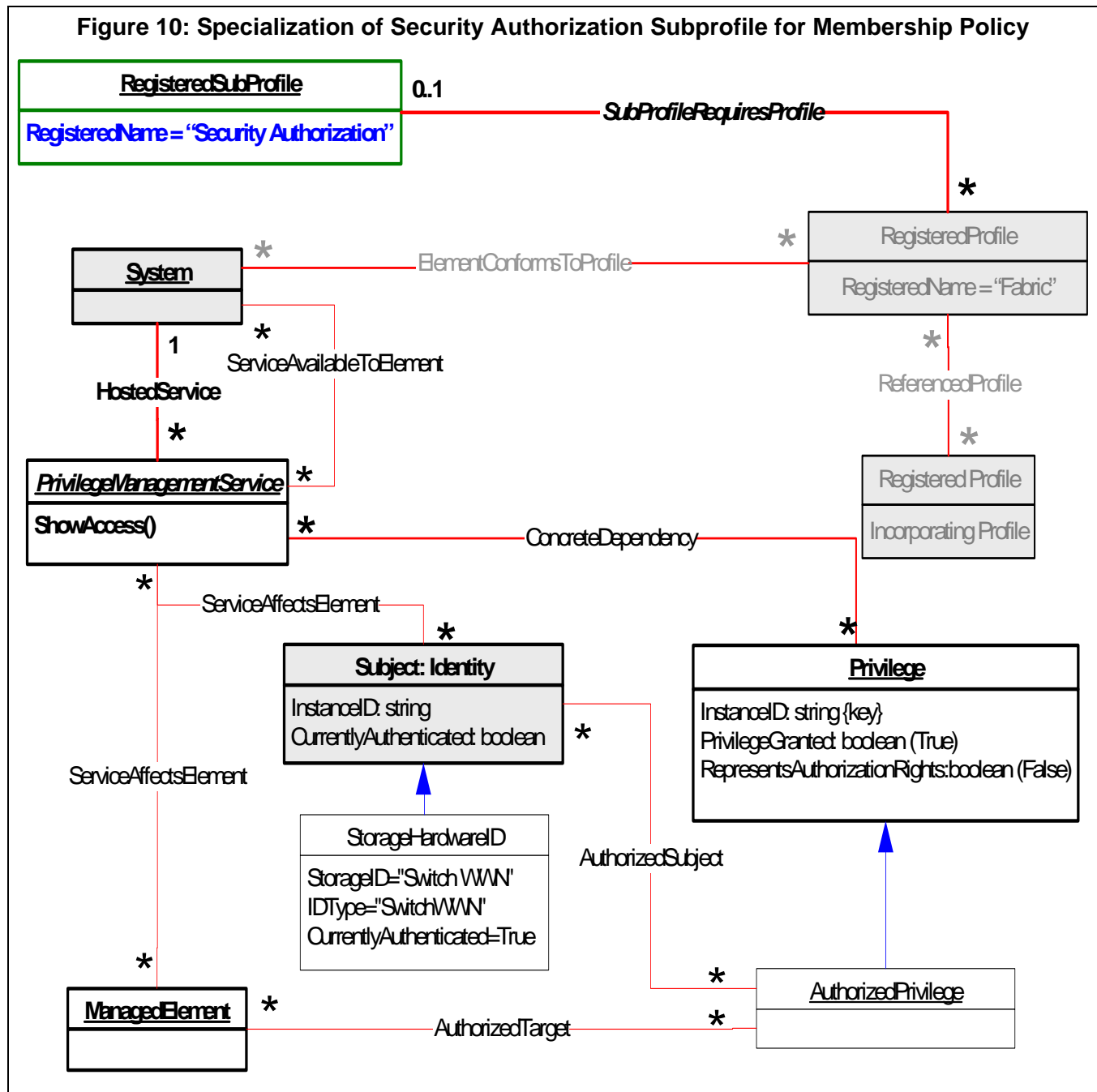
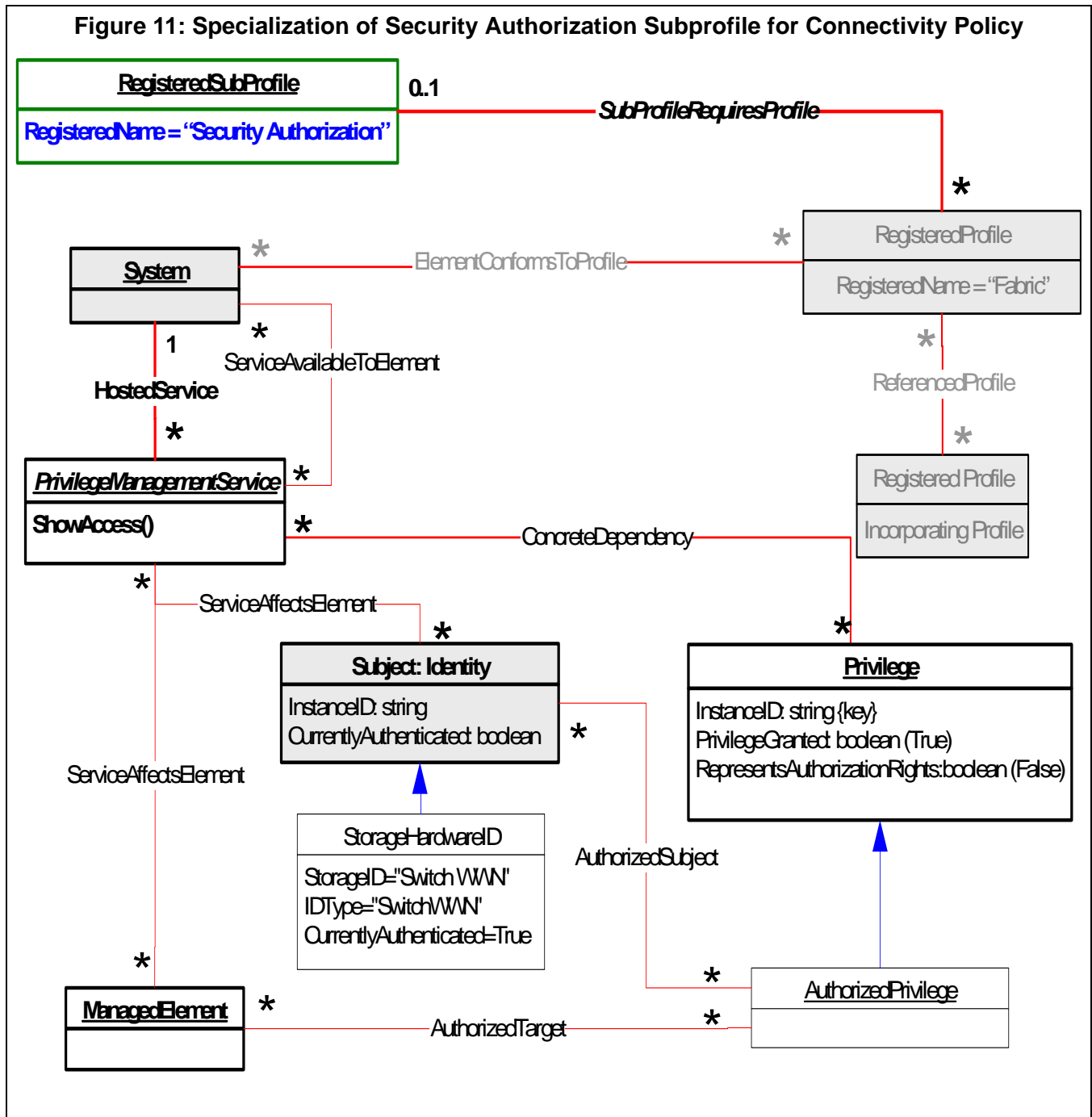




Figure 11 shows both a switch having Switch Membership and a device having Device Membership. A switch is identified by its switch name (WWN) and a device by its node WWN.



## 9.2 Health and Fault Management Consideration

None

## 9.3 Cascading Considerations

None

## 9.4 Supported Profiles, Subprofiles, and Packages

None

## 9.5 Methods of the Profile

None

## 9.6 Client Considerations and Recipes

None

## 9.7 Registered Name and Version

FabricSecurity version 1.2.0

## 9.8 CIM Elements

**Table 79: CIM Elements for FabricSecurity**

| Element Name  | Requirement | Description   |
|---|-------------|---|
| CIM_AuthorizationService (9.8.1)                                      | Mandatory   | The service controlling the security policy in the fabric   |
| CIM_HostedService (9.8.2)   | Mandatory   | Associates the AuthorizationService to the AdminDomain that is hosting it   |
| CIM_AuthorizedPrivilege (9.8.3)                                       | Mandatory   | Privilege granted to the Switch or the Node   |
| CIM_AuthorizedTarget (9.8.4)  | Mandatory   | The association of the privilege (or a switch or node) to the target (switch or the fabric) that the switch or node is being granted access to. |
| CIM_AuthorizedSubject (9.8.5)   | Mandatory   | The association of the Node or Switch to its Privilege.   |
| CIM_StorageHardwareID (9.8.6)   | Mandatory   | The class that identifies the subject to be granted access  |
| CIM_ServiceAvailableToElement (Fabric AdminDomain to Service) (9.8.7) | Mandatory   | Service available to fabric   |
| CIM_ServiceAffectsElement (ManagedElement to Service) (9.8.8)         | Mandatory   | Service Affects Managed Element   |
| CIM_ServiceAffectsElement (StorageHardwareID to Service) (9.8.9)      | Mandatory   | Service affects StorageHardwareID   |

### 9.8.1 CIM\_AuthorizationService

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 80 describes class CIM\_AuthorizationService.

**Table 80: SMI Referenced Properties/Methods for CIM\_AuthorizationService**

| Properties              | Flags | Requirement | Description & Notes                           |
|-------------------------|-------|-------------|---|
| SystemCreationClassName |       | Mandatory   | Name of System class                          |
| SystemName              |       | Mandatory   | Name of class                                 |
| CreationClassName       |       | Mandatory   | Name of AuthorizationService class            |
| Name                    |       | Mandatory   | Uniquely identifies the service in the fabric |

### 9.8.2 CIM\_HostedService

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 81 describes class CIM\_HostedService.

**Table 81: SMI Referenced Properties/Methods for CIM\_HostedService**

| Properties | Flags | Requirement | Description & Notes                 |
|------------|-------|-------------|-------------------------------------|
| Antecedent |       | Mandatory   | AdminDomain representing the Fabric |
| Dependent  |       | Mandatory   | AuthorizationService                |

### 9.8.3 CIM\_AuthorizedPrivilege

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 82 describes class CIM\_AuthorizedPrivilege.

**Table 82: SMI Referenced Properties/Methods for CIM\_AuthorizedPrivilege**

| Properties                    | Flags | Requirement | Description & Notes |
|-------------------------------|-------|-------------|---------------------|
| InstanceID                    |       | Mandatory   | Opaque              |
| PrivilegeGranted              |       | Mandatory   |                     |
| RepresentsAuthorizationRights |       | Mandatory   |                     |

#### 9.8.4 CIM\_AuthorizedTarget

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 83 describes class CIM\_AuthorizedTarget.

**Table 83: SMI Referenced Properties/Methods for CIM\_AuthorizedTarget**

| Properties    | Flags | Requirement | Description & Notes         |
|---------------|-------|-------------|-----------------------------|
| Privilege     |       | Mandatory   |                             |
| TargetElement |       | Mandatory   | Reference to target element |

#### 9.8.5 CIM\_AuthorizedSubject

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 84 describes class CIM\_AuthorizedSubject.

**Table 84: SMI Referenced Properties/Methods for CIM\_AuthorizedSubject**

| Properties        | Flags | Requirement | Description & Notes     |
|-------------------|-------|-------------|-------------------------|
| PrivilegedElement |       | Mandatory   |                         |
| Privilege         |       | Mandatory   | CIM_AuthorizedPrivilege |

### 9.8.6 CIM\_StorageHardwareID

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 85 describes class CIM\_StorageHardwareID.

**Table 85: SMI Referenced Properties/Methods for CIM\_StorageHardwareID**

| Properties             | Flags | Requirement | Description & Notes               |
|------------------------|-------|-------------|-----------------------------------|
| InstanceID             |       | Mandatory   | Opaque                            |
| StorageID              |       | Mandatory   | Node WWN, Switch WWN, Port WWN    |
| IDType                 |       | Mandatory   | "NodeWWN" "SwitchWWN" or PortWWN" |
| CurrentlyAuthenticated |       | Mandatory   |                                   |

### 9.8.7 CIM\_ServiceAvailableToElement (Fabric AdminDomain to Service)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 86 describes class CIM\_ServiceAvailableToElement (Fabric AdminDomain to Service).

**Table 86: SMI Referenced Properties/Methods for CIM\_ServiceAvailableToElement (Fabric AdminDomain to Service)**

| Properties      | Flags | Requirement | Description & Notes                 |
|-----------------|-------|-------------|-------------------------------------|
| ServiceProvided |       | Mandatory   | AuthorizationService                |
| UserOfService   |       | Mandatory   | AdminDomain representing the Fabric |

### 9.8.8 CIM\_ServiceAffectsElement (ManagedElement to Service)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 87 describes class CIM\_ServiceAffectsElement (ManagedElement to Service).

**Table 87: SMI Referenced Properties/Methods for CIM\_ServiceAffectsElement (ManagedElement to Service)**

| Properties       | Flags | Requirement | Description & Notes                 |
|------------------|-------|-------------|-------------------------------------|
| AffectingElement |       | Mandatory   | AuthorizationService                |
| AffectedElement  |       | Mandatory   | AdminDomain representing the Switch |

#### 9.8.9 CIM\_ServiceAffectsElement (StorageHardwareID to Service)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 88 describes class CIM\_ServiceAffectsElement (StorageHardwareID to Service).

**Table 88: SMI Referenced Properties/Methods for CIM\_ServiceAffectsElement (StorageHardwareID to Service)**

| Properties       | Flags | Requirement | Description & Notes                 |
|------------------|-------|-------------|-------------------------------------|
| AffectingElement |       | Mandatory   | AuthorizationService                |
| AffectedElement  |       | Mandatory   | AdminDomain representing the Switch |

## EXPERIMENTAL

---



---

---

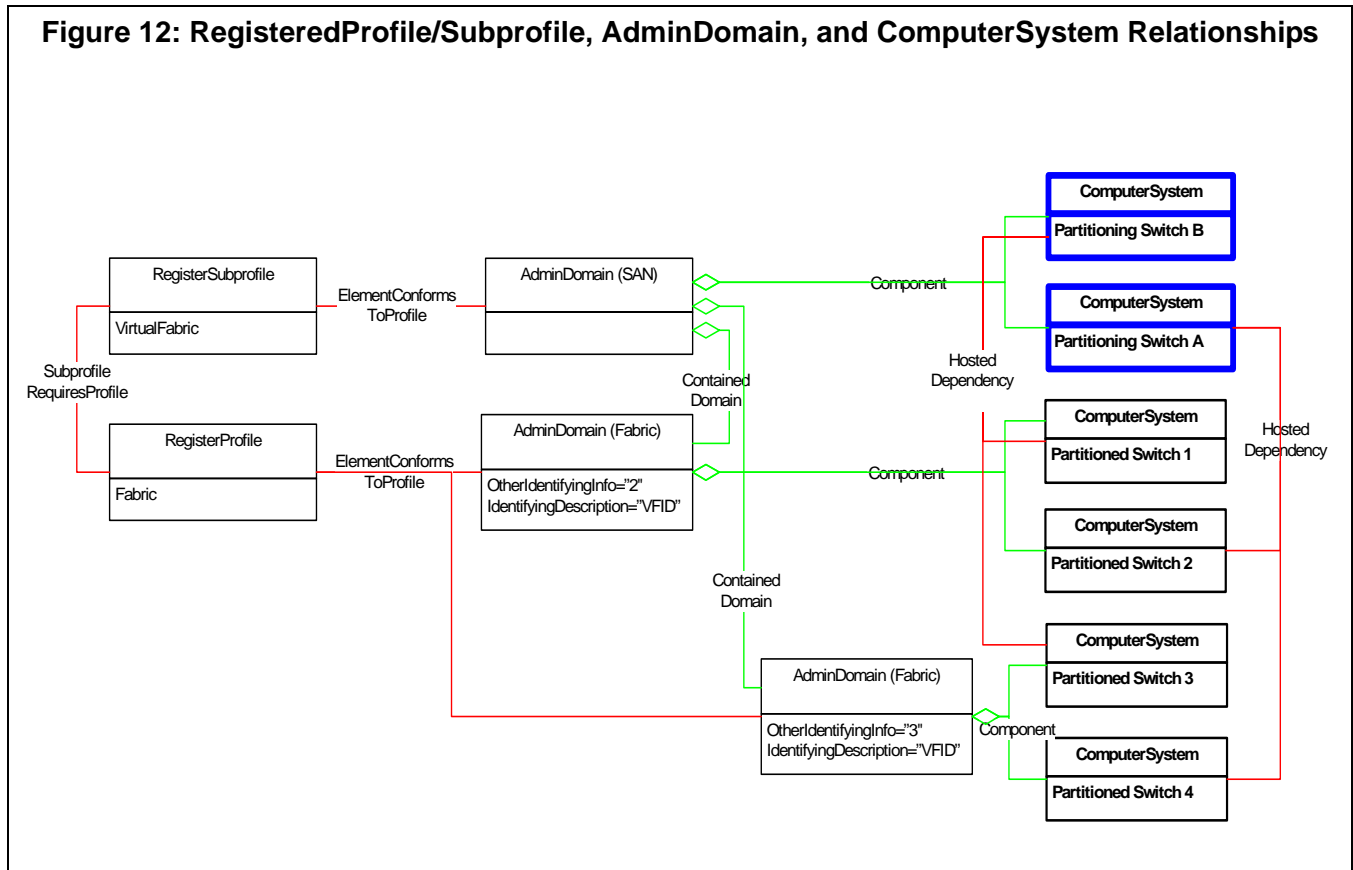


---

**EXPERIMENTAL**
**Clause 10: Virtual Fabrics Subprofile**
**10.1 Description**

Technologies have recently been implemented for Fibre Channel SANs to logically separate the hardware into multiple fabrics yet keep them physically interconnected. The term for this technology defined by ANSI T11 is "Virtual Fabrics". ANSI T11 calls the underlying hardware "Core Switches". To be consistent with a more broader use model and DMTF schematics, this subprofile will call these the partitioning systems. ANSI T11 calls the switching construct that resides in the partitioning system, a virtual switch. Again for the broader use model and consistency with the DMTF, this system will be call the partitioned system. The Fabric Profile already provides the discovery of the "virtual fabrics" and the "virtual switches". This subprofile allows for the discovery of the underlying partitioning system. The Switch Partitioning subprofile provides the mechanism for configuring the partitioning system.

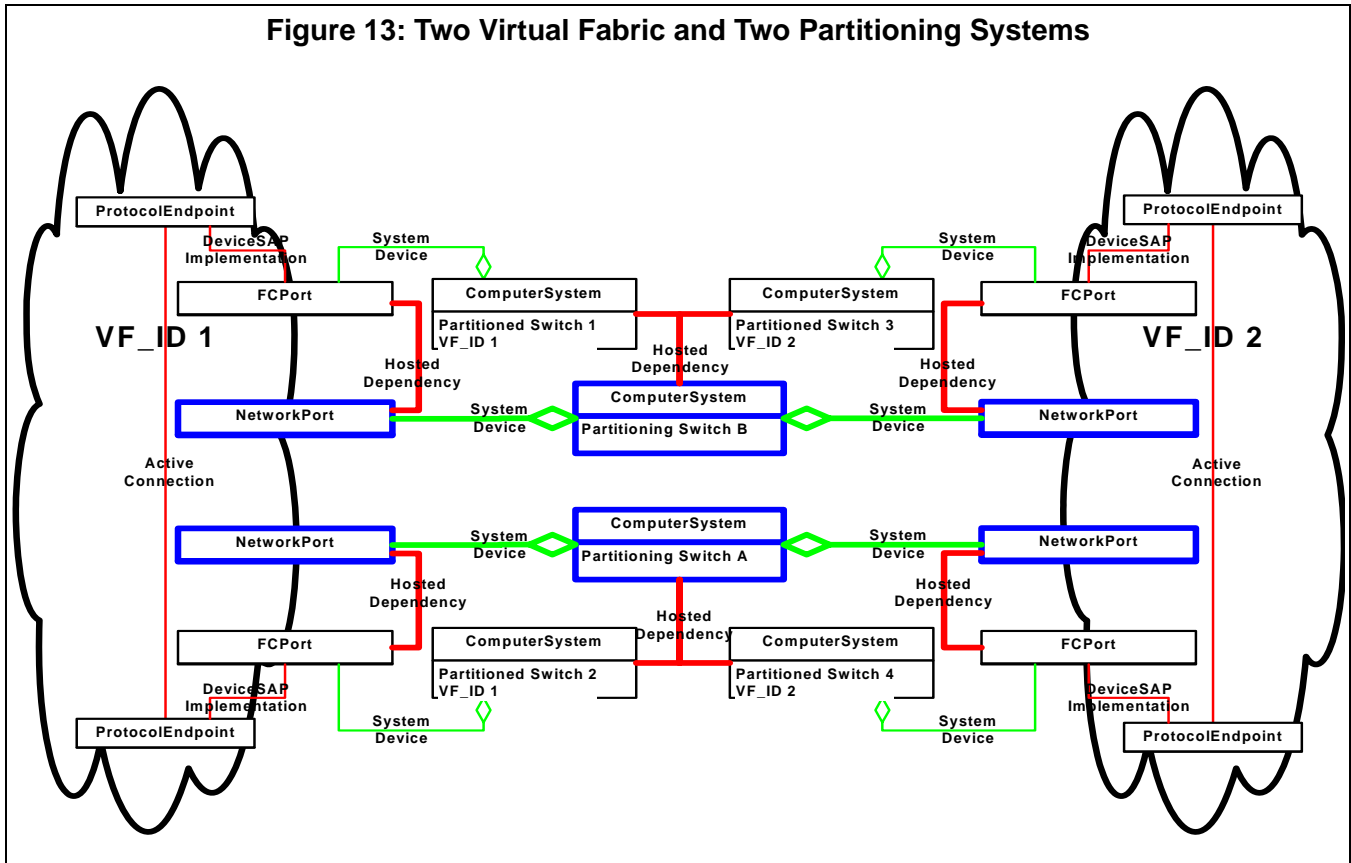
All Virtual Fabrics shall be associated to the same SAN for the same underlying "physical" topology. Also the Registered Subprofile associates to the SAN object to allow the client to arrive at the proper SAN object without traversing upwards from the fabric instance. In Figure 12, the relationship is shown. Note also that the partitioning ComputerSystems are associated with the SAN.

**Figure 12: RegisteredProfile/Subprofile, AdminDomain, and ComputerSystem Relationships**


In Figure 13, the partitioning systems are shown presenting two "virtual fabric" with only one link (ActiveConnection) in each fabric. See Clause 4: Fabric Profile for more information with regards to fabric topology

using ActiveConnection. In this configuration each fabric, virtual switch, and port are uniquely identified. Note that the new instances to support this subprofile are outlined in a bolded line.

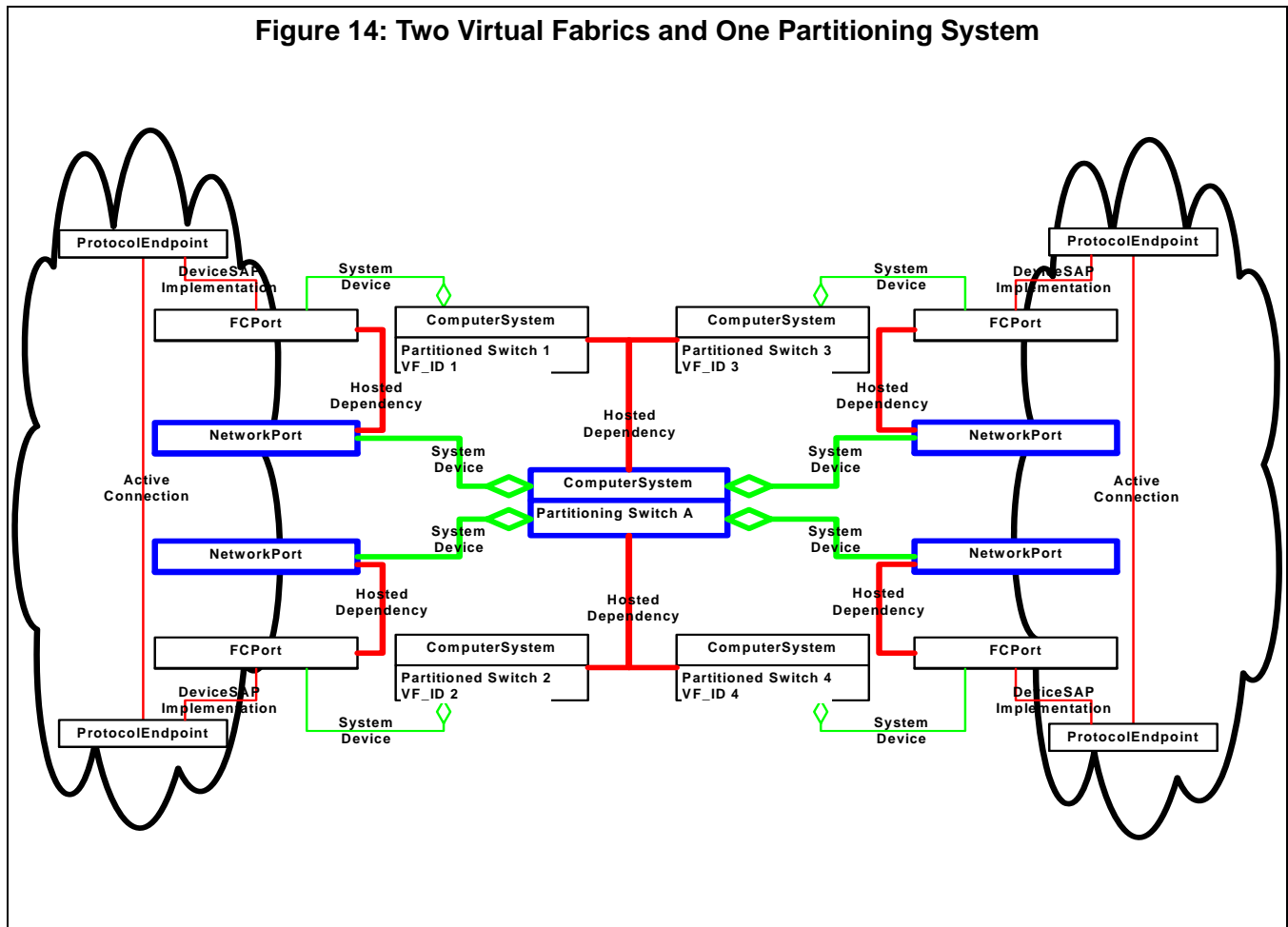
**Figure 13: Two Virtual Fabric and Two Partitioning Systems**





In Figure 13, a single partitioning system is creating two virtual fabrics and in Figure 14, the system is partitioning

**Figure 14: Two Virtual Fabrics and One Partitioning System**



the network port. When the NetworkPort is partitioned, each port is not guaranteed to be uniquely identified (though the key to the instance is) and the client needs to map it to the fabric it is participating in to maintain the ports identity.

In all cases you can have a one to one or many to one relationships between partitioned switches and the partitioning switch. The same is true between the partitioned FCPorts and the NetworkPorts.

To determine the end to end path of a fabric, the client follows the rules outlined in the fabric profile with the exception that when entering a switch, if there exists a hosted dependency on the switch port, the partitioning system shall be traversed instead of the partitioned system. So in Figure 14, when the client traverses across ActiveConnection to the ProtocolEndpoint to the FCPort and down the HostedDependency, it realizes that both ends of the link are actually in the same partitioned system.

## 10.2 Health and Fault Management Consideration

The possible Health and Fault information through LifeCycle indications are addressed in the Fabric Profile.

## 10.3 Cascading Considerations

None

## 10.4 Supported Profiles, Subprofiles, and Packages

Not defined in this standard.

## 10.5 Methods of the Profile

None

## 10.6 Client Considerations and Recipes

None

## 10.7 Registered Name and Version

FabricVirtualFabrics version 1.2.0

## 10.8 CIM Elements

**Table 89: CIM Elements for FabricVirtualFabrics**

| Element Name  | Requirement | Description   |
|---|-------------|---|
| CIM_AdminDomain (SAN) (10.8.1)                                    | Mandatory   | AdminDomain representing the SAN  |
| SNIA_AdminDomain (Fabric) (10.8.2)                                | Mandatory   | AdminDomain representing the Fabric.  |
| CIM_HostedDependency (NetworkPort to FCPort) (10.8.3)             | Mandatory   | Association between NetworkPort to FCPort.  |
| CIM_HostedDependency (Partitioning CS to Partitioned CS) (10.8.4) | Mandatory   | Association between the Partitioning ComputerSystem and Partitioned ComputerSystem.   |
| CIM_ComputerSystem (Partitioning) (10.8.5)                        | Mandatory   | The partitioning ComputerSystem.  |
| CIM_ComputerSystem (Partitioned) (10.8.6)                         | Mandatory   | The partitioned ComputerSystem.   |
| CIM_SystemDevice (NetworkPort to ComputerSystem) (10.8.7)         | Mandatory   | Associates the partitioning classes (NetworkPort to the ComputerSystem)   |
| CIM_Component (AdminDomain to Partitioning CS) (10.8.8)           | Mandatory   | Associates the partitioning ComputerSystems representing the underlying physical switches to the AdminDomain representing the SAN |
| CIM_NetworkPort (10.8.9)  | Mandatory   | The partitioning port.  |

### 10.8.1 CIM\_AdminDomain (SAN)

AdminDomain representing the SAN

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 90 describes class CIM\_AdminDomain (SAN).

**Table 90: SMI Referenced Properties/Methods for CIM\_AdminDomain (SAN)**

| Properties        | Flags | Requirement | Description & Notes                          |
|-------------------|-------|-------------|--|
| CreationClassName |       | Mandatory   | Name of Class                                |
| Name              |       | Mandatory   | An arbitrary name (implementation dependent) |
| NameFormat        |       | Mandatory   | Dependent on the arbitrary name chosen.      |

### 10.8.2 SNIA\_AdminDomain (Fabric)

AdminDomain representing the fabric. This is a logical entity and can represent virtual fabrics.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 91 describes class SNIA\_AdminDomain (Fabric).

**Table 91: SMI Referenced Properties/Methods for SNIA\_AdminDomain (Fabric)**

| Properties              | Flags | Requirement | Description & Notes  |
|-------------------------|-------|-------------|--|
| CreationClassName       |       | Mandatory   | Name of Class  |
| Name                    | C     | Mandatory   | WWN of Fabric  |
| NameFormat              |       | Mandatory   | 'WWN'  |
| OtherIdentifyingInfo    | C     | Mandatory   | Virtual Fabric ID  |
| IdentifyingDescriptions |       | Mandatory   | 'VF_ID' is placed into corresponding index of OtherIdentifyingInfo |

### 10.8.3 CIM\_HostedDependency (NetworkPort to FCPort)

Association between NetworkPort to FCPort

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 92 describes class CIM\_HostedDependency (NetworkPort to FCPort).

**Table 92: SMI Referenced Properties/Methods for CIM\_HostedDependency (NetworkPort to FCPort)**

| Properties | Flags | Requirement | Description & Notes                              |
|------------|-------|-------------|--|
| Antecedent |       | Mandatory   | NetworkPort representing the Partitioning System |
| Dependent  |       | Mandatory   | CFCPort representing the Partitioned System      |

#### 10.8.4 CIM\_HostedDependency (Partitioning CS to Partitioned CS)

Association between the Partitioning ComputerSystem and Partitioned ComputerSystem

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 93 describes class CIM\_HostedDependency (Partitioning CS to Partitioned CS).

**Table 93: SMI Referenced Properties/Methods for CIM\_HostedDependency (Partitioning CS to Partitioned CS)**

| Properties | Flags | Requirement | Description & Notes                                 |
|------------|-------|-------------|---|
| Antecedent |       | Mandatory   | ComputerSystem representing the Partitioning System |
| Dependent  |       | Mandatory   | ComputerSystem representing the Partitioned System  |

#### 10.8.5 CIM\_ComputerSystem (Partitioning)

The ComputerSystem that "hosts" the partitioned ComputerSystems.

Created By: External

Modified By: Static

Deleted By: External

Class Mandatory:

Table 94 describes class CIM\_ComputerSystem (Partitioning).

**Table 94: SMI Referenced Properties/Methods for CIM\_ComputerSystem (Partitioning)**

| Properties        | Flags | Requirement | Description & Notes |
|-------------------|-------|-------------|---------------------|
| CreationClassName |       | Mandatory   |                     |
| Name              |       | Mandatory   |                     |
| Dedicated         |       | Mandatory   |                     |

### 10.8.6 CIM\_ComputerSystem (Partitioned)

The ComputerSystem representing the Interconnect Element (e.g. a switch) or Platform (e.g. Host and Array). This System will always be the Dependent in the association HostedDependency. Traversing this association will place you at the System that is being partitioned.

Created By: External

Modified By: Static

Deleted By: External

Class Mandatory:

Table 95 describes class CIM\_ComputerSystem (Partitioned).

**Table 95: SMI Referenced Properties/Methods for CIM\_ComputerSystem (Partitioned)**

| Properties              | Flags | Requirement | Description & Notes  |
|-------------------------|-------|-------------|--|
| CreationClassName       |       | Mandatory   |  |
| Name                    |       | Mandatory   |  |
| Dedicated               |       | Mandatory   |  |
| OtherIdentifyingInfo    | C     | Mandatory   | Virtual Fabric ID  |
| IdentifyingDescriptions |       | Mandatory   | 'VF_ID' is placed into corresponding index of OtherIdentifyingInfo |

### 10.8.7 CIM\_SystemDevice (NetworkPort to ComputerSystem)

Associates the NetworkPort to the ComputerSystem

Created By: External

Modified By: Static

Deleted By: External

Class Mandatory:

Table 96 describes class CIM\_SystemDevice (NetworkPort to ComputerSystem).

**Table 96: SMI Referenced Properties/Methods for CIM\_SystemDevice (NetworkPort to ComputerSystem)**

| Properties     | Flags | Requirement | Description & Notes                              |
|----------------|-------|-------------|--|
| GroupComponent |       | Mandatory   | The reference to the partitioning ComputerSystem |
| PartComponent  |       | Mandatory   | The reference to the NetworkPort                 |

### 10.8.8 CIM\_Component (AdminDomain to Partitioning CS)

Associates the partitioning ComputerSystems representing the underlying physical switches to the AdminDomain representing the SAN. This allows the client another means to determine which ComputerSystems are part of the

Fabric versus those that are part of the underlying physical topology. ComputerSystems representing Fibre Channel switches are associated to the AdminDomain representing the fabric (see the Fabric Profile).

Created By: External

Modified By: Static

Deleted By: External

Class Mandatory:

Table 97 describes class CIM\_Component (AdminDoman to Partitioning CS).

**Table 97: SMI Referenced Properties/Methods for CIM\_Component (AdminDoman to Partitioning CS)**

| Properties     | Flags | Requirement | Description & Notes                                   |
|----------------|-------|-------------|---|
| GroupComponent |       | Mandatory   | The reference to the AdminDomain representing the SAN |
| PartComponent  |       | Mandatory   | The reference to the partitioning ComputerSystem      |

#### 10.8.9 CIM\_NetworkPort

The partitioning port.

Created By: External

Modified By: Static

Deleted By: External

Class Mandatory:

Table 98 describes class CIM\_NetworkPort.

**Table 98: SMI Referenced Properties/Methods for CIM\_NetworkPort**

| Properties              | Flags | Requirement | Description & Notes                     |
|-------------------------|-------|-------------|---|
| SystemCreationClassName |       | Mandatory   | The scoping System's CreationClassName. |
| SystemName              |       | Mandatory   | The scoping System's Name.              |
| CreationClassName       |       | Mandatory   | Name of Class                           |
| DeviceID                |       | Mandatory   | Opaque                                  |

---

## EXPERIMENTAL

---

---

---

**STABLE****Clause 11: Switch Profile****11.1 Description**

The switch profile models logical and physical aspects of a Fibre Channel switch. The ComputerSystem class constitutes the center of the switch model (and is the top level object which the profile registration points to). An instance of a ComputerSystem is identified as a switch by the property Dedicated set to "switch".

This profile includes discovery components including ports, port statistics, product information, software, and chassis information. It also includes configuration of the switch including switch and port state change, port speed, switch name, symbolic names, and DomainID.

Both the Switch and Port have a capabilities class, FCSwitchCapabilities and FCPortCapabilities, respectively, defining which configuration options are supported by the switch. The capabilities define what components are configurable and any restrictions that apply. Except for state change, an associated settings class is defined for both the switch and port, FCSwitchSettings and FCPortSettings, which the client uses to request configuration changes to the Switch or Port, respectively. A setting does not necessarily result in a change to the underlying Switch or Port. The client can determine whether the setting was applied by looking at the associated property in the Switch or Port class.

The model for configuration is made up of three components, capabilities, settings, and the ManagedElements, ComputerSystem and FCPort. The capabilities define what components are configurable and any restrictions that apply, the settings define what the client requests, and the ManagedElements expose the actual changes that were applied.

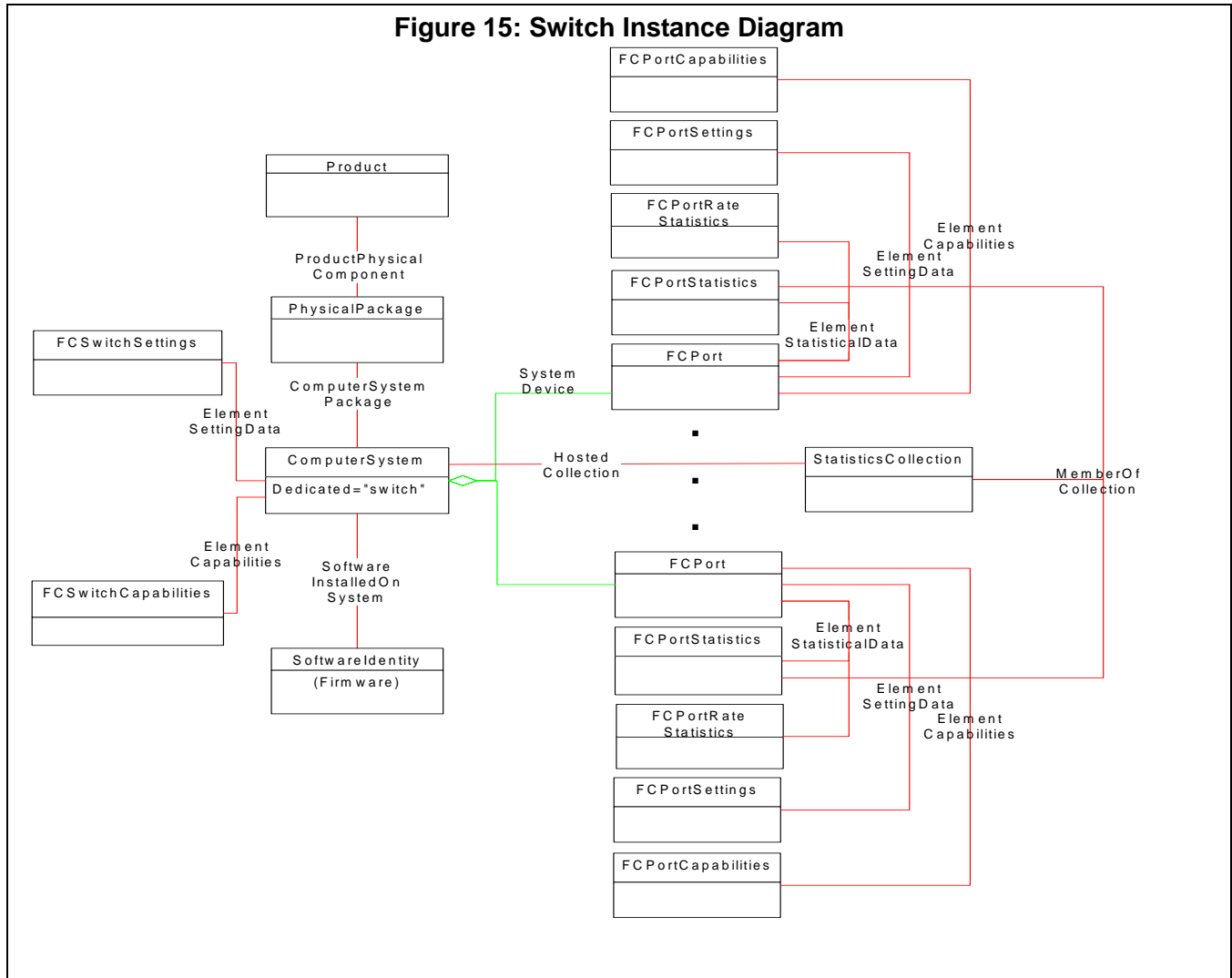
The ComputerSystem (Dedicated as Switch) and FCPort classes have the method RequestStateChange() for requesting that the state be changed and an associated property RequestedState on the classes which indicates the current state change that has been requested. To determine whether the state change has completed, the property EnabledState can be examined to determine whether the device has completed the state change.

If a switch is modular, for instance if the switch is comprised of multiple blades on a backplane, LogicalModule can optionally be used to model each sub-module, and as an aggregation point for the switch ports. This is described in the Blade Subprofile.

FCPort describes the logical aspects of the port link and the data layers. PhysicalConnector models the physical aspects of a port. An instance of the FCPortStatistics class is expected for each instance of the FCPort class. FCPortStatistics expose real time port health and traffic information.

If the instrumentation is embedded in a switch, it shall provide a switch profile implementation for the hosting switch, and it may proxy a switch profile implementation for other switches reported in the Fabric Profile.

**Figure 15: Switch Instance Diagram**



### 11.1.1 FC Port Settings and Capabilities

Capabilities describe the possible features that a ManagedElement supports. Settings are used to describe the requested configuration. The ManagedElement itself describes what settings have been applied and operating.

- For FC Port Type, there are settings that are not in the actual ManagedElement, FCPort.Types. These are settings that allow a subrange of possible port types. They are:
- A G\_Port is a Switch Port that is capable of either operating as an E\_Port or F\_Port. A G\_Port determines through Port Initialization whether it operates as an E\_Port or as an F\_Port.
- A GL\_Port is a G\_Port that is also capable of operating as an FL\_Port.
- A Fx\_Port is a switch port capable of operating as an F\_Port or FL\_Port.

The actual FCPort when operating shall run one of the port types as per FC-GS. In most cases a switch has a default setting to autonegotiate, which in most cases equates to GL or G being set in FCPortSetting.RequestedType. It is required that this setting, FCPortSetting.RequestedType, be shown regardless of whether it was set administratively or is the default behavior of the switch. FCPortSetting.RequestedType



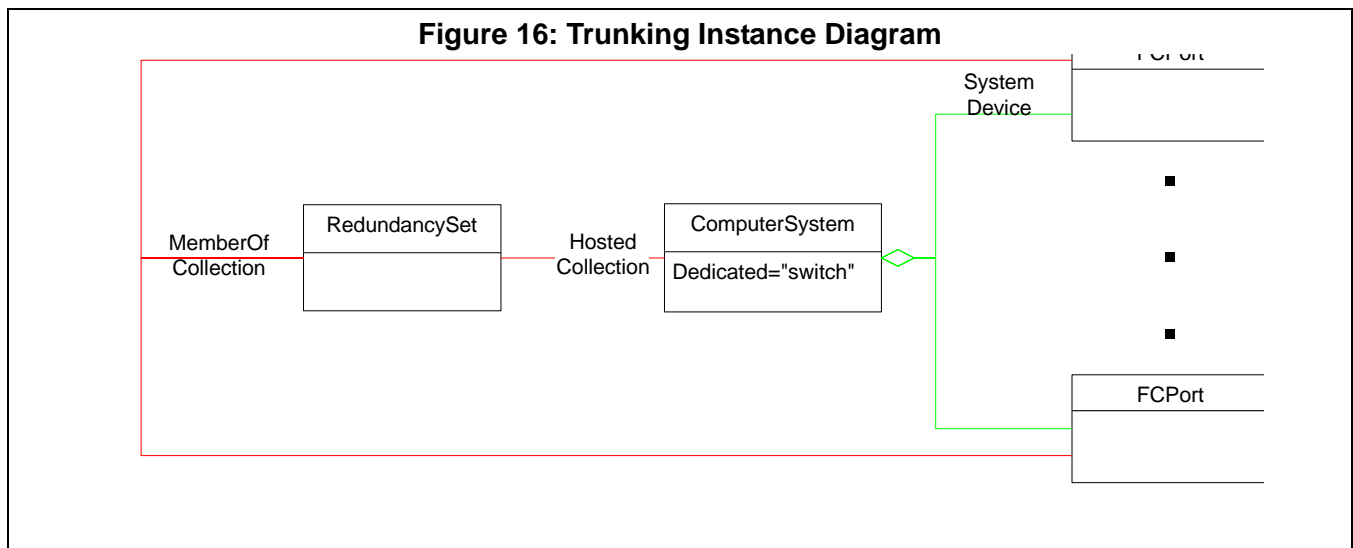
represents a setting that the administrator can understand and clearly identify why a switch port ends up running a particular port type. If the switch does not support setting the port type, the RequestedTypesSupported array will be empty. It is valid to have a port type of "Unknown" until the link has been established. In 11.8.14 CIM\_FCPortStatistics, the table is showing the relationship of particular port types and how the capabilities relate to the setting for these port types.

The same concept applies for FCPort settings for speed except there is a separate property indicating auto negotiate, FCPortSettings.AutoSenseSpeed (LogicalPortSettings.AutoSenseSpeed). Note that this setting may have been previously set through some other administrative interface (e.g. CLI), but should still be reported in FCPortSettings.RequestedSpeed. If FCPortSetting.AutoSenseSpeed is true, then the value of FCPortSettings.RequestedSpeed is ignored and the speed will be negotiated by the hardware. If it is disabled, the port will operate at the speed configured in FCPortSettings.RequestedSpeed.

FCPortSettings.RequestedSpeed allows the port speed to be administratively set (WRITE qualifier). It also indicates to the client that the port has been administratively set (now or at a previous time). This property can only be set administratively if FCPortCapabilities.RequestedSpeedsSupported[] is not empty, and may only be set to one of the values in FCPortCapabilities.RequestedSpeedsSupported[].

FCPortCapabilities.RequestedSpeedsSupported indicates whether the device allows the speed to be administratively set. For instance, a 4Gb port may allow 1, 2, and 4 Gb. FCPort.Speed (LogicalPort.Speed) represents the actual speed the port is running and a speed of zero represents that the link has not been established.

### 11.1.2 Trunking



Trunking describes from a switch perspective which ports are working together passing frames using the class RedundancySet. The RedundancySet has a property TypeOfSet which is used to identify what type of redundancy or trunking is occurring among the switch ports associated to the RedundancySet using MemberOfCollection.

## 11.2 Health and Fault Management

The following classes report possible Health and Fault information through LifeCycle indications:

- ComputerSystem
- FCPort

These LifeCycle indications are more fully described in *Storage Management Technical Specification, Part 2 Common Profiles* Table 280.

Also in Table 100: CIM Elements for Switch are a list of AlertIndications which may also be indicators for Health and Fault Management.

### 11.3 Cascading Considerations

None

### 11.4 Dependencies on Profiles, Subprofiles, and Packages

**Table 99: Supported Profiles for Switch**

| Registered Profile Names   | Mandatory | Version |
|----------------------------|-----------|---------|
| Blades                     | No        | 1.2.0   |
| Access Points              | No        | 1.2.0   |
| Software Installation      | No        | 1.2.0   |
| Multiple Computer System   | No        | 1.2.0   |
| Switch Configuration Data  | No        | 1.2.0   |
| Physical Package           | Yes       | 1.2.0   |
| Software                   | Yes       | 1.2.0   |
| Fabric Switch Partitioning | No        | 1.2.0   |
| Power Supply               | No        | 1.0.0   |
| Fan                        | No        | 1.0.0   |
| RecordLog                  | No        | 1.0.0   |

### 11.5 Methods of this Profile

None in this version of the specification

### 11.6 Client Considerations and Recipes

#### 11.6.1 Enable FCPort

```
// DESCRIPTION
// This recipe describes how to enable a port on a Fibre Channel Switch.
//
// PRE-EXISTING CONDITIONS AND ASSUMPTIONS
// 1. The instance of the port to be enabled is known as $Port.
```

## Switch Profile

```
// MAIN
// Step 1. Retrieve the capabilities of the port.
$PortCapabilities[] = Associators($Port.getObjectPath(),
    "CIM_ElementCapabilities",
    "CIM_FCPortCapabilities",
    "ManagedElement",
    "Capabilities",
    false,
    false,
    {"RequestedStatesSupported"})
if ($PortCapabilities[] == null || PortCapabilities[].length != 1) {
    <ERROR! The required port capabilities are not available>
}

// Step 2. Verify that the port can be enabled.
if (!contains(2, $PortCapabilities[0].RequestedStatesSupported)) {
    <EXIT! Enabling the specified port is not supported>
}

// Step 3. Verify that the port is in a state in which enabling is appropriate.
if ($Port.EnabledState != 2 && $Port.RequestedState == 5) {

    // Step 4. Enable the port.
    %InArguments["RequestedState"] = 2// "Enabled"
    // Timeout request after 90 seconds
    %InArguments["TimeoutPeriod"] = 00000000000130.000000:000
    #ReturnValue = InvokeMethod($Port.getObjectPath(),
        "RequestStateChange",
        %InArguments,
        %OutArguments)
    if (#ReturnValue == 0) {// "Completed with No Error"
        <EXIT! Port successfully enabled>
    } else if (#ReturnValue == 4098) {// "Timeout Parameter Not Supported"
        %InArguments["RequestedState"] = 2// "Enabled"
        %InArguments["TimeoutPeriod"] = 0// No timeout
        #ReturnValue = InvokeMethod($Port.getObjectPath(),
            "RequestStateChange",
            %InArguments,
            %OutArguments)
        if (#ReturnValue == 0) {// "Completed with No Error"
            <EXIT! Port successfully enabled>
        } else {
            <ERROR! Port state transition failed>
        }
    }
} else {
    <ERROR! The specified port is already enabled or currently in a
```

```

        state transition>
    }

```

## 11.6.2 Disable Port

```

// DESCRIPTION
// This recipe describes how to disable a port on a Fibre Channel Switch.
//
// PRE-EXISTING CONDITIONS AND ASSUMPTIONS
// 1. The instance of the port to be disabled is known as $Port.

// MAIN
// Step 1. Retrieve the capabilities of the port.
$PortCapabilities[] = Associators($Port.getObjectPath(),
    "CIM_ElementCapabilities",
    "CIM_FCPortCapabilities",
    "ManagedElement",
    "Capabilities",
    false,
    false,
    {"RequestedStatesSupported"})
if ($PortCapabilities[] == null || PortCapabilities[].length != 1) {
    <ERROR! The required port capabilities are not available>
}

// Step 2. Verify that the port can be disabled.
if (!contains(3, $Capabilities.RequestedStatesSupported)) {
    <EXIT! Disabling the specified port is not supported>
}

// Step 3. Verify that the port is in a state in which disabling is appropriate.
if ($Port.EnabledState != 3 && $Port.RequestedState == 5) {

    // Step 4. Disable the port.
    %InArguments["RequestedState"] = 3// "Disabled"
    // Timeout request after 90 seconds
    %InArguments["TimeoutPeriod"] = 00000000000130.000000:000
    #ReturnValue = InvokeMethod($Port.getObjectPath(),
        "RequestStateChange",
        %InArguments,
        %OutArguments)
    if (#ReturnValue == 0) {// "Completed with No Error"
        <EXIT! Port successfully disabled>
    } else if (#ReturnValue == 4098) {// "Timeout Parameter Not Supported"
        %InArguments["RequestedState"] = 3 // "Disabled"
        %InArguments["TimeoutPeriod"] = 0 // No timeout
        #ReturnValue = InvokeMethod($Port.getObjectPath(),
            "RequestStateChange",

```

```

        %InArguments,
        %OutArguments)
    if (#ReturnValue == 0) { // "Completed with No Error"
        <EXIT! Port successfully disabled>
    } else {
        <ERROR! Port state transition failed>
    }
}
} else {
    <ERROR! The specified port is already disabled or currently in a
    state transition>
}

```

### 11.6.3 Enable Switch

```

// DESCRIPTION
// This recipe describes how to enable a Fibre Channel Switch.
//
// PRE-EXISTING CONDITIONS AND ASSUMPTIONS
// 1. A reference to the Switch to enable is known and defined in the
//    variable $Switch->.

// MAIN
// Step 1. Retrieve the relevant Switch instance information.
$Switch = GetInstance($Switch->,
    false,
    false,
    false,
    {"EnabledState", "RequestedState"})

// Step 2. Retrieve the capabilities of the Switch.
$SwitchCapabilities[] = Associators($Switch->,
    "CIM_ElementCapabilities",
    "CIM_FCSwitchCapabilities",
    "ManagedElement",
    "Capabilities",
    false,
    false,
    {"RequestedStatesSupported"})
if ($SwitchCapabilities[] == null || SwitchCapabilities[].length != 1) {
    <ERROR! The required Switch capabilities are not available>
}

// Step 3. Verify that the Switch can be enabled.
if (!contains(2, $SwitchCapabilities[0].RequestedStatesSupported)) {
    <EXIT! Enabling the specified Switch is not supported>
}

```

## Switch Profile

```
// Step 4. Verify that the Switch is in a state in which enabling is
// appropriate.
if ($Switch.EnabledState != 2 && $Switch.RequestedState == 5) {

    // Step 5. Enable the Switch.
    %InArguments["RequestedState"] = 2// "Enabled"
    // Timeout request after 90 seconds
    %InArguments["TimeoutPeriod"] = 000000000000130.000000:000
    #ReturnValue = InvokeMethod($Switch->,
        "RequestStateChange",
        %InArguments,
        %OutArguments)
    if (#ReturnValue == 0) {// "Completed with No Error"
        <EXIT! Switch successfully enabled>
    } else if (#ReturnValue == 4098) {// "Timeout Parameter Not Supported"
        %InArguments["RequestedState"] = 2// "Enabled"
        %InArguments["TimeoutPeriod"] = 0// No timeout
        #ReturnValue = InvokeMethod($Switch->,
            "RequestStateChange",
            %InArguments,
            %OutArguments)
        if (#ReturnValue == 0) {// "Completed with No Error"
            <EXIT! Switch successfully enabled>
        } else {
            <ERROR! Switch state transition failed>
        }
    }
} else {
    <ERROR! The specified Switch is already enabled or currently in
    a state transition>
}
```

### 11.6.4 Disable Switch

```
// DESCRIPTION
// This recipe describes how to disable a Fibre Channel Switch.
//
// PRE-EXISTING CONDITIONS AND ASSUMPTIONS
// 1. A reference to the Switch to disable is known and defined in the
//     variable $Switch->.
//
// MAIN
// Step 1. Retrieve the relevant Switch instance information.
$Switch = GetInstance($Switch->,
    false,
    false,
    false,
    {"EnabledState", "RequestedState"})
```

```

// Step 2. Retrieve the capabilities of the Switch.
$SwitchCapabilities[] = Associators($Switch->,
    "CIM_ElementCapabilities",
    "CIM_FCSwitchCapabilities",
    "ManagedElement",
    "Capabilities",
    false,
    false,
    {"RequestedStatesSupported"})
if ($SwitchCapabilities[] == null || $SwitchCapabilities[].length != 1) {
    <ERROR! The required Switch capabilities are not available>
}

// Step 3. Verify that the Switch can be disabled.
if (contains(3, $SwitchCapabilities[0].RequestedStatesSupported)) {
    <EXIT! Disabling the specified Switch is not supported>
}

// Step 4. Verify that the Switch is in a state in which disabling is
// appropriate.
if ($Switch.EnabledState != 3 && $Switch.RequestedState == 5) {

    // Step 5. Disable the Switch.
    %InArguments["RequestedState"] = 3// "Disabled"
    // Timeout request after 90 seconds
    %InArguments["TimeoutPeriod"] = 00000000000130.000000:000
    #ReturnValue = InvokeMethod($Switch->,
        "RequestStateChange",
        %InArguments,
        %OutArguments)
    if (#ReturnValue == 0) {// "Completed with No Error"
        <EXIT! Switch successfully disabled>
    } else if (#ReturnValue == 4098) {// "Timeout Parameter Not Supported"
        %InArguments["RequestedState"] = 3// "Disabled"
        %InArguments["TimeoutPeriod"] = 0// No timeout
        #ReturnValue = InvokeMethod($Switch->,
            "RequestStateChange",
            %InArguments,
            %OutArguments)
        if (#ReturnValue == 0) {// "Completed with No Error"
            <EXIT! Switch successfully disabled>
        } else {
            <ERROR! Switch state transition failed>
        }
    } else {
        <ERROR! Switch state transition failed>
    }
}

```

```

    }
} else {
    <ERROR! The specified Switch is already disabled or currently in
        a state transition>
}

```

### 11.6.5 Reset Switch

```

// DESCRIPTION
// This recipe describes how to reset a Fibre Channel Switch.
//
// PRE-EXISTING CONDITIONS AND ASSUMPTIONS
// 1. A reference to the Switch to reset is known and defined in the
//     variable $Switch->.

// MAIN
// Step 1. Retrieve the relevant Switch instance information.
$Switch = GetInstance($Switch->,
    false,
    false,
    false,
    {"EnabledState", "RequestedState"})

// Step 2. Retrieve the capabilities of the Switch.
$SwitchCapabilities[] = Associators($Switch->,
    "CIM_ElementCapabilities",
    "CIM_FCSwitchCapabilities",
    "ManagedElement",
    "Capabilities",
    false,
    false,
    {"RequestedStatesSupported"})
if ($SwitchCapabilities[] == null || $SwitchCapabilities[0].length != 1) {
    <ERROR! The required Switch capabilities are not available>
}

// Step 3. Verify that the Switch can be reset.
if (contains(11, $SwitchCapabilities[0].RequestedStatesSupported)) {
    <EXIT! Resetting the specified Switch is not supported>
}

// Step 4. Verify that the Switch is in a state in which resetting is
// appropriate.
if ($Switch.EnabledState == 2 && $Switch.RequestedState == 5) {

    // Step 5. Reset the Switch.
    %InArguments["RequestedState"] = 11// "Reset"
    // Timeout request after 90 seconds

```



```

%InArguments["TimeoutPeriod"] = 00000000000130.000000:000
#ReturnValue = InvokeMethod($Switch->,
    "RequestStateChange",
    %InArguments,
    %OutArguments)
if (#ReturnValue == 0) { // "Completed with No Error"
    <EXIT! Switch successfully reset>
} else if (#ReturnValue == 4098) { // "Timeout Parameter Not Supported"
    %InArguments["RequestedState"] = 11 // "Reset"
    %InArguments["TimeoutPeriod"] = 0 // No timeout
    #ReturnValue = InvokeMethod($Switch->,
        "RequestStateChange",
        %InArguments,
        %OutArguments)
    if (#ReturnValue == 0) { // "Completed with No Error"
        <EXIT! Switch successfully reset>
    } else {
        <ERROR! Switch state transition failed>
    }
} else {
    <ERROR! Switch state transition failed>
}
} else {
    <ERROR! The specified Switch is already reset or currently in
        a state transition>
}
}

```

### 11.6.6 Set Port Speed

```

// DESCRIPTION
// This recipe describes how to modify the speed of a port on a Fibre Channel
// Switch.
//
// PREEXISTING CONDITIONS AND ASSUMPTIONS
// 1. The instance of the port to whose speed to modify is known as $Port.
// 2. The desired port speed is known and defined in the variable #Speed.

// MAIN
// Step 1. Retrieve the capabilities of the port.
$PortCapabilities[] = Associators($Port.GetObjectPath(),
    "CIM_ElementCapabilities",
    "CIM_FCPortCapabilities",
    "ManagedElement",
    "Capabilities",
    false,
    false,
    {"AutoSenseSpeedConfigurable", "RequestedSpeedsSupported"})
if ($PortCapabilities[] == null || PortCapabilities[].length != 1) {

```

## Switch Profile

```
<ERROR! The required port capabilities are not available>
}
$Capabilities = $PortCapabilities[0]

// Step 2. Verify that the port speed can be set to the specified speed.
if (contains(#Speed, $Capabilities.RequestedSpeedsSupported)) {

    // Step 3. Retrieve the port settings.
    $Settings[] = Associators($Port.GetObjectPath(),
        "CIM_ElementSettingData",
        "CIM_FCPortSettings",
        "ManagedSetting",
        "SettingData",
        false,
        false,
        {"InstanceID", "AutoSenseSpeed", "RequestedSpeed"})
    if ($Settings[] == null || $Settings[].length != 1) {
        <ERROR! The required port settings are not available>
    }
    $PortSetting = $Settings[0]

    // Step 4. Port speed is ignored unless AutoSenseSpeed is disabled,
    if ($PortSetting.AutoSenseSpeed) {
        if ($Capabilities.AutoSenseSpeedConfigurable) {
            $PortSetting.AutoSenseSpeed = false
        } else {
            //Unlikely, but not an error
        }
    }
}

// Step 5. Modify the port speed to the specified speed.
$PortSetting.RequestedSpeed = #Speed
ModifyInstance($PortSetting.GetObjectPath(),
    $PortSetting,
    false,
    {"AutoSenseSpeed", "RequestedSpeed"})

// Step 6. Verify that the port speed modification was applied.
$Port = GetInstance($Port.GetObjectPath(),
    false,
    false,
    false,
    {"Speed"})
if ($Port.Speed == #Speed) {
    <EXIT! Port speed modified successfully>
} else {
    <ERROR! Port speed was not modified as specified>
}
```

```

    }
  } else {
    <EXIT! Specified port speed is not supported>
  }
}

```

### 11.6.7 Set Port Type

```

// DESCRIPTION
// This recipe describes how to modify the port type on a Fibre Channel Switch.
//
// PREEXISTING CONDITIONS AND ASSUMPTIONS
// 1. The instance of the port to whose type to modify is known as $Port.
// 2. The desired port type is known and defined in the variable #Type.

// MAIN
// Step 1. Retrieve the capabilities of the port.
$PortCapabilities[] = Associators($Port.getObjectPath(),
    "CIM_ElementCapabilities",
    "CIM_FCPortCapabilities",
    "ManagedElement",
    "Capabilities",
    false,
    false,
    {"RequestedTypesSupported"})
if ($PortCapabilities[] == null || PortCapabilities[].length != 1) {
    <ERROR! The required port capabilities are not available>
}

// Step 2. Verify that the port type can be modified as specified.
$Capabilities = $PortCapabilities[0]
if (contains(#Type, $Capabilities.RequestedTypesSupported)) {

    // Step 3. Retrieve the port settings.
    $Settings[] = Associators($Port.getObjectPath(),
        "CIM_ElementSettingData",
        "CIM_FCPortSettings",
        "ManagedSetting",
        "SettingData",
        false,
        false,
        {"RequestedType"})
    if ($Settings[] == null || Settings[].length != 1) {
        <ERROR! The required port settings are not available>
    }
    $PortSetting = $Settings[0]

    // Step 4. Modify the port type to the specified type.

```

## Switch Profile

```
$PortSetting.RequestedType = #Type
ModifyInstance($PortSetting.getObjectPath(),
    $PortSetting,
    false,
    {"RequestedType"})

// Step 5. Verify that the port type modification was applied.
$Port = GetInstance($PortSetting.getObjectPath(),
    false,
    false,
    false,
    {"RequestedType"})
if ($PortSetting.RequestedType == #Type) {
    <EXIT! Port type request successfully>
}
<ERROR! Port type request was not modified as specified>
} else {
    <ERROR! Port type request cannot be set to specified type>
}
```

### 11.6.8 Set Fibre Channel Switch Principal Priority

```
// DESCRIPTION
// This recipe describes how to modify the Principal Priority of a Fibre Channel
// Switch.
//
// PRE-EXISTING CONDITIONS AND ASSUMPTIONS
//
// 1. A reference to the Switch whose Principal Priority to modify is known and
//    defined in the variable $Switch->
// 2. The desired Principal Priority of the Switch is known as #Priority.

// MAIN
// Step 1. Retrieve the capabilities of the Switch.
$SwitchCapabilities[] = Associators($Switch->,
    "CIM_ElementCapabilities",
    "CIM_FCSwitchCapabilities",
    "ManagedElement",
    "Capabilities",
    false,
    false,
    {"PrincipalPrioritiesSupported"})
if ($SwitchCapabilities[] == null || $SwitchCapabilities[].length != 1) {
    <ERROR! The required Switch capabilities are not available>
}

// Step 2. Verify that the Switch Principal Priority can be modified.
$Capabilities = $SwitchCapabilities[0]
```

```

if (!contains(5, $Capabilities.PrincipalPrioritiesSupported[])) {

    $SwitchSettings[] = Associators($Switch->,
        "CIM_ElementSettingData",
        "CIM_FCSwitchSettings",
        "ManagedElement",
        "SettingData",
        false,
        false,
        {"PrincipalPriority"})
    if ($SwitchSettings[] == null || SwitchSettings[].length != 1) {
        <ERROR! Required Switch settings are not available>
    }
    $Settings = $SwitchSettings[0]

    // Step 3. Ensure a new Principal Priority is being set.
    if (#Priority != $Settings.PrincipalPriority)) {

        // Step 4. Modify the Principal Priority of the Switch.
        $Settings.PrincipalPriority = #Priority
        ModifyInstance($Settings.getObjectPath(),
            $Settings,
            false,
            {"PrincipalPriority"})
        // Step 5. Verify that the Switch priority modification was applied.
        $Settings = GetInstance($Settings.getObjectPath(),
            false,
            false,
            false,
            {"PrincipalPriority"})
        if ($Settings.PrincipalPriority == #Priority) {
            <EXIT! Switch Principal Priority was modified successfully>
        }
        <EXIT! Switch Principal Priority was not modified successfully>
    } else {
        <ERROR! Principal Priority specified is already set>
    }
} else {
    // "Not Applicable"
    <EXIT! The Switch does not support Principal Priority modification>
}

```

### 11.6.9 Set Switch Name

```

// DESCRIPTION
// This recipe describes how to modify the name of a Fibre Channel Switch.
//
// PRE-EXISTING CONDITIONS AND ASSUMPTIONS

```

## Switch Profile

```
//
// 1. A reference to the Switch whose name to modify is known and defined in
//     the variable $Switch->
// 2. The desired name of the Switch is known as #Name.

// MAIN
// Step 1. Retrieve the capabilities of the Switch.
$SwitchCapabilities[] = Associators($Switch->,
    "CIM_ElementCapabilities",
    "CIM_FCSwitchCapabilities",
    "ManagedElement",
    "Capabilities",
    false,
    false,
    {"ElementNameEditSupported", "MaxElementNameLen"})
if ($SwitchCapabilities[] == null || $SwitchCapabilities[].length != 1) {
    <ERROR! The required Switch capabilities are not available>
}

// Step 2. Verify that the Switch name can be modified.
$Capabilities = $SwitchCapabilities[0]
if ($Capabilities.ElementNameEditSupported) {

    // Step 3. Verify that the new name to be specified is within the
    // constraints of the name length supported by the Switch.
    if (#Name.length() < $Capabilities.MaxElementNameLen) {

        // Step 4. Retrieve the instance representing the Switch.
        $Switch = GetInstance($Switch->,
            false,
            false,
            false,
            {"ElementName"})

        // Step 5. Modify the name of the Switch.
        $Switch.ElementName = #Name
        ModifyInstance($Switch->,
            $Switch,
            false,
            {"ElementName"})

        // Step 6. Verify that the Switch name change was applied.
        $Switch = GetInstance($Switch->,
            false,
            false,
            false,
            {"ElementName"})
    }
}
```

```

    if (compare(#Name, $Switch.ElementName)) {
        <EXIT! Switch name was modified successfully>
    }
    <ERROR! Switch name was not modified successfully>
}
<ERROR! Specified Switch name exceeds length limit>
} else {
    <EXIT! The Switch does not support name modification>
}

```

### 11.6.10 Set Port Name

```

// DESCRIPTION
// This recipe describes how to modify the name of a Port on a Fibre Channel
// Switch.
//
// PRE-EXISTING CONDITIONS AND ASSUMPTIONS
// 1. The instance of the port to whose type to modify is known as $Port.
// 3. The desired name of the port is known as #Name.

// MAIN
// Step 1. Retrieve the capabilities of the port.
$PortCapabilities[] = Associators($Port.getObjectPath(),
    "CIM_ElementCapabilities",
    "CIM_FCPortCapabilities",
    "ManagedElement",
    "Capabilities",
    false,
    false,
    {"ElementNameEditSupported", "MaxElementNameLen"})
if ($PortCapabilities[] == null || $PortCapabilities[].length != 1) {
    <ERROR! The required Port capabilities are not available>
}

// Step 2. Verify that the port name can be modified.
$Capabilities = $PortCapabilities[0]
if ($Capabilities.ElementNameEditSupported) {

    // Step 3. Verify that the new name to be specified is within the
    // constraints of the name length supported by the port.
    if (#Name.length() < $Capabilities.MaxElementNameLen) {

        // Step 4. Modify the name of the port.
        $Port.ElementName = #Name
        ModifyInstance($Port.getObjectPath(),
            $Port,
            false,
            {"ElementName"})
    }
}

```

```

// Step 5. Verify that the port name change was applied.
$Port = GetInstance($Port.getObjectPath(),
    false,
    false,
    false,
    {"ElementName"})
if (compare(#Name, $Port.ElementName)) {
    <EXIT! Port name was modified successfully>
}
<ERROR! Port name was not modified successfully>
}
<ERROR! Specified Port name exceeds length limit>
} else {
    <EXIT! The Port does not support name modification>
}

```

#### 11.6.11 Set Fibre Channel Switch Preferred Domain ID

```

// DESCRIPTION
//
// This recipe describes how to modify the preferred Domain ID of a Switch.
//
// PRE-EXISTING CONDITIONS AND ASSUMPTIONS
// 1. A reference to the Switch to reset is known and defined in the
//    variable $Switch->.
// 2. The new preferred Domain ID to be set on the Switch is known as #DomainID.

// MAIN
// Step 1. Retrieve the capabilities of the Switch.
$SwitchCapabilities[] = Associators($Switch->,
    "CIM_ElementCapabilities",
    "CIM_FCSwitchCapabilities",
    "ManagedElement",
    "Capabilities",
    false,
    false,
    {"DomainIDConfigureable", "MinDomainID", "MaxDomainID"})
if ($SwitchCapabilities[] == null || $SwitchCapabilities[].length != 1) {
    <ERROR! The required Switch capabilities are not available>
}

// Step 2. Verify that the Switch's preferred Domain ID can be modified.
$Capabilities = $SwitchCapabilities[0]
if ($Capabilities.DomainIDConfigureable) {

    // Step 3. Verify that the desired Domain ID is within the permissible
    // range.

```



## Switch Profile

```
if (#DomainID >= $Capabilities.MinDomainID
    && #DomainID <= $Capabilities.MaxDomainID) {

    // Step 4. Retrieve the Switch settings.
    $Settings[] = Associators($Switch->,
        "CIM_ElementSettingData",
        "CIM_FCSwitchSettings",
        "ManagedSetting",
        "SettingData",
        false,
        false,
        {"PreferredDomainID"})
    if ($Settings[] == null || $Settings[].length != 1) {
        <ERROR! The required Switch settings are not available>
    }
    $SwitchSetting = $Settings[0]

    // Step 5. Modify the Switch Domain ID to the specified preferred value.
    $SwitchSetting.PreferredDomainID = #DomainID
    ModifyInstance($SwitchSetting.GetObjectPath(),
        $SwitchSetting,
        false,
        {"PreferredDomainID"})

    // Step 6. Verify that the Switch Domain ID modification was applied.
    $Switch = GetInstance($Switch->,
        false,
        false,
        false,
        {"IdentifyingDescriptions", "OtherIdentifyingInfo"})
    // NOTE: The Domain ID value is contained in the OtherIdentifyingInfo
    // property at the same index as the "DomainID" element index in the
    // IdentifyingDescriptions property.
    #index = -1
    while (#i < $Switch.IdentifyingDescriptions[].length
        && #index < 0) {
        if ($Switch.IdentifyingDescriptions[#i] == "DomainID") {
            #index = #i
        }
    }
    if (#index >= 0 && $Switch.OtherIdentifyingInfo[#index] == #DomainID) {
        <EXIT! Switch Domain ID successfully modified>
    }
    <ERROR! Switch Domain ID was not modified as specified>
} else {
    <ERROR! Domain ID specified is not within permitted range>
}
```

```

} else {
    <EXIT! Domain ID configuration on the specified Switch is not supported>
}

```

### 11.6.12 Lock Fibre Channel Switch Domain ID

```

// DESCRIPTION
//
// This recipe describes how to set the Domain ID Lock of a Switch.
//
// PRE-EXISTING CONDITIONS AND ASSUMPTIONS
// 1. A reference to the Switch whose Domain ID to lock is known and defined
// in the variable $Switch->.

// MAIN
// Step 1. Retrieve the capabilities of the Switch.
$SwitchCapabilities[] = Associators($Switch->,
    "CIM_ElementCapabilities",
    "CIM_FCSwitchCapabilities",
    "ManagedElement",
    "Capabilities",
    false,
    false,
    {"DomainIDLockedSupported"})
if ($SwitchCapabilities[] == null || $SwitchCapabilities[].length != 1) {
    <ERROR! The required Switch capabilities are not available>
}

// Step 2. Verify that the Switch's Domain ID Lock can be set.
$Capabilities = $SwitchCapabilities[0]
if ($Capabilities.DomainIDLockedSupported) {

    // Step 3. Retrieve the Switch settings.
    $Settings[] = Associators($Switch->,
        "CIM_ElementSettingData",
        "CIM_FCSwitchSettings",
        "ManagedSetting",
        "SettingData",
        false,
        false,
        {"DomainIDLocked", "PreferredDomainID"})
    if ($Settings[] == null || $Settings[].length != 1) {
        <ERROR! The required Switch settings are not available>
    }
    $SwitchSetting = $Settings[0]
    #PreferredDomainID = $SwitchSetting.PreferredDomainID

    // Step 4. Verify that the Domain ID is not already locked.

```

## Switch Profile

```
if ($SwitchSetting.DomainIDLocked) {
    <EXIT! The Domain ID Lock is already set>
}

// Step 5. Lock the Switch Domain ID.
$SwitchSetting.DomainIDLocked = true
ModifyInstance($SwitchSetting.GetObjectPath(),
    $SwitchSetting,
    false,
    {"DomainIDLocked"})

// Step 6. Verify that the Switch Domain ID specifies the preferred
// Domain ID.
$Switch = GetInstance($Switch->,
    false,
    false,
    false,
    {"IdentifyingDescriptions", "OtherIdentifyingInfo"})
// NOTE: The Domain ID value is contained in the OtherIdentifyingInfo
// property at the same index as the "DomainID" element index in the
// IdentifyingDescriptions property.
#index = -1
while (#i < $Switch.IdentifyingDescriptions[0].length && #index < 0) {
    if ($Switch.IdentifyingDescriptions[#i] == "DomainID") {
        #index = #i
    }
}
if (#index >= 0 &&
    $Switch.OtherIdentifyingInfo[#index] == #PreferredDomainID) {
    <EXIT! Switch Domain ID successfully locked>
}
<ERROR! Switch Domain ID does not reflect the preferred Domain ID>
} else {
    <EXIT! Domain ID configuration on the specified Switch is not supported>
}
```

## 11.7 Registered Name and Version

Switch version 1.2.0

## 11.8 CIM Elements

**Table 100: CIM Elements for Switch**

| Element Name  | Requirement | Description  |
|---|-------------|--|
| CIM_ComputerSystemPackage (11.8.1)  | Mandatory   | Associates PhysicalPackage to the ComputerSystem (Switch)  |
| CIM_ComputerSystem (11.8.2)   | Mandatory   | Represents the Switch  |
| CIM_FCSwitchCapabilities (11.8.3)   | Mandatory   | Fibre Channel Switch Capabilities  |
| CIM_FCSwitchSettings (11.8.4)   | Mandatory   | Fibre Channel Switch Settings  |
| CIM_ElementCapabilities (System to FCSwitchCapabilities) (11.8.5)                 | Mandatory   | Associates FCSwitchCapabilities to the ComputerSystem (Switch)   |
| CIM_ElementCapabilities (FCPort to FCPortCapabilities) (11.8.6)                   | Mandatory   | Associates FCPort to the FcPortCapabilities  |
| CIM_ElementSettingData (FCSwitchSettings to ComputerSystem) (11.8.7)              | Conditional | Conditional requirement: Support for the Switch Configuration Data profile. Associates FCSwitchSettings to ComputerSystem (Switch) |
| CIM_ElementSettingData (FCPortSettings to FCPort) (11.8.8)                        | Optional    | Associates FCPortSettings to FcPort  |
| CIM_ElementStatisticalData (FCPortStatistics to FCPort) (11.8.9)                  | Mandatory   | Associates FCPortStatistics to the FCPort  |
| CIM_ElementStatisticalData (FCPortRateStatistics to FCPort) (11.8.10)             | Optional    | Associates FCPortRateStatistics to the FCPort  |
| CIM_FCPort (11.8.11)  | Mandatory   | Fibre Channel Switch Port  |
| CIM_FCPortCapabilities (11.8.12)  | Mandatory   | Switch Port Capabilities   |
| CIM_FCPortSettings (11.8.13)  | Optional    | Switch Port Settings   |
| CIM_FCPortStatistics (11.8.14)  | Mandatory   | Fibre Channel Switch Port Statistics.  |
| CIM_FCPortRateStatistics (11.8.15)  | Optional    | Fibre Channel Switch Port Rate Statistics  |
| CIM_MemberOfCollection (NetworkPortStatistics to StatisticalCollection) (11.8.16) | Mandatory   | Associates the NetworkPortStatistics to the StatisticsCollection.  |
| CIM_MemberOfCollection (FcPort to REdundancySet) (11.8.17)                        | Optional    | Associates the FCPort to the RedundancySet.  |
| CIM_HostedCollection (11.8.18)  | Mandatory   | Associates the Statistics Collection to the Network representing the fabric.   |
| CIM_StatisticsCollection (11.8.19)  | Mandatory   | Collection to aggregate the FCPortStatistics for each switch   |
| CIM_SystemDevice (11.8.20)  | Conditional | Conditional requirement: Support for the Blades profile. Associates FCPort to the ComputerSystem (Switch)                          |

**Table 100: CIM Elements for Switch**

| Element Name  | Requirement | Description  |
|---|-------------|--|
| CIM_ProtocolEndpoint (11.8.21)  | Optional    | The endpoint of a link (ActiveConnection).   |
| CIM_RedundancySet (11.8.22)   | Optional    | The class RedundancySet along with the association MemberOfCollection in this profile is used to show port aggregation for Fibre Channel trunking. |
| SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_ComputerSystem  | Mandatory   | New Switch Instance  |
| SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_ComputerSystem  | Mandatory   | Deletion of Switch Instance  |
| SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ComputerSystem AND SourceInstance.OperationalStatus <> PreviousInstance.OperationalStatus   | Mandatory   | Deprecated WQL - Modification of OperationalStatus in Switch Instance  |
| SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_FCPort AND SourceInstance.OperationalStatus <> PreviousInstance.OperationalStatus   | Mandatory   | Deprecated WQL - Modification of OperationalStatus in FC Port Instance   |
| SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ComputerSystem AND SourceInstance.CIM_ComputerSystem::OperationalStatus <> PreviousInstance.CIM_ComputerSystem::OperationalStatus | Optional    | Experimental CQL - Modification of OperationalStatus in Switch Instance  |
| SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_FCPort AND SourceInstance.CIM_FCPort::OperationalStatus <> PreviousInstance.CIM_FCPort::OperationalStatus                         | Optional    | Experimental CQL - Modification of OperationalStatus in FC Port Instance   |

**11.8.1 CIM\_ComputerSystemPackage**

Associates PhysicalPackage to the ComputerSystem (Switch)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 101 describes class CIM\_ComputerSystemPackage.

**Table 101: SMI Referenced Properties/Methods for CIM\_ComputerSystemPackage**

| Properties | Flags | Requirement | Description & Notes                  |
|------------|-------|-------------|--------------------------------------|
| Antecedent |       | Mandatory   | The reference to the PhysicalPackage |
| Dependent  |       | Mandatory   | The reference to the ComputerSystem  |

### 11.8.2 CIM\_ComputerSystem

Represents the Switch

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 102 describes class CIM\_ComputerSystem.

**Table 102: SMI Referenced Properties/Methods for CIM\_ComputerSystem**

| Properties              | Flags | Requirement | Description & Notes  |
|-------------------------|-------|-------------|--|
| CreationClassName       |       | Mandatory   | Name of Class  |
| Name                    | D     | Mandatory   | Switch Name (WWN)  |
| ElementName             |       | Optional    | User friendly name. Can be set if FCSwitchCapabilities.ElementNameEditSupported for the switch is True.  |
| NameFormat              |       | Mandatory   | Shall be 'WWN'.  |
| OtherIdentifyingInfo    | C     | Optional    | DomainID stored in decimal format, with a value between 0 and 255.   |
| OperationalStatus       |       | Mandatory   |  |
| IdentifyingDescriptions |       | Optional    | 'DomainID' is placed into corresponding index of OtherIdentifyingInfo  |
| Dedicated               |       | Mandatory   | 'Switch'   |
| EnabledState            |       | Optional    | See Table ...  |
| RequestedState          |       | Mandatory   | The Switch state requested via RequestStateChange(). Shall be of the range specified in FCSwitchCapabilities.RequestedStatesSupported if a state change has been requested. Otherwise shall be 'Not Applicable'. |
| EnabledDefault          |       | Optional    | Default startup for the Switch   |
| RequestStateChange()    |       | Optional    |  |

### 11.8.3 CIM\_FCSwitchCapabilities

Fibre Channel Switch Capabilities

#### EXPERIMENTAL |

When the attached HBA supports FC-SB, and the switch is in a cascaded configuration, DomainIDConfigurable and DomainIDLockedSupported shall be TRUE.

#### | EXPERIMENTAL

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 103 describes class CIM\_FCSwitchCapabilities.

**Table 103: SMI Referenced Properties/Methods for CIM\_FCSwitchCapabilities**

| Properties                   | Flags | Requirement | Description & Notes  |
|------------------------------|-------|-------------|--|
| InstanceID                   |       | Mandatory   | Opaque   |
| ElementName                  |       | Mandatory   | Shall be 'FC Switch Capabilities'.   |
| ElementNameEditSupported     |       | Mandatory   | Capability indicating whether ComputerSystem.ElementName for the switch can be set.                        |
| MaxElementNameLength         |       | Mandatory   | Capability specifying the maximum name of ComputerSystem.ElementName for the switch                        |
| RequestedStatesSupported     |       | Mandatory   | The states the switch can support via ComputerSystem.RequestedState.                                       |
| DomainIDConfigurable         |       | Mandatory   | Indicates whether the DomainID setting can be modified.  |
| MinDomainID                  |       | Conditional | Conditional requirement: Support for DomainID configuration. Shall be set if DomainIDConfigurable is true. |
| MaxDomainID                  |       | Conditional | Conditional requirement: Support for DomainID configuration. Shall be set if DomainIDConfigurable is true. |
| DomainIDLockedSupported      |       | Mandatory   |  |
| PrincipalPrioritiesSupported |       | Mandatory   |  |

### 11.8.4 CIM\_FCSwitchSettings

Fibre Channel Switch Settings

#### EXPERIMENTAL |

When the attached HBA supports FC-SB, and the switch is in a cascaded configuration, PreferredDomainID shall be set to a unique value, and DomainIDLocked shall be true.

#### | EXPERIMENTAL

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 104 describes class CIM\_FCSwitchSettings.

**Table 104: SMI Referenced Properties/Methods for CIM\_FCSwitchSettings**

| Properties        | Flags | Requirement | Description & Notes  |
|-------------------|-------|-------------|--|
| InstanceID        |       | Mandatory   | Opaque   |
| ElementName       |       | Mandatory   | Shall be 'FC Switch Settings'.   |
| PreferredDomainID | M     | Conditional | Conditional requirement: Support for DomainID configuration. Required if FCSwitchCapabilities.DomainIDConfigureable is TRUE.                             |
| DomainIDLocked    | M     | Conditional | Conditional requirement: Support for DomainID locking. Required if FCSwitchCapabilities.DomainIDLockSupported is TRUE.                                   |
| PrincipalPriority | M     | Conditional | Conditional requirement: Support for Principal Priorities. Required if FCSwitchCapabilities.PrincipalPrioritiesSupported is not set to 'Not Applicable'. |

#### 11.8.5 CIM\_ElementCapabilities (System to FCSwitchCapabilities)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 105 describes class CIM\_ElementCapabilities (System to FCSwitchCapabilities).

**Table 105: SMI Referenced Properties/Methods for CIM\_ElementCapabilities (System to FCSwitchCapabilities)**

| Properties     | Flags | Requirement | Description & Notes                       |
|----------------|-------|-------------|---|
| ManagedElement |       | Mandatory   | The reference to the ComputerSystem       |
| Capabilities   |       | Mandatory   | The reference to the FCSwitchCapabilities |



### 11.8.6 CIM\_ElementCapabilities (FCPort to FCPortCapabilities)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 106 describes class CIM\_ElementCapabilities (FCPort to FCPortCapabilities).

**Table 106: SMI Referenced Properties/Methods for CIM\_ElementCapabilities (FCPort to FCPortCapabilities)**

| Properties     | Flags | Requirement | Description & Notes                     |
|----------------|-------|-------------|---|
| ManagedElement |       | Mandatory   | The reference to the FcPort             |
| Capabilities   |       | Mandatory   | The reference to the FCPortCapabilities |

### 11.8.7 CIM\_ElementSettingData (FCSwitchSettings to ComputerSystem)

Associates FCSwitchSettings to ComputerSystem (Switch)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: SwitchConfigurationData

Table 107 describes class CIM\_ElementSettingData (FCSwitchSettings to ComputerSystem).

**Table 107: SMI Referenced Properties/Methods for CIM\_ElementSettingData (FCSwitchSettings to ComputerSystem)**

| Properties     | Flags | Requirement | Description & Notes                   |
|----------------|-------|-------------|---------------------------------------|
| SettingData    |       | Mandatory   | The reference to the FCSwitchSettings |
| ManagedElement |       | Mandatory   | The reference to the ComputerSystem   |

### 11.8.8 CIM\_ElementSettingData (FCPortSettings to FCPort)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Optional

Table 108 describes class CIM\_ElementSettingData (FCPortSettings to FCPort).

**Table 108: SMI Referenced Properties/Methods for CIM\_ElementSettingData (FCPortSettings to FCPort)**

| Properties     | Flags | Requirement | Description & Notes                 |
|----------------|-------|-------------|-------------------------------------|
| ManagedElement |       | Mandatory   |                                     |
| SettingData    |       | Mandatory   | The reference to the FCPortSettings |

#### 11.8.9 CIM\_ElementStatisticalData (FCPortStatistics to FCPort)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 109 describes class CIM\_ElementStatisticalData (FCPortStatistics to FCPort).

**Table 109: SMI Referenced Properties/Methods for CIM\_ElementStatisticalData (FCPortStatistics to FCPort)**

| Properties     | Flags | Requirement | Description & Notes                   |
|----------------|-------|-------------|---------------------------------------|
| Stats          |       | Mandatory   | The reference to the FCPortStatistics |
| ManagedElement |       | Mandatory   | The reference to the FCPort           |

#### 11.8.10 CIM\_ElementStatisticalData (FCPortRateStatistics to FCPort)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Optional

Table 110 describes class CIM\_ElementStatisticalData (FCPortRateStatistics to FCPort).

**Table 110: SMI Referenced Properties/Methods for CIM\_ElementStatisticalData (FCPortRateStatistics to FCPort)**

| Properties | Flags | Requirement | Description & Notes                       |
|------------|-------|-------------|---|
| Stats      |       | Mandatory   | The reference to the FCPortRateStatistics |

**Table 110: SMI Referenced Properties/Methods for CIM\_ElementStatisticalData (FCPortRateStatistics to FCPort)**

| Properties     | Flags | Requirement | Description & Notes         |
|----------------|-------|-------------|-----------------------------|
| ManagedElement |       | Mandatory   | The reference to the FCPort |

**11.8.11 CIM\_FCPort**

The Fibre Channel Switch Port.

Created By: Static

Modified By: Extrinsic: RequestStateChange

Deleted By: Static

Class Mandatory: Mandatory

Table 111 describes class CIM\_FCPort.

**Table 111: SMI Referenced Properties/Methods for CIM\_FCPort**

| Properties              | Flags | Requirement | Description & Notes  |
|-------------------------|-------|-------------|--|
| SystemCreationClassName |       | Mandatory   | The scoping System's CreationClassName.  |
| SystemName              |       | Mandatory   | The scoping System's Name.   |
| CreationClassName       |       | Mandatory   | The Class Name   |
| DeviceID                |       | Mandatory   | Opaque   |
| ElementName             |       | Mandatory   | User friendly name. Can be set if FCPortCapabilities.ElementNameEditSupported is True.   |
| OperationalStatus       |       | Mandatory   | See ...  |
| EnabledState            |       | Optional    | See ...  |
| RequestedState          |       | Mandatory   | The port state requested via RequestStateChange(). Shall be of the range specified in FCPortCapabilities.RequestedStatesSupported if a state change has been requested. Otherwise shall be 'Not Applicable'. |
| EnabledDefault          |       | Mandatory   | Default startup for the port. Used in conjunction with RequestedState can allow for persistent disabling of a port.  |

**Table 111: SMI Referenced Properties/Methods for CIM\_FCPort**

| Properties                  | Flags | Requirement | Description & Notes   |
|-----------------------------|-------|-------------|---|
| Speed                       |       | Mandatory   | Speed of zero represents a link not established.<br>1Gb is 1062500000 bps<br>2Gb is 2125000000 bps<br>4Gb is 4250000000 bps<br>10Gb single channel variants are 10518750000 bps<br>10Gb four channel variants are 12750000000 bps<br>This is the raw bit rate.  |
| MaxSpeed                    |       | Mandatory   | The max speed of the Port in Bits per Second using the same algorithm as Speed.   |
| PortType                    |       | Mandatory   | FC-GS Port.Type The specific mode currently enabled for the Port. The values:<br><br>'N' = Node Port<br><br>'NL' = Node Port supporting FC arbitrated loop<br><br>'E' = Expansion Port connecting fabric elements (for example, FC switches)<br><br>'F' = Fabric (element) Port<br><br>'FL' = Fabric (element) Port supporting FC arbitrated loop<br><br>'B' = Bridge Port. PortTypes are defined in the ANSI INCITS FC-GS standards.<br><br>Can be set using FCPortSettings.RequestedType. |
| PortNumber                  |       | Mandatory   | NetworkPorts are often numbered relative to either a logical modules or a network element.  |
| PermanentAddress            |       | Mandatory   | Fibre Channel Port WWN.   |
| LinkTechnology              |       | Mandatory   | 'FC'  |
| RequestStateChange()<br>( ) |       | Optional    | Method to change the port state.<br>FCPortCapabilities.RequestedStatesSupported indicates what states can be set.   |

**11.8.12 CIM\_FCPortCapabilities**

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 112 describes class CIM\_FCPortCapabilities.

**Table 112: SMI Referenced Properties/Methods for CIM\_FCPortCapabilities**

| Properties                 | Flags | Requirement | Description & Notes   |
|----------------------------|-------|-------------|---|
| InstanceID                 |       | Mandatory   | Opaque  |
| ElementName                |       | Mandatory   | Shall be 'FC Port Capabilities'   |
| ElementNameEditSupported   |       | Mandatory   | Indicates whether FCPort.ElementName is settable  |
| MaxElementNameLength       |       | Mandatory   | Indicates the maximum string length of FCPort.ElementName                                   |
| RequestedStatesSupported   |       | Mandatory   | Indicates the supported states for calling FCPort.RequestStateChange().                     |
| RequestedSpeedsSupported   |       | Mandatory   | Indicates the supported speeds that can be set in FCPortSettings.RequestedSpeed             |
| AutoSenseSpeedConfigurable |       | Mandatory   | Indicates whether FCPortSettings.AutoSenseSpeed can be set to auto-negotiate speed.         |
| RequestedTypesSupported    |       | Mandatory   | Indicates the list of supported port types that can be set in FCPortSettings.RequestedType. |

### 11.8.13 CIM\_FCPortSettings

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Optional

Table 113 describes class CIM\_FCPortSettings.

**Table 113: SMI Referenced Properties/Methods for CIM\_FCPortSettings**

| Properties     | Flags | Requirement | Description & Notes  |
|----------------|-------|-------------|--|
| InstanceID     |       | Mandatory   | Opaque   |
| ElementName    |       | Mandatory   | Shall be 'FC Port Settings'  |
| RequestedSpeed | M     | Mandatory   | The requested value to which FCPort.Speed should be set            |
| AutoSenseSpeed | M     | Mandatory   | The request for the FCPort to auto sense the speed (FCPort.Speed). |
| RequestedType  | M     | Mandatory   | The requested setting for the FCPort.PortType.                     |

**11.8.14 CIM\_FCPortStatistics**

Snapshot of performance and error counters for the Fibre Channel Switch.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 114 describes class CIM\_FCPortStatistics.

**Table 114: SMI Referenced Properties/Methods for CIM\_FCPortStatistics**

| Properties                       | Flags | Requirement | Description & Notes   |
|----------------------------------|-------|-------------|---|
| InstanceID                       |       | Mandatory   | Opaque  |
| StatisticTime                    |       | Optional    | The time the statistics were collected. If historical data is instantiated (present), this property shall be set with the time representing the time the statistic was collected. |
| ElementName                      |       | Optional    |   |
| BytesTransmitted                 |       | Mandatory   |   |
| BytesReceived                    |       | Mandatory   |   |
| PacketsTransmitted               |       | Mandatory   |   |
| PacketsReceived                  |       | Mandatory   |   |
| CRCErrors                        |       | Mandatory   |   |
| LinkFailures                     |       | Mandatory   |   |
| PrimitiveSeqProtocol<br>ErrCount |       | Mandatory   |   |
| LIPCount                         |       | Optional    |   |
| NOSCount                         |       | Optional    |   |
| ErrorFrames                      |       | Optional    |   |
| DumpedFrames                     |       | Optional    |   |
| LossOfSignalCounter              |       | Optional    |   |
| LossOfSyncCounter                |       | Optional    |   |
| InvalidTransmission<br>Words     |       | Optional    |   |
| FramesTooShort                   |       | Optional    |   |
| FramesTooLong                    |       | Optional    |   |
| AddressErrors                    |       | Optional    |   |

**Table 114: SMI Referenced Properties/Methods for CIM\_FCPortStatistics**

| Properties                 | Flags | Requirement | Description & Notes |
|----------------------------|-------|-------------|---------------------|
| BufferCreditNotProvided    |       | Optional    |                     |
| BufferCreditNotReceived    |       | Optional    |                     |
| DelimiterErrors            |       | Optional    |                     |
| EncodingDisparityErrors    |       | Optional    |                     |
| LinkResetsReceived         |       | Optional    |                     |
| LinkResetsTransmitted      |       | Optional    |                     |
| MulticastFramesReceived    |       | Optional    |                     |
| MulticastFramesTransmitted |       | Optional    |                     |
| FBSYFrames                 |       | Optional    |                     |
| PBSYFrames                 |       | Optional    |                     |
| FRJTFrames                 |       | Optional    |                     |
| PRJTFrames                 |       | Optional    |                     |
| RXClass1Frames             |       | Optional    |                     |
| TXClass1Frames             |       | Optional    |                     |
| Class1FBSY                 |       | Optional    |                     |
| Class1PBSY                 |       | Optional    |                     |
| Class1FRJT                 |       | Optional    |                     |
| Class1PRJT                 |       | Optional    |                     |
| RXClass2Frames             |       | Optional    |                     |
| TXClass2Frames             |       | Optional    |                     |
| Class2FBSY                 |       | Optional    |                     |
| Class2PBSY                 |       | Optional    |                     |
| Class2FRJT                 |       | Optional    |                     |
| Class2PRJT                 |       | Optional    |                     |
| RXClass3Frames             |       | Optional    |                     |
| TXClass3Frames             |       | Optional    |                     |
| Class3FramesDiscarded      |       | Optional    |                     |

**Table 114: SMI Referenced Properties/Methods for CIM\_FCPortStatistics**

| Properties        | Flags | Requirement | Description & Notes |
|-------------------|-------|-------------|---------------------|
| RxBroadcastFrames |       | Optional    |                     |
| TxBroadcastFrames |       | Optional    |                     |

**11.8.15 CIM\_FCPortRateStatistics**

Fibre Channel Switch Port Rate Statistics represent the rate per second over the SampleInterval. An instance of this class can represent the statistics for the current statistics, archived and consolidated statistics, or both.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Optional

Table 115 describes class CIM\_FCPortRateStatistics.

**Table 115: SMI Referenced Properties/Methods for CIM\_FCPortRateStatistics**

| Properties     | Flags | Requirement | Description & Notes                            |
|----------------|-------|-------------|--|
| InstanceID     |       | Mandatory   | Opaque   |
| StatisticTime  |       | Mandatory   | The time the statistic was collected.          |
| SampleInterval |       | Mandatory   | The interval at which the rates are calculated |
| TxFrameRate    |       | Optional    |  |
| RxFrameRate    |       | Optional    |  |
| MaxTxFrameRate |       | Optional    |  |
| MaxRxFrameRate |       | Optional    |  |
| TxRate         |       | Mandatory   |  |
| RxRate         |       | Mandatory   |  |
| PeakTxRate     |       | Optional    |  |
| PeakRxRate     |       | Optional    |  |

**11.8.16 CIM\_MemberOfCollection (NetworkPortStatistics to StatisticalCollection)**

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory



Table 116 describes class CIM\_MemberOfCollection (NetworkPortStatistics to StatisticalCollection).

**Table 116: SMI Referenced Properties/Methods for CIM\_MemberOfCollection (NetworkPortStatistics to StatisticalCollection)**

| Properties | Flags | Requirement | Description & Notes                       |
|------------|-------|-------------|---|
| Member     |       | Mandatory   | The reference to the FcPortStatistics     |
| Collection |       | Mandatory   | The reference to the StatisticsColelction |

#### 11.8.17 CIM\_MemberOfCollection (FcPort to REdundancySet)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Optional

Table 117 describes class CIM\_MemberOfCollection (FcPort to REdundancySet).

**Table 117: SMI Referenced Properties/Methods for CIM\_MemberOfCollection (FcPort to REdundancySet)**

| Properties | Flags | Requirement | Description & Notes                |
|------------|-------|-------------|------------------------------------|
| Member     |       | Mandatory   | The reference to the FcPort        |
| Collection |       | Mandatory   | The reference to the RedundancySet |

#### 11.8.18 CIM\_HostedCollection

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 118 describes class CIM\_HostedCollection.

**Table 118: SMI Referenced Properties/Methods for CIM\_HostedCollection**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Antecedent |       | Mandatory   |                     |
| Dependent  |       | Mandatory   |                     |

**11.8.19 CIM\_StatisticsCollection**

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 119 describes class CIM\_StatisticsCollection.

**Table 119: SMI Referenced Properties/Methods for CIM\_StatisticsCollection**

| Properties      | Flags | Requirement | Description & Notes |
|-----------------|-------|-------------|---------------------|
| InstanceID      |       | Mandatory   | Opaque              |
| ElementName     |       | Mandatory   |                     |
| SampleInterval  |       | Mandatory   |                     |
| TimeLastSampled |       | Mandatory   |                     |

**11.8.20 CIM\_SystemDevice**

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Blades

Table 120 describes class CIM\_SystemDevice.

**Table 120: SMI Referenced Properties/Methods for CIM\_SystemDevice**

| Properties     | Flags | Requirement | Description & Notes         |
|----------------|-------|-------------|-----------------------------|
| PartComponent  |       | Mandatory   | The reference to the FcPort |
| GroupComponent |       | Mandatory   | The reference to the System |

**11.8.21 CIM\_ProtocolEndpoint**

The endpoint of a link (ActiveConnection). ProtocolEndpoint shall be implemented when BroadcastReset() is supported (Force LIP). It is expected that the Fabric Profile is also implemented which defines the necessary information for determining who will receive the Force LIP on the loop.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Optional

Table 121 describes class CIM\_ProtocolEndpoint.

**Table 121: SMI Referenced Properties/Methods for CIM\_ProtocolEndpoint**

| Properties              | Flags | Requirement | Description & Notes   |
|-------------------------|-------|-------------|---|
| SystemCreationClassName |       | Mandatory   | The scoping System's CreationClassName.   |
| SystemName              |       | Mandatory   | The scoping System's Name.  |
| CreationClassName       |       | Mandatory   | Name of Class   |
| Name                    | CD    | Mandatory   | The Fibre Channel Port WWN.   |
| NameFormat              |       | Mandatory   | 'WWN'   |
| ProtocolIFType          |       | Mandatory   | Shall be 56 (Fibre Channel).  |
| BroadcastResetSupported |       | Mandatory   |   |
| BroadcastReset()        |       | Optional    | Sends a Force LIP to all attached Ports. Required if BroadcastResetSupported is TRUE. |

#### 11.8.22 CIM\_RedundancySet

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Optional

Table 122 describes class CIM\_RedundancySet.

**Table 122: SMI Referenced Properties/Methods for CIM\_RedundancySet**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| InstanceID |       | Mandatory   | Opaque              |
| TypeOfSet  |       | Mandatory   |                     |

**STABLE**

---



---



---

## EXPERIMENTAL

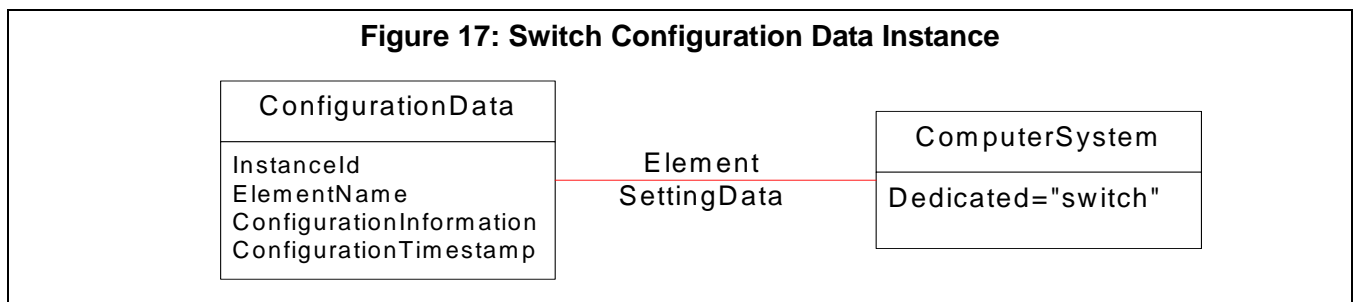
### Clause 12: Switch Configuration Data Subprofile

#### 12.1 Description

This subprofile describes the ability to retrieve a configuration from a switch and latter apply that configuration back on the switch (similar to an image backup and restoration of a computer system).

The profile only has three classes providing all the functionality. When a client needs to obtain a snapshot of the switch configuration, he enumerates `ConfigurationData` which will return the current configuration with the timestamp set appropriately.

When the client wants to apply a configuration, the client creates an instance of `ConfigurationData` and calls the method `ApplyConfiguration()` on the instance containing the property `ConfigurationInformation` which is to be applied to the switch.



#### 12.2 Durable Names and Correlatable IDs of the Profile

None in this version of the specification.

#### 12.3 Instrumentation Requirements

None in this version of the specification.

#### 12.4 Health and Fault Management

None

#### 12.5 Cascading Considerations

None

#### 12.6 Methods of this Profile

##### 12.6.1 ApplyConfiguration

This method applies the configuration data to the switch. The data in the instance's `ConfigurationInformation` property is used as the configuration to apply. Note that it is not necessary for the element to be associated with the `ConfigurationData` instance at the time that this method is called.

```

uint32 ApplyConfiguration (
    boolean ValidateOnly,

```

```
uint16 TypeOfConfiguration
CIM_ManagedElement REF ManagedElement);
```

## 12.7 Client Considerations and Recipes

### 12.7.1 Get Switch Configuration

```
// DESCRIPTION
//
// This recipe describes how to retrieve Switch configuration data.
//
// PRE-EXISTING CONDITIONS AND ASSUMPTIONS
// 1. A reference to the Switch whose configuration data to retrieve is known
// and defined in the variable $Switch->.

// MAIN
// Step 1. Retrieve the configuration of the Switch.
$ConfigData[] = Associators($Switch->,
    "CIM_ElementSettingData",
    "CIM_ConfigurationData",
    "ManagedElement",
    "SettingData",
    false,
    false,
    {"ConfigurationInformation", "ConfigurationTimestamp"})
if ($ConfigData[] == null || $ConfigData[].length != 1) {
    <ERROR! The required Switch configuration data is not available>
}
$SwitchConfig = $ConfigData[0]
```

### 12.7.2 Set Switch Configuration

```
// DESCRIPTION
//
// Set Switch Configuration
//
// PREEXISTING CONDITIONS AND ASSUMPTIONS
//
// None

Placeholder File
```

## 12.8 Registered Name and Version

Switch Configuration Data version 1.1.0

## 12.9 CIM Elements

**Table 123: CIM Elements for Switch Configuration Data**

| Element Name                    | Requirement | Description                                |
|---------------------------------|-------------|--|
| CIM_ComputerSystem (12.9.1)     | Mandatory   | Represents the Switch                      |
| CIM_ElementSettingData (12.9.2) | Mandatory   | Associates ConfigurationData to the switch |
| CIM_ConfigurationData (12.9.3)  | Mandatory   | Switch Configuration Data                  |

### 12.9.1 CIM\_ComputerSystem

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 124 describes class CIM\_ComputerSystem.

**Table 124: SMI Referenced Properties/Methods for CIM\_ComputerSystem**

| Properties        | Flags | Requirement | Description & Notes |
|-------------------|-------|-------------|---------------------|
| CreationClassName |       | Mandatory   | The class name      |
| Name              |       | Mandatory   | Switch Name (WWN)   |
| NameFormat        |       | Mandatory   | "WWN"               |
| Dedicated         |       | Mandatory   | "Switch"            |

### 12.9.2 CIM\_ElementSettingData

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 125 describes class CIM\_ElementSettingData.

**Table 125: SMI Referenced Properties/Methods for CIM\_ElementSettingData**

| Properties  | Flags | Requirement | Description & Notes |
|-------------|-------|-------------|---------------------|
| SettingData |       | Mandatory   |                     |

**Table 125: SMI Referenced Properties/Methods for CIM\_ElementSettingData**

| Properties     | Flags | Requirement | Description & Notes |
|----------------|-------|-------------|---------------------|
| ManagedElement |       | Mandatory   |                     |

**12.9.3 CIM\_ConfigurationData**

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 126 describes class CIM\_ConfigurationData.

**Table 126: SMI Referenced Properties/Methods for CIM\_ConfigurationData**

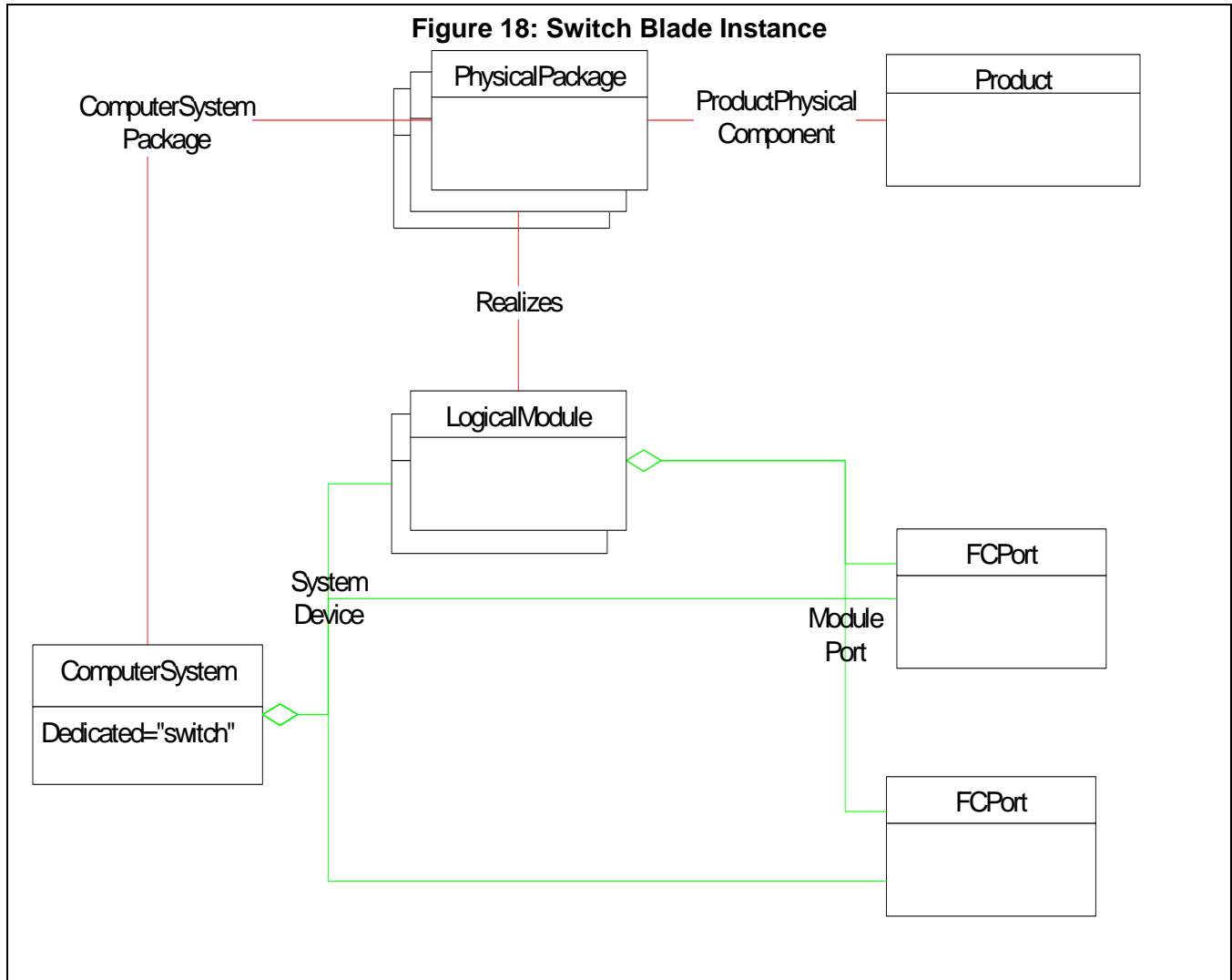
| Properties               | Flags | Requirement | Description & Notes   |
|--------------------------|-------|-------------|---|
| InstanceID               |       | Mandatory   | Opaque  |
| ElementName              |       | Mandatory   | User friendly for configuration file.   |
| ConfigurationInformation |       | Mandatory   | The configuration data of the switch.   |
| ConfigurationTimestamp   |       | Mandatory   | Time the configuration data was obtained  |
| ApplyConfiguration()     |       | Mandatory   | Method that processes the configuration in the same instance and applies it to the switch |

**EXPERIMENTAL**



**STABLE****Clause 13: Blades Subprofile****13.1 Description**

This subprofile describes how blades in a director class switch can be discovered and managed.

**Instance Diagram****13.2 Health and Fault Management**

None

**13.3 Cascading Considerations**

None

### **13.4 Methods of this Profile**

None

### **13.5 Client Considerations and Recipes**

None

### **13.6 Registered Name and Version**

Blades version 1.2.0

## 13.7 CIM Elements

**Table 127: CIM Elements for Blades**

| Element Name   | Requirement | Description  |
|--|-------------|--|
| CIM_LogicalModule (13.7.1)   | Mandatory   | The Blade  |
| CIM_ModulePort (13.7.2)  | Mandatory   | Associates the LogicalModule to the FCPort   |
| CIM_PhysicalPackage (13.7.3)   | Mandatory   | The physical package within which the LogicalModule is contained   |
| CIM_Product (13.7.4)   | Optional    | The product information for the Blade  |
| CIM_ProductPhysicalComponent (13.7.5)  | Optional    | Associates the Product to the PhysicalPackage  |
| CIM_Realizes (13.7.6)  | Conditional | Conditional requirement: Support for the Physical Package profile. Associates the LogicalModule to its PhysicalPackage |
| CIM_SystemDevice (13.7.7)  | Mandatory   | Associates the LogicalModule to the ComputerSystem representing the Switch   |
| SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_FCPort   | Optional    | Creation of an Creation LogicalModule instance.  |
| SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_FCPort   | Optional    | Deletion of an LogicalModule instance.   |
| SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_LogicalModule AND SourceInstance.OperationalStatus <> PreviousInstance.OperationalStatus                                       | Optional    | Deprecated WQL - Change in status of LogicalModule.  |
| SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_LogicalModule AND SourceInstance.CIM_LogicalModule::OperationalStatus <> PreviousInstance.CIM_LogicalModule::OperationalStatus | Optional    | Experimental CQL - Change in status of LogicalModule.  |

### 13.7.1 CIM\_LogicalModule

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 128 describes class CIM\_LogicalModule.

**Table 128: SMI Referenced Properties/Methods for CIM\_LogicalModule**

| Properties              | Flags | Requirement | Description & Notes |
|-------------------------|-------|-------------|---------------------|
| SystemCreationClassName |       | Mandatory   |                     |
| SystemName              |       | Mandatory   |                     |
| CreationClassName       |       | Mandatory   |                     |
| DeviceID                |       | Mandatory   | Opaque              |
| ElementName             |       | Mandatory   |                     |
| OperationalStatus       |       | Mandatory   |                     |
| ModuleNumber            |       | Mandatory   |                     |

### 13.7.2 CIM\_ModulePort

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 129 describes class CIM\_ModulePort.

**Table 129: SMI Referenced Properties/Methods for CIM\_ModulePort**

| Properties     | Flags | Requirement | Description & Notes |
|----------------|-------|-------------|---------------------|
| PartComponent  |       | Mandatory   |                     |
| GroupComponent |       | Mandatory   |                     |

### 13.7.3 CIM\_PhysicalPackage

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 130 describes class CIM\_PhysicalPackage.

**Table 130: SMI Referenced Properties/Methods for CIM\_PhysicalPackage**

| Properties        | Flags | Requirement | Description & Notes |
|-------------------|-------|-------------|---------------------|
| CreationClassName |       | Mandatory   |                     |
| Tag               |       | Mandatory   |                     |
| ElementName       |       | Optional    |                     |
| Name              |       | Optional    |                     |
| Manufacturer      |       | Mandatory   |                     |
| Model             |       | Mandatory   |                     |
| SerialNumber      |       | Optional    |                     |
| Version           |       | Optional    |                     |
| PartNumber        |       | Optional    |                     |

#### 13.7.4 CIM\_Product

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Optional

Table 131 describes class CIM\_Product.

**Table 131: SMI Referenced Properties/Methods for CIM\_Product**

| Properties        | Flags | Requirement | Description & Notes  |
|-------------------|-------|-------------|--|
| Name              |       | Mandatory   | Commonly used Product name.  |
| IdentifyingNumber |       | Mandatory   | Product identification such as a serial number.                        |
| Vendor            |       | Mandatory   | The manufacturer or the OEM.   |
| Version           |       | Mandatory   | Product version information.   |
| ElementName       |       | Mandatory   | User Friendly name. Suggested use is Vendor, Version and product name. |

#### 13.7.5 CIM\_ProductPhysicalComponent

Associates the Product to the PhysicalPackage. This is necessary to link the Product information to the Blade.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Optional

Table 132 describes class CIM\_ProductPhysicalComponent.

**Table 132: SMI Referenced Properties/Methods for CIM\_ProductPhysicalComponent**

| Properties     | Flags | Requirement | Description & Notes |
|----------------|-------|-------------|---------------------|
| GroupComponent |       | Mandatory   |                     |
| PartComponent  |       | Mandatory   |                     |
| PartComponent  |       | Mandatory   |                     |

### 13.7.6 CIM\_Realizes

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: PhysicalPackage

Table 133 describes class CIM\_Realizes.

**Table 133: SMI Referenced Properties/Methods for CIM\_Realizes**

| Properties | Flags | Requirement | Description & Notes  |
|------------|-------|-------------|--|
| Antecedent |       | Mandatory   | The reference to the PhysicalPackage.                      |
| Dependent  |       | Mandatory   | The reference to the LogicalModule representing the Blade. |
| Antecedent |       | Mandatory   |  |
| Dependent  |       | Mandatory   |  |

### 13.7.7 CIM\_SystemDevice

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 134 describes class CIM\_SystemDevice.

**Table 134: SMI Referenced Properties/Methods for CIM\_SystemDevice**

| Properties     | Flags | Requirement | Description & Notes |
|----------------|-------|-------------|---------------------|
| GroupComponent |       | Mandatory   |                     |
| PartComponent  |       | Mandatory   |                     |

**STABLE**

---

---





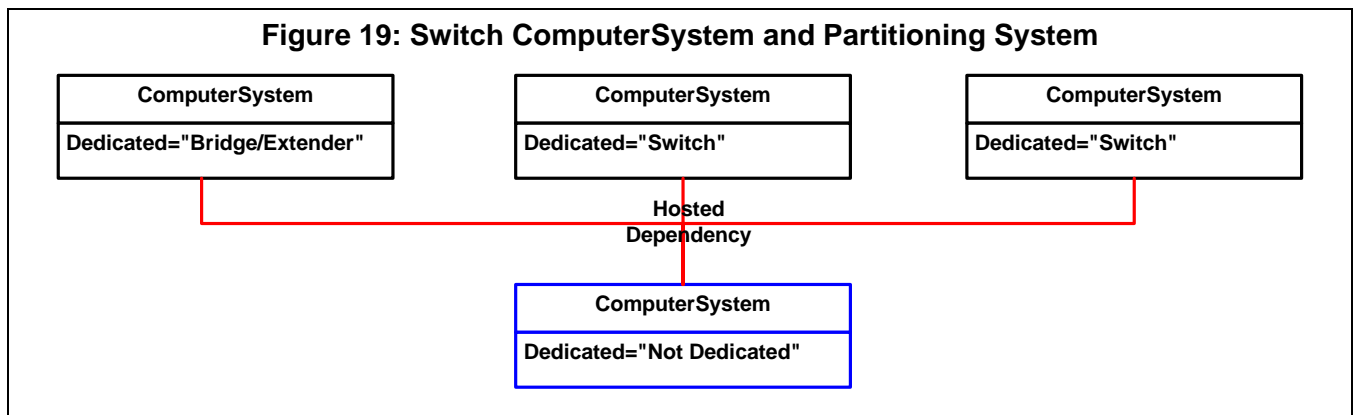
---



---

**EXPERIMENTAL**
**Clause 14: Switch Partitioning Subprofile**
**14.1 Description**

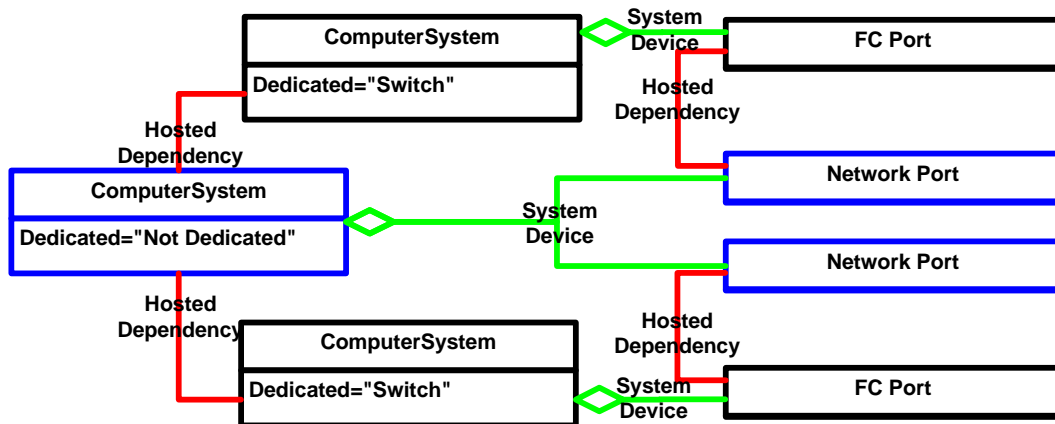
The Switch Partitioning subprofile is used when a “switch” actually is implementing multiple instances of a profile. The instances of the profile can be the same profiles, for example the Switch profile, or different profiles, for example the Switch Profile and the Extender Profile. For the context of further discussion, the “switch” representing the entire set of systems will be called the Partitioning System and the systems that it is “hosting” are the Partitioned System. For virtual fabrics, ANSI T11 calls the partitioning system the “Core Switch” and the partitioned system the “Virtual Switch”. In Figure 19, an example is shown with a Partitioning System hosting a Partitioned System running the Extender Profile and the Switch Profile. The nomenclature in this profile uses blue to identify the partitioning entity.



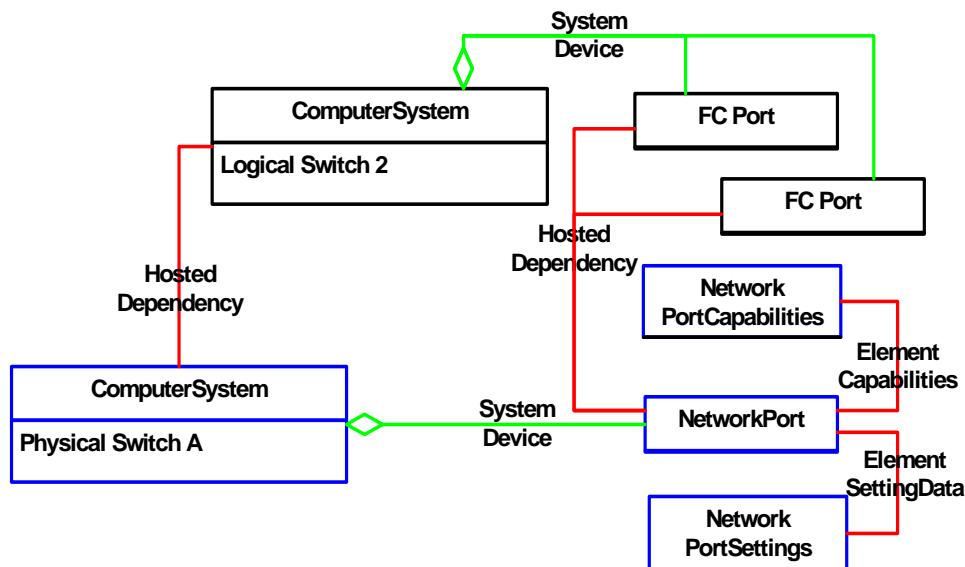
The relationship between the Partitioning System and its partitions is indicated with the HostedDependency association. The Partitioning System can be determined by locating the ComputerSystem that is not a dependent in a HostedDependency relationship to another ComputerSystem. The Partitioning System also has the Dedicated property set to “Not Dedicated”.

In Figure 20, the Partitioning Ports have been added. For this Profile, the NetworkPort class or a subclass is required to identify the partitioning entity.

**Figure 20: Switch and Partitioning System and Partitioning Ports**



**Figure 21: Underlying System Port Settings and Capabilities**



In Figure 21, there is the class NetworkPortCapabilities with the property NetworkIDsConfigurable which indicates whether this port can be assigned to a particular fibre channel fabric. If this property is true, then the property, NetworkIDsFormat should be VF\_ID and the array property NetworkIDs can be set to define which fibre channel virtual fabrics this port can belong to. When the port connects to one of the fibre channel fabrics in the setting, an FCPort instance shall be instantiated to represent the Fibre Channel Port that is active in the particular fabric with the association HostedDependency showing which NetworkPort the FCPort was partitioned from. If the setting contained more than one NetworkID, for every active connection to a fabric, an instance of FCPort shall be instantiated.

The FCSwitches partitioned from the partitioning ComputerSystem are implicitly created when a FCPort is connected to a virtual fabric. Currently there is no mechanism to explicitly create a partitioned switch.

## **14.2 Health and Fault Management Consideration**

None

## **14.3 Cascading Considerations**

None

## **14.4 Supported Profiles, Subprofiles, and Packages**

Not defined in this standard.

## **14.5 Methods of the Profile**

None

## **14.6 Client Considerations and Recipes**

None

## **14.7 Registered Name and Version**

FabricSwitchPartitioning version 1.2.0

## 14.8 CIM Elements

**Table 135: CIM Elements for FabricSwitchPartitioning**

| Element Name   | Requirement | Description  |
|--|-------------|--|
| CIM_HostedDependency (NetworkPort to FCPort) (14.8.1)  | Mandatory   | Association between NetworkPort to FCPort  |
| CIM_HostedDependency (Partitioning CS to Partitioned CS) (14.8.2)                              | Mandatory   | Association between the Partitioning ComputerSystem and Partitioned ComputerSystem |
| CIM_ComputerSystem (Partitioning) (14.8.3)   | Mandatory   | The partitioning ComputerSystem  |
| CIM_SystemDevice (NetworkPort to ComputerSystem) (14.8.4)                                      | Mandatory   | Associates the partitioning classes (NetworkPort to the ComputerSystem)            |
| CIM_NetworkPortCapabilities (14.8.5)   | Mandatory   | The NetworkPort Capabilities.  |
| CIM_NetworkPortSettings (14.8.6)   | Mandatory   | Defines the Virtual Fabrics the switch port can connect to.                        |
| CIM_ElementCapabilities (Association between NetworkPort and NetworkPortCapabilities) (14.8.7) | Mandatory   | Association between NetworkPort and NetworkPortCapabilities.                       |
| CIM_ElementSettingData (Association between NetworkPort and NetworkPortSettings) (14.8.8)      | Mandatory   | Association between NetworkPort and NetworkPortSettings                            |
| CIM_NetworkPort (14.8.9)   | Mandatory   | The partitioning port.   |

### 14.8.1 CIM\_HostedDependency (NetworkPort to FCPort)

The association between the partitioning NetworkPort and the partitioned FCPort. The Antecedent references the partitioning port and the Dependent references the partitioned port. The association can be used to determine whether the system is in the underlying SAN topology or the Fabric topology.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 136 describes class CIM\_HostedDependency (NetworkPort to FCPort).

**Table 136: SMI Referenced Properties/Methods for CIM\_HostedDependency (NetworkPort to FCPort)**

| Properties | Flags | Requirement | Description & Notes                              |
|------------|-------|-------------|--|
| Antecedent |       | Mandatory   | NetworkPort representing the Partitioning System |
| Dependent  |       | Mandatory   | FCPort representing the Partitioned System       |

#### 14.8.2 CIM\_HostedDependency (Partitioning CS to Partitioned CS)

The association between the Partitioning ComputerSystem and the Partitioned ComputerSystem. The Antecedent references the partitioning system and the Dependent references the partitioned system. The association can be used to determine whether the system is in the underlying SAN topology or the Fabric topology.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 137 describes class CIM\_HostedDependency (Partitioning CS to Partitioned CS).

**Table 137: SMI Referenced Properties/Methods for CIM\_HostedDependency (Partitioning CS to Partitioned CS)**

| Properties | Flags | Requirement | Description & Notes                                 |
|------------|-------|-------------|---|
| Antecedent |       | Mandatory   | ComputerSystem representing the Partitioning System |
| Dependent  |       | Mandatory   | ComputerSystem representing the Partitioned System  |

#### 14.8.3 CIM\_ComputerSystem (Partitioning)

The ComputerSystem representing the Interconnect Element (e.g. a switch) or Platform (e.g. Host and Array).

Created By: External

Modified By: Static

Deleted By: External

Class Mandatory:

Table 138 describes class CIM\_ComputerSystem (Partitioning).

**Table 138: SMI Referenced Properties/Methods for CIM\_ComputerSystem (Partitioning)**

| Properties        | Flags | Requirement | Description & Notes          |
|-------------------|-------|-------------|------------------------------|
| CreationClassName |       | Mandatory   | Name of Class                |
| Name              |       | Mandatory   | The Partitioning System Name |
| Dedicated         |       | Mandatory   |                              |

#### 14.8.4 CIM\_SystemDevice (NetworkPort to ComputerSystem)

Associates the NetworkPort to the ComputerSystem

Created By: External

Modified By: Static

Deleted By: External

Class Mandatory:

Table 139 describes class CIM\_SystemDevice (NetworkPort to ComputerSystem).

**Table 139: SMI Referenced Properties/Methods for CIM\_SystemDevice (NetworkPort to ComputerSystem)**

| Properties     | Flags | Requirement | Description & Notes                 |
|----------------|-------|-------------|-------------------------------------|
| GroupComponent |       | Mandatory   | The reference to the ComputerSystem |
| PartComponent  |       | Mandatory   | The reference to the NetworkPort    |

#### 14.8.5 CIM\_NetworkPortCapabilities

The NetworkPort Capabilities.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 140 describes class CIM\_NetworkPortCapabilities.

**Table 140: SMI Referenced Properties/Methods for CIM\_NetworkPortCapabilities**

| Properties             | Flags | Requirement | Description & Notes   |
|------------------------|-------|-------------|---|
| InstanceID             |       | Mandatory   | Opaque  |
| ElementName            |       | Mandatory   | Must be set to "NetworkPortCapabilities"  |
| NetworkIDsConfigurable |       | Mandatory   | If the switch supports configuring virtual fabrics, this property must be TRUE. If the switch only supports discovery, this property must be FALSE. |
| NetworkIDsFormat       |       | Mandatory   | For configuring virtual fabrics, this property must for VF_ID.  |

#### 14.8.6 CIM\_NetworkPortSettings

Defines the Virtual Fabrics the switch port can connect to. The property NetworkIDs is an array which should contain the Virtual Fabric IDs (VF\_ID) that the NetworkPort will partition FCPorts for.

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 141 describes class CIM\_NetworkPortSettings.

**Table 141: SMI Referenced Properties/Methods for CIM\_NetworkPortSettings**

| Properties  | Flags | Requirement | Description & Notes   |
|-------------|-------|-------------|---|
| InstanceID  |       | Mandatory   | Opaque  |
| ElementName |       | Mandatory   | Must be set to "NetworkPortSettings"  |
| NetworkIDs  |       | Mandatory   | Contains the Virtual Fabric IDs that the NetworkPort will host FCPorts for. |

#### 14.8.7 CIM\_ElementCapabilities (Association between NetworkPort and NetworkPortCapabilities)

Association between NetworkPort and NetworkPortCapabilities

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 142 describes class CIM\_ElementCapabilities (Association between NetworkPort and NetworkPortCapabilities).

**Table 142: SMI Referenced Properties/Methods for CIM\_ElementCapabilities (Association between NetworkPort and NetworkPortCapabilities)**

| Properties     | Flags | Requirement | Description & Notes                          |
|----------------|-------|-------------|--|
| ManagedElement |       | Mandatory   | The reference to the ComputerSystem          |
| Capabilities   |       | Mandatory   | The reference to the NetworkPortCapabilities |

#### 14.8.8 CIM\_ElementSettingData (Association between NetworkPort and NetworkPortSettings)

Association between NetworkPort and NetworkPortSettings

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 143 describes class CIM\_ElementSettingData (Association between NetworkPort and NetworkPortSettings).

**Table 143: SMI Referenced Properties/Methods for CIM\_ElementSettingData (Association between NetworkPort and NetworkPortSettings)**

| Properties  | Flags | Requirement | Description & Notes                      |
|-------------|-------|-------------|--|
| SettingData |       | Mandatory   | The reference to the NetworkPortSettings |

**Table 143: SMI Referenced Properties/Methods for CIM\_ElementSettingData (Association between NetworkPort and NetworkPortSettings)**

| Properties     | Flags | Requirement | Description & Notes                 |
|----------------|-------|-------------|-------------------------------------|
| ManagedElement |       | Mandatory   | The reference to the ComputerSystem |

**14.8.9 CIM\_NetworkPort**

The partitioning port.

Created By: External

Modified By: Static

Deleted By: External

Class Mandatory:

Table 144 describes class CIM\_NetworkPort.

**Table 144: SMI Referenced Properties/Methods for CIM\_NetworkPort**

| Properties              | Flags | Requirement | Description & Notes                     |
|-------------------------|-------|-------------|---|
| SystemCreationClassName |       | Mandatory   | The scoping System's CreationClassName. |
| SystemName              |       | Mandatory   | The scoping System's Name.              |
| CreationClassName       |       | Mandatory   | Name of Class                           |
| DeviceID                |       | Mandatory   | Opaque                                  |

**EXPERIMENTAL**



---

---

**EXPERIMENTAL****Clause 15: Extender Profile****15.1 Description**

A FC Extender is a logical entity representing an inter-switch link consisting of two FC Extender Node devices and the Network pipes that connect them.

A FC Extender is used to connect two Fabrics across a LAN, MAN, WAN, or other network communications media.

A FC Extender Node is a physical device that converts Fibre Channel protocol for transmission over different network communication technologies.

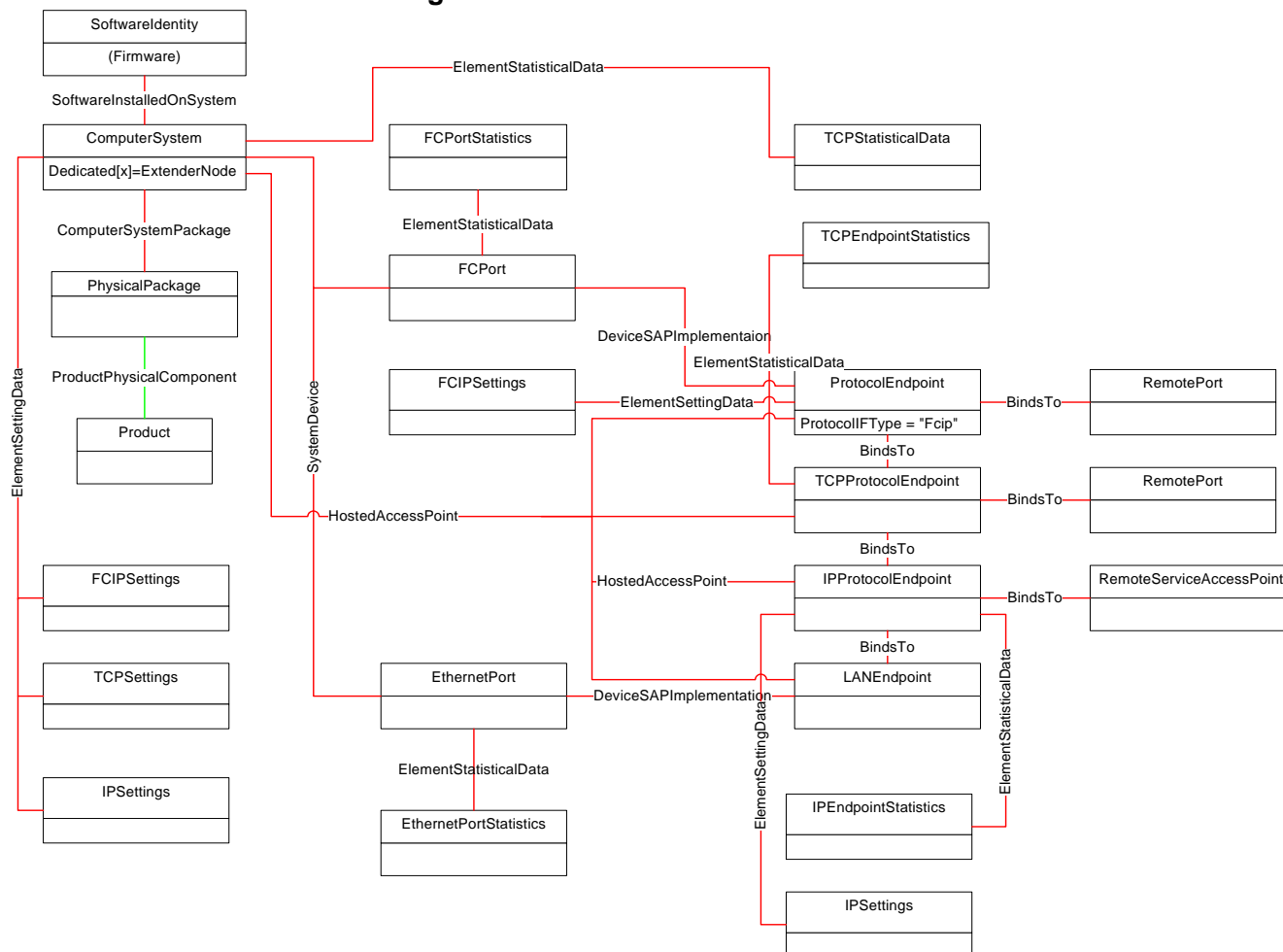
The domain of the Extender Group is defined by Network, which is a subclass of AdminDomain.

**15.1.1 FC Extender Node Topology Classes**

The ComputerSystem class is the core of the model. It is identified as an Extender node by the dedicated attribute being set to ExtenderNode.

The TCPSettings and IPSettings classes represent the global configuration of the FC Extender transport layer.

The Port group of classes contains the following classes: FCPort, and EthernetPort. The FCPort class represents the connection of a FC Extender to a SAN. This class connects to other FCPort classes to represent Fibre channel connections. This class could be replaced with other port types to represent SANs based on other interconnect technology. The EthernetPort class represents an Ethernet link between FC Extender nodes.

**Figure 22: FC Extender Node Instance**

### 15.1.2 FC Extender Node Network Connectivity Classes

Each FC Extender node local ProtocolEndpoint (e.g., FCProtocolEndpoints, TCPProtocolEndpoints) has a BindsTo dependency on a RemotePort that describes access or addressing information to a remote ProtocolEndpoint for a specific connection.

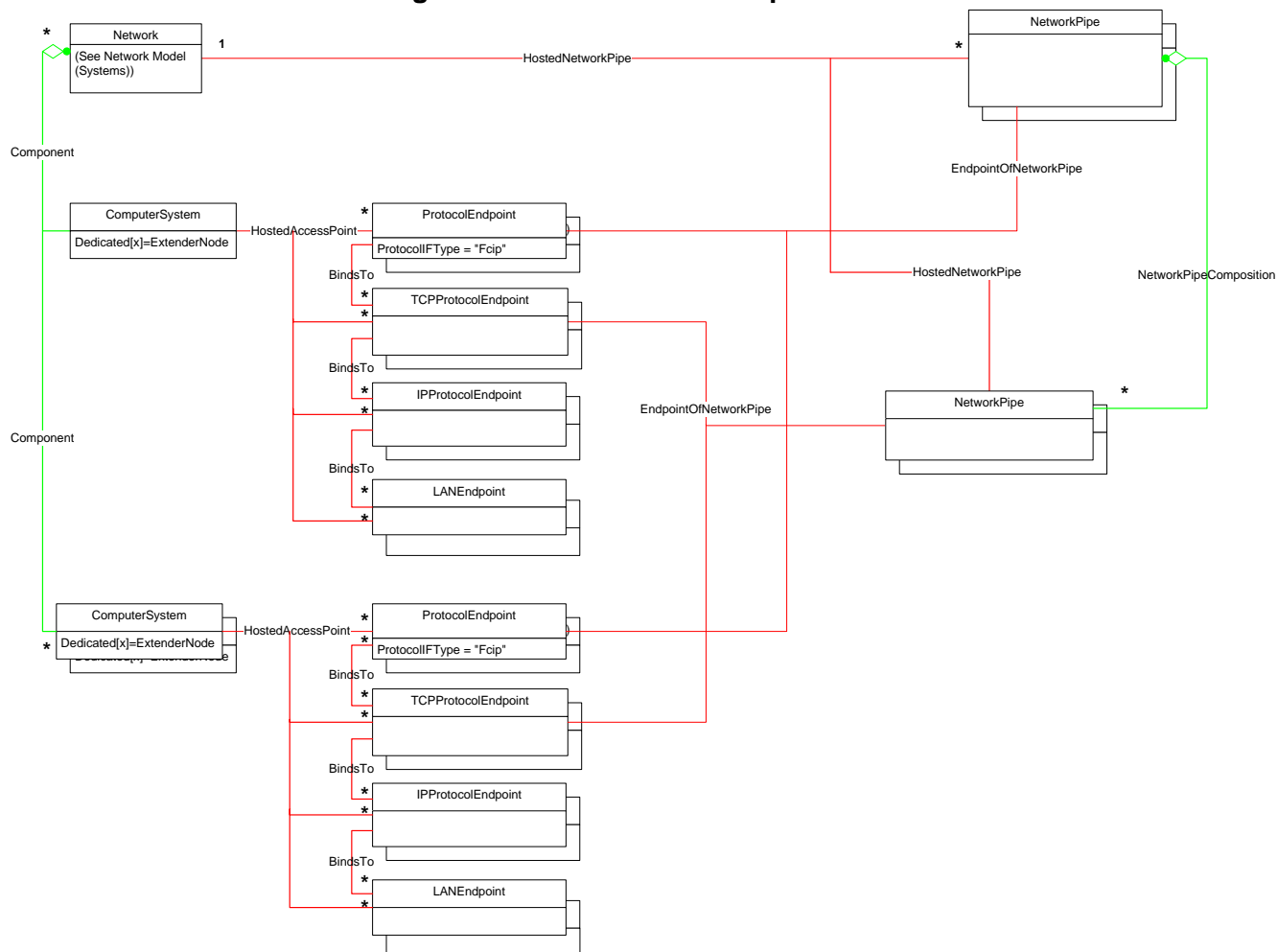
The Extender node represents ProtocolEndpoints dependencies (e.g., FC ProtocolEndpoint on TCPProtocolEndpoints, TCPProtocolEndpoint on IPProtocolEndpoint, IPProtocolEndpoint on EthernetProtocolEndpoint) with a BindsTo association.

### 15.1.3 FC Extender Group Network Connectivity Classes

A FC Extender connection is represented by a NetworkPipe class associated with FCProtocolEndpoints. A FCExtender Network class groups multiple NetworkPipes.

The NetworkPipe between FCProtocolEndpoints is composed of lower-level TCP network pipes.

**Figure 23: FC Extender Group Instance**



## 15.2 Health and Fault Management

None

## 15.3 Cascading Considerations

None

## 15.4 Supported Subprofiles and Packages

None

## 15.5 Methods of this Profile

None

## 15.6 Client Considerations and Recipes

### 15.6.1 Extender Connectivity Settings

// Description

## Extender Profile

```
// Collecting settings
// of Extender node connectivity elements participating in the Extender
// of interest.
// $extenderNodeFCIPSettings(fcip protocol endpoint settings)
// $extenderNodeTCPSettings (transport layer settings)
// $extenderIPSettings (ip protocol endpoint settings)
// PREEXISTING CONDITIONS AND ASSUMPTIONS
// The Extender fcip ProtocolEndpoint has been previously
// identified and defined in the $fcipProtocolEndpoint-> variable
// 1. Get ComputerSystem associated with $fcipProtocolEndpoint

$extenderNodes[] = Associators(
    $fcipProtocolEndpoint->,
    CIM_HostedAccessPoint,
    CIM_ComputerSystem,
    Antecedent,
    Dependent,
    false,
    false,
    [Dedicated])
if (contains(23, $extenderNodes[0].Dedicated))
{
    #extenderNodeAccessA = true
    $extenderNode-> = $extenderNodes[0].getObjectPath()
}
if(#extenderNodeAccessA)
{
    //2. Get fcip protocol endpoint

    // find FCIP Settings
    $fcipSettings[]= Associators(
        $fcipProtocolEndpoint,
        CIM_ElementSettingData,
        CIM_FCIPSettings,
        ManagedElement,
        SettingData,
        false,
        false,
        null)
    if ($fcipSettings[].length != 0)
        $extenderNodeFCIPSettings = $fcipSettings[0]

    //3. Get transport layer settings

    $tcpSettings[]= Associators(
        $extenderNode->,

```

```

        CIM_ElementSettingData,
        CIM_TCPSettings,
        ManagedElement,
        SettingData,
        false,
        false,
        null)
    if ($tcpSettings[0].length != 0)
        $extenderNodeTCPSettings = $tcpSettings[0]

//4. Find TCPProtocolEndpoint bound to the extender fcip ProtocolEndpoint

    $tcpProtocolEndpoint->[] = AssociatorNames(
        $fcipProtocolEndpoint->,
        CIM_BindTo,
        CIM_TCPProtocolEndpoint,
        Dependent,
        Antecedent,
        false,
        false,
        null))

//5. Find IPProtocolEndpoint bound to the extender tcp ProtocolEndpoint

    $ipProtocolEndpoints->[] = AssociatorNames(
        $tcpProtocolEndpoint->,
        CIM_BindTo,
        CIM_IPProtocolEndpoint,
        Dependent,
        Antecedent,
        false,
        false,
        null))
    $ipProtocolEndpoint-> = $ipProtocolEndpoints->[0]

//6. Find IPProtocolEndpoint settings

    $ipSettings[] = Associators(
        $ipProtocolEndpoint->,
        CIM_ElementSettingData,
        CIM_IPSettings,
        ManagedElement,
        SettingData,

```

```

        false,
        false,
        null)
    if ($ipSettings[].length != 0)
        $extenderNodeIPSettings = $ipSettings[0]

}

```

### 15.6.2 Extender Connective Statistics

```

// Description
// Collecting statistical data
// of Extender node connectivity elements participating in the Extender
// of interest.
// $extenderNodeTCPStatisticalData (transport layer stats)
// $extenderIPEndpointStatistics (IP protocol endpoint stats)
// PREEXISTING CONDITIONS AND ASSUMPTIONS
// The Extender fcip ProtocolEndpoint has been previously
// identified and defined in the $fcipProtocolEndpoint-> variable
// 1. Get ComputerSystem associated with $fcipProtocolEndpoint

$extenderNodes[] = Associators(
    $fcipProtocolEndpoint->,
    CIM_HostedAccessPoint,
    CIM_ComputerSystem,
    Antecedent,
    Dependent,
    false,
    false,
    [Dedicated])
if (contains(23, $extenderNodes[0].Dedicated))
{
    #extenderNodeAccess = true
    $extenderNode-> = $extenderNodes[0].getObjectPath()
}
if(#extenderNodeAccess)
{
    //2. Get transport layer statistics

    $tcpStatistics[] = Associators(
        $extenderNode->,
        CIM_ElementStatisticalData,
        CIM_TCPStatisticalData,

```

## Extender Profile

```
        ManagedElement,
        Stats,
        false,
        false,
        null))
$extenderNodeTCPStatisticalData = $tcpStatistics[0]

//3. Find TCPProtocolEndpoint bound to the extender fcip ProtocolEndpoint

$tcpProtocolEndpoint->[] = AssociatorNames(
    $fcipProtocolEndpoint->,
    CIM_BindTo,
    CIM_TCPProtocolEndpoint,
    Dependent,
    Antecedent,
    false,
    false,
    null))

//4. Find IPProtocolEndpoint bound to the extender tcp ProtocolEndpoint

$ipProtocolEndpoint->[] = AssociatorNames(
    $tcpProtocolEndpoint->,
    CIM_BindTo,
    CIM_IPProtocolEndpoint,
    Dependent,
    Antecedent,
    false,
    false,
    null))

//5. Find IPProtocolEndpoint statistics

$ipStatistics[] = Associators(
    $ipProtocolEndpoint->,
    CIM_ElementStatisticalData,
    CIM_IPEndpointStatistics,
    ManagedElement,
    Stats,
    false,
    false,
    null)

$extenderIPEndpointStatistics = $ipStatistics[0]
```

```
}
```

### 15.6.3 Extender Port Group Information

```
// Description
// Collecting configuration and statistical data
// of Extender node ports participating in the Extender
// of interest.
// $extenderNodeFCPort (connected to a switch)
// $extenderNodeFCPortStatistics
// $extenderNodeEthernetPort (connected to a peer Extender node)
// $extenderNodeEthernetStatistics
// PREEXISTING CONDITIONS AND ASSUMPTIONS
// The Extender fcip ProtocolEndpoint has been previously
// identified and defined in the $fcipProtocolEndpoint-> variable

// 1. Get ComputerSystem associated with $fcipProtocolEndpoint

$extenderNodes[] = Associators(
    $fcipProtocolEndpoint->,
    CIM_HostedAccessPoint,
    CIM_ComputerSystem,
    Antecedent,
    Dependent,
    false,
    false,
    [Dedicated])
if (contains(23, $extenderNodes[0].Dedicated))
{
    #extenderNodeAccess = true
}
if(#extenderNodeAccess)
{
    // 2. Get FC port

    $fcPorts[] = Associators(
        $fcipProtocolEndpoints->,
        CIM_DeviceSAPImplementation,
        CIM_FCPort,
        Dependent,
        Antecedent,
        false,
        false,
        null)
    $extenderNodeFCPort = $fcPorts[0]
```



```
// 2. Get FC port statistics

$fcPortStatistics->[] = Associators(
    $extenderNodeFCPort.getObjectPath(),
    CIM_ElementStatisticalData,
    CIM_FCPortStatistics,
    ManagedElement,
    Stats,
    false,
    false,
    null))
$extenderNodeFCPortStatistics = $fcPortsStatistics[0]

//3. Find TCPProtocolEndpoint bound to the extender FCIP ProtocolEndpoint

$tcpProtocolEndpoints->[] = AssociatorNames(
    $fcipProtocolEndpoint->,
    CIM_BindTo,
    CIM_TCPProtocolEndpoint,
    Dependent,
    Antecedent,
    false,
    false,
    null))
// at least one should exist
$tcpProtocolEndpoint->=$tcpProtocolEndpoints->[0]

//4. Find IPProtocolEndpoint bound to the extender TCP ProtocolEndpoint

$ipProtocolEndpoint->[] = AssociatorNames(
    $tcpProtocolEndpoint->,
    CIM_BindTo,
    CIM_IPProtocolEndpoint,
    Dependent,
    Antecedent,
    false,
    false,
    null))
$ipProtocolEndpoint->=$ipProtocolEndpoints->[0]

//5. Get Ethernet port

$ethernetPorts[] = Associators(
    $ipProtocolEndpoints->,
    CIM_DeviceSAPImplementation,
    CIM_EthernetPort,
```

## Extender Profile

```
        Dependent,
        Antecedent,
        false,
        false,
        null)
    $extenderNodeEthernetPort = $ethernetPorts[0]

//6. Get Ethernet port statistics

    $ethernetPortStatistics->[] = Associators(
        $extenderNodeEthernetPort.getObjectPath(),
        CIM_ElementStatisticalData,
        CIM_EthernetStatistics,
        ManagedElement,
        Stats,
        false,
        false,
        null))
    $extenderNodeEthernetPortStatistics = $ethernetPortsStatistics[0]

}
```

### 15.6.4 Extender Topology Mapping

```
// This recipe describes how to build a topology graph of a fabric.
//
// 1. Identifies all the Switches and adds their objects paths and the
// object paths of the FC Ports belonging to these Switches to the $nodes
// array
//
// 2. Creates a suitable Association instance (e.g. a SystemDevice
// Association instance between a Switch and a FC Port), setting its
// GroupComponent and PartComponent. Adds the object path of the
// Association to the $links array
//
// 3. Creates a map of all connected FC Ports (i.e., belonging to Switches
// that are ISL'd together and to Host HBAs and Storage System Front End
// Controllers)
//
// In this map, the FC Ports (i.e., the ones that are connected) are
// cross-connected.
//
// e.g., For a pair of FC Ports, one belonging to a Switch and the other
```

```

// belonging to a Host (HBA), the map indexed by the Switch Port WWN returns
// the Host (HBA) FC Port object path and the map indexed by the Host (HBA)
// FC Port WWN returns the Switch FC Port object path.
//
// Similar relationship exists between the pairs of FC Ports where one
// belongs to a Switch and the other belonging belongs to a Storage System
// Front End Controller and for FC Ports each of which belongs to a Switch.
//
// 4. Identifies all the Hosts and adds their objects paths to the $nodes
// array. Note that the object paths of the FC Ports (HBA Ports) belonging
// to these Hosts are already added to the $nodes array in step-3.
//
// 5. Creates a suitable Association instance (e.g. a SystemDevice
// Association instance between a Host and a FC Port), setting its
// GroupComponent and PartComponent. Adds the object path of the Association
// to the $links array.
//
// 6. Identifies all the Storage Systems and adds their objects paths to the
// $nodes array.
// Note that the object paths of the FC Ports (i.e., Front End Controller
// FC Ports) belonging to these Storage Systems are already added to the
// $nodes array in step-3.
//
// 7. Creates a suitable Association instance (e.g. a SystemDevice
// Association instance between a Storage System and a FC Port), setting
// its GroupComponent and PartComponent. Adds the object path of the
// Association to the $links array.

// DESCRIPTION
// Create a map of how elements in a SAN are connected together via
// Fibre-ChannelFC ports.
//
// The map is built in array $attachedFcPorts->[], where the index is a
// WWN of any device port on the SAN, and the value at that index is
// the object path of the connected Switch or HBA or Storage System FC port.
//
// First find all the switches in a SAN. Get all the FC Ports for each
// switch and get the Attached FC Ports for each Switch FC Port. Save these
// device FC ports in the map described above.

// PREEXISTING CONDITIONS AND ASSUMPTIONS
// 1. All agents/namespaces supporting Fabric Profile previously identified
// using SLP. Do this for each CIMOM supporting Fabric Profile

switches[] = enumerateInstances("CIM_ComputerSystem", true, false, true, true,
                                null)

for #i in $switches[]
{

```

## Extender Profile

```
if (!contains(5, $switches[#i].Dedicated))
    continue

// only process switches, not other computer systems

// Add the switch to the $nodes array

$nodes.addIfNotAlreadyAdded ($switches[#i].getObjectPath());

// Get all the SystemDevice associations between this switch and its
// FC Ports

$sysDevAssoc[] = ReferenceNames($switches[#i],
                                "CIM_FCPort",
                                "GroupComponent");

// Add these associations to the $links array

for #a in $sysDevAssoc->[]
$links.addIfNotAlreadyAdded ($sysDevAssoc->[#a];

$fcPorts->[] = AssociatorNames(
    $switches[#i].getObjectPath(),
    "CIM_SystemDevice",
    "CIM_FCPort",
    "GroupComponent",
    "PartComponent")
for #j in $fcPorts->[]
{

    // Add the FC Port in $nodes array

    $nodes.addIfNotAlreadyAdded (fcPorts->[#j];

    $protocolEndpoints->[] = AssociatorNames(
        fcPorts->[#j],
        "CIM_DeviceSAPImplementation",
        "CIM_ProtocolEndpoint",
        "Antecedent",
        "Dependent");

    // NOTE - It is possible for this collection to be empty (i.e., ports
    // that are not connected). It is possible for this collection to
    // have more than one element (loops attached to a switch port is the
    // most common example).

    if ($protocolEndpoints->[].length == 0)
```

```

        continue

// Add the Protocol End Point to the nodes array.
// Currently this recipe is designed to only save one
// ProtocolEndpoint.

$nodes.addIfNotAlreadyAdded (protocolEndpoints[0]);

// Add the associations between the fcPort and the Protocol end point
// to the links array

$devSAPImplassoc[] = ReferenceNames($fcPorts->[#j],
                                   "CIM_ProtocolEndpoint",
                                   null);
for #a in $devSAPImplassoc->[]
    $links.addIfNotAlreadyAdded ($devSAPImplassoc->[#a];

$attachedProtocolEndpoints->[] = AssociatorNames(
    $protocolEndpoints->[0],
    "CIM_ActiveConnection",
    "CIM_ProtocolEndpoint",
    null, null)

// Add the Attached Protocol End Point to the nodes array

$nodes.addIfNotAlreadyAdded (attachedProtocolEndpoints->[0]);

// Add the associations between the Protocol end point and the
// Attached protocol endpoint to the links array

$actConnassoc[] = ReferenceNames($protocolEndpoint->[#0],
                                "CIM_ActiveConnection",
                                null);
for #a in $actConnassoc->[]
    $links.addIfNotAlreadyAdded ($actConnassoc->[#a];

// NOTE: role & resultRole are null as the direction of the
// association is not dictated by the specification

// $attachedFcPort is either a device FC port or an ISL'd switch FC
// port from another switch. We store this result is stored (i.e.
// which device FC Port is connected // to which switch FC Port) in
// a suitable data structure for subsequent correlation to ports
// discovered on devices.

for #k in $attachedProtocolEndpoints->[]
{

```

```

$attachedFcPorts->[] = Associators(
    $attachedProtocolEndpoints->[#k],
    "CIM_DeviceSAPIImplementation",
    "CIM_FCPort",
    "Dependent",
    "Antecedent",
    false,
    false,
    ["PermanentAddress"])
$attachedFcPort = $attachedFcPorts[0] // Exactly one member guaranteed
                                     by model

    // Add the attached FC Port to the $nodes array
    if $attachedFcPort != null
        $nodes.addIfNotAlreadyAdded ($attachedFcPort);
}
}
}

```

## 15.7 Registered Name and Version

Extender version 1.2.0

## 15.8 CIM Elements

**Table 145: CIM Elements for Extender**

| Element Name  | Requirement | Description  |
|---|-------------|--|
| CIM_BindsTo (IPPE to REmoteSAP) (15.8.1)                                      | Mandatory   | Associates Extender Node ProtocolEndpoints from different layers in the protocol stack |
| CIM_BindsTo (TCPPE to IPPE) (15.8.2)  | Mandatory   | Associates Extender Node ProtocolEndpoints from different layers in the protocol stack |
| CIM_BindsTo (IPPE to PE) (15.8.3)   | Mandatory   | Associates Extender Node ProtocolEndpoints from different layers in the protocol stack |
| CIM_BindsTo (PE to RemotePort) (15.8.4)                                       | Mandatory   | Associates Extender Node ProtocolEndpoints from different layers in the protocol stack |
| CIM_BindsTo (TCPPE to RemotePort) (15.8.5)                                    | Mandatory   | Associates Extender Node ProtocolEndpoints from different layers in the protocol stack |
| CIM_BindsTo (IPPE to RemoteSAP) (15.8.6)                                      | Mandatory   | Associates Extender Node ProtocolEndpoints from different layers in the protocol stack |
| CIM_BindsTo (TCPPE to PE) (15.8.7)  | Mandatory   | Associates Extender Node ProtocolEndpoints from different layers in the protocol stack |
| CIM_Component (15.8.8)  | Mandatory   | Aggregates Extender Nodes in the Network that represents the group of Extenders        |
| CIM_ComputerSystem (15.8.9)   | Mandatory   | Represents the Extender Node   |
| CIM_EndpointOfNetworkPipe (PE to NetworkPipe) (15.8.10)                       | Mandatory   |  |
| CIM_EndpointOfNetworkPipe (TCPPE to NetworkPipe) (15.8.11)                    | Mandatory   |  |
| CIM_ElementSettingData (System to FCIPSettings) (15.8.12)                     | Mandatory   | Associates SettingData to Extender Node or ProtocolEndpoints                           |
| CIM_ElementSettingData (System to TCPSettings) (15.8.13)                      | Mandatory   | Associates SettingData to Extender Node or ProtocolEndpoints                           |
| CIM_ElementSettingData (System to IPSettings) (15.8.14)                       | Mandatory   | Associates SettingData to Extender Node or ProtocolEndpoints                           |
| CIM_ElementSettingData (IPPE to IPSettings) (15.8.15)                         | Mandatory   | Associates SettingData to Extender Node or ProtocolEndpoints                           |
| CIM_ElementSettingData (PE to FCIPSettings) (15.8.16)                         | Mandatory   | Associates SettingData to Extender Node or ProtocolEndpoints                           |
| CIM_ElementSettingData (TCPPE to TCPSettings) (15.8.17)                       | Mandatory   | Associates SettingData to Extender Node or ProtocolEndpoints                           |
| CIM_ElementStatisticalData (EthernetPort to EthernetPortStatistics) (15.8.18) | Mandatory   | Associates StatisticalData to Extender Node or ProtocolEndpoints                       |
| CIM_ElementStatisticalData (FCPort to FCPortStatistics) (15.8.19)             | Mandatory   |  |

**Table 145: CIM Elements for Extender**

| Element Name  | Requirement | Description   |
|---|-------------|---|
| CIM_ElementStatisticalData (IPPE to IPEndpointStatistics) (15.8.20)     | Mandatory   |   |
| CIM_ElementStatisticalData (TCPPE to TCPEndpointStatistics) (15.8.21)   | Mandatory   |   |
| CIM_ElementStatisticalData (System to TCPStatisticalData) (15.8.22)     | Mandatory   |   |
| CIM_EthernetPort (15.8.23)  | Mandatory   |   |
| CIM_EthernetPortStatistics (15.8.24)                                    | Mandatory   |   |
| CIM_FCPort (15.8.25)  | Mandatory   |   |
| CIM_FCPortStatistics (15.8.26)  | Mandatory   |   |
| CIM_FCIPSettings (15.8.27)  | Mandatory   | Defines FCIP settings for a group of ProtocolEndpoints (ProtocolType - "Fcip") which belongs to the ComputerSystem (Extender Node)  |
| CIM_HostedAccessPoint (ComputerSystem to ProtocolEndpoint) (15.8.28)    | Mandatory   | Associates the ProtocolEndpoint to the ComputerSystem or Network  |
| CIM_HostedAccessPoint (ComputerSystem to TCPProtocolEndpoint) (15.8.29) | Mandatory   | Associates the ProtocolEndpoint to the ComputerSystem or Network  |
| CIM_HostedAccessPoint (ComputerSystem to IPPProtocolEndpoint) (15.8.30) | Mandatory   | Associates the ProtocolEndpoint to the ComputerSystem or Network  |
| CIM_HostedNetworkPipe (15.8.31)   | Mandatory   | Associates NetworkPipe to the Network   |
| CIM_IPEndpointStatistics (15.8.32)                                      | Mandatory   |   |
| CIM_IPProtocolEndpoint (15.8.33)  | Mandatory   |   |
| CIM_IPSettings (15.8.34)  | Mandatory   | Defines IP settings for a group of IPPProtocolEndpoints which belongs to the ComputerSystem   |
| CIM_Network (15.8.35)   | Mandatory   | Network represents a network connectivity domain. It groups NetworkPipes.   |
| CIM_NetworkPipe (15.8.36)   | Mandatory   | NetworkPipe represents state, configuration of a connection between endpoints in the context of a Network   |
| CIM_NetworkPipeComposition (15.8.37)                                    | Mandatory   |   |
| CIM_DeviceSAPImplementation (15.8.38)                                   | Mandatory   |   |
| CIM_ProtocolEndpoint (15.8.39)  | Mandatory   | The ProtocolEndpoint representing the FCIP layer. ProtocolEndpoint shall be implemented when an ActiveConnection or NetworkPipe exists. It may be implemented if no ActiveConnection or NetworkPipe exists. |



**Table 145: CIM Elements for Extender**

| Element Name  | Requirement | Description  |
|---|-------------|--|
| CIM_LANEndpoint (15.8.40)   | Mandatory   | The ProtocolEndpoint for the Ethernet port.  |
| CIM_RemotePort (15.8.41)  | Optional    |  |
| CIM_RemoteServiceAccessPoint (15.8.42)  | Optional    |  |
| CIM_SystemDevice (System to EthernetPort) (15.8.43)   | Mandatory   | Associated FCPort and EthernetPort to the ComputerSystem                                     |
| CIM_SystemDevice (System to FCPort) (15.8.44)   | Mandatory   | Associated FCPort and EthernetPort to the ComputerSystem                                     |
| CIM_TCPEndpointStatistics (15.8.45)   | Mandatory   | Opaque   |
| CIM_TCPProtocolEndpoint (15.8.46)   | Mandatory   |  |
| CIM_TCPSettings (15.8.47)   | Mandatory   | Defines TCP settings for a group of TCPProtocolEndpoints which belongs to the ComputerSystem |
| CIM_TCPStatisticalData (15.8.48)  | Mandatory   |  |
| SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_ComputerSystem  | Mandatory   | Creation of a ComputerSystem instance  |
| SELECT * FROM CIM_InstDeletion WHERE SourceInstance CIM_ComputerSystem  | Mandatory   | Deletion of a computer system instance   |
| SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ComputerSystem AND SourceInstance.Operationalstatus ** PreviousInstance.Operationalstatus   | Mandatory   | Deprecated WQL - Change of OperationalStatus for a Computer System                           |
| SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ComputerSystem AND SourceInstance.CIM_ComputerSystem::Operationalstatus ** PreviousInstance.CIM_ComputerSystem::Operationalstatus | Optional    | Experimental CQL - Change of OperationalStatus for a Computer System                         |
| SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ComputerSystem AND SourceInstance.Operationalstatus ** PreviousInstance.Operationalstatus   | Mandatory   | Deprecated WQL - Change of OperationalStatus for a Computer System                           |
| SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ComputerSystem AND SourceInstance.CIM_ComputerSystem::Operationalstatus ** PreviousInstance.CIM_ComputerSystem::Operationalstatus | Optional    | Experimental CQL - Change of OperationalStatus for a Computer System                         |

**15.8.1 CIM\_BindsTo (IPPE to REmoteSAP)**

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 146 describes class CIM\_BindsTo (IPPE to REmoteSAP).

**Table 146: SMI Referenced Properties/Methods for CIM\_BindsTo (IPPE to REmoteSAP)**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Dependent  |       | Mandatory   |                     |
| Antecedent |       | Mandatory   |                     |

**15.8.2 CIM\_BindsTo (TCPPE to IPPE)**

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 147 describes class CIM\_BindsTo (TCPPE to IPPE).

**Table 147: SMI Referenced Properties/Methods for CIM\_BindsTo (TCPPE to IPPE)**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Antecedent |       | Mandatory   |                     |
| Dependent  |       | Mandatory   |                     |

**15.8.3 CIM\_BindsTo (IPPE to PE)**

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 148 describes class CIM\_BindsTo (IPPE to PE).

**Table 148: SMI Referenced Properties/Methods for CIM\_BindsTo (IPPE to PE)**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Antecedent |       | Mandatory   |                     |
| Dependent  |       | Mandatory   |                     |

#### 15.8.4 CIM\_BindsTo (PE to RemotePort)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 149 describes class CIM\_BindsTo (PE to RemotePort).

**Table 149: SMI Referenced Properties/Methods for CIM\_BindsTo (PE to RemotePort)**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Antecedent |       | Mandatory   |                     |
| Dependent  |       | Mandatory   |                     |

#### 15.8.5 CIM\_BindsTo (TCPPE to RemotePort)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 150 describes class CIM\_BindsTo (TCPPE to RemotePort).

**Table 150: SMI Referenced Properties/Methods for CIM\_BindsTo (TCPPE to RemotePort)**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Antecedent |       | Mandatory   |                     |
| Dependent  |       | Mandatory   |                     |

#### 15.8.6 CIM\_BindsTo (IPPE to RemoteSAP)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 151 describes class CIM\_BindsTo (IPPE to RemoteSAP).

**Table 151: SMI Referenced Properties/Methods for CIM\_BindsTo (IPPE to RemoteSAP)**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Antecedent |       | Mandatory   |                     |
| Dependent  |       | Mandatory   |                     |

### 15.8.7 CIM\_BindsTo (TCPPE to PE)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 152 describes class CIM\_BindsTo (TCPPE to PE).

**Table 152: SMI Referenced Properties/Methods for CIM\_BindsTo (TCPPE to PE)**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Antecedent |       | Mandatory   |                     |
| Dependent  |       | Mandatory   |                     |

### 15.8.8 CIM\_Component

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 153 describes class CIM\_Component.

**Table 153: SMI Referenced Properties/Methods for CIM\_Component**

| Properties     | Flags | Requirement | Description & Notes |
|----------------|-------|-------------|---------------------|
| PartComponent  |       | Mandatory   |                     |
| GroupComponent |       | Mandatory   |                     |

### 15.8.9 CIM\_ComputerSystem

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 154 describes class CIM\_ComputerSystem.

**Table 154: SMI Referenced Properties/Methods for CIM\_ComputerSystem**

| Properties              | Flags | Requirement | Description & Notes         |
|-------------------------|-------|-------------|-----------------------------|
| CreationClassName       |       | Mandatory   |                             |
| Name                    |       | Mandatory   | IP Address                  |
| ElementName             |       | Optional    | User friendly name          |
| NameFormat              |       | Mandatory   | IP Address                  |
| OtherIdentifyingInfo    |       | Optional    | DNS name                    |
| OperationalStatus       |       | Mandatory   | Status of Computer System.  |
| IdentifyingDescriptions |       | Optional    | Fully qualified domain name |
| Dedicated               |       | Mandatory   | ExtenderNode                |

### 15.8.10 CIM\_EndpointOfNetworkPipe (PE to NetworkPipe)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 155 describes class CIM\_EndpointOfNetworkPipe (PE to NetworkPipe).

**Table 155: SMI Referenced Properties/Methods for CIM\_EndpointOfNetworkPipe (PE to NetworkPipe)**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Dependent  |       | Mandatory   |                     |
| Antecedent |       | Mandatory   |                     |

#### 15.8.11 CIM\_EndpointOfNetworkPipe (TCPPE to NetworkPipe)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 156 describes class CIM\_EndpointOfNetworkPipe (TCPPE to NetworkPipe).

**Table 156: SMI Referenced Properties/Methods for CIM\_EndpointOfNetworkPipe (TCPPE to NetworkPipe)**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Dependent  |       | Mandatory   |                     |
| Antecedent |       | Mandatory   |                     |

#### 15.8.12 CIM\_ElementSettingData (System to FCIPSettings)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 157 describes class CIM\_ElementSettingData (System to FCIPSettings).

**Table 157: SMI Referenced Properties/Methods for CIM\_ElementSettingData (System to FCIPSettings)**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| IsDefault  |       | Mandatory   |                     |
| IsCurrent  |       | Mandatory   |                     |

**Table 157: SMI Referenced Properties/Methods for CIM\_ElementSettingData (System to FCIPSettings)**

| Properties     | Flags | Requirement | Description & Notes |
|----------------|-------|-------------|---------------------|
| SettingData    |       | Mandatory   |                     |
| ManagedElement |       | Mandatory   |                     |

#### 15.8.13 CIM\_ElementSettingData (System to TCPSettings)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 158 describes class CIM\_ElementSettingData (System to TCPSettings).

**Table 158: SMI Referenced Properties/Methods for CIM\_ElementSettingData (System to TCPSettings)**

| Properties     | Flags | Requirement | Description & Notes |
|----------------|-------|-------------|---------------------|
| IsDefault      |       | Mandatory   |                     |
| IsCurrent      |       | Mandatory   |                     |
| SettingData    |       | Mandatory   |                     |
| ManagedElement |       | Mandatory   |                     |

#### 15.8.14 CIM\_ElementSettingData (System to IPSettings)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 159 describes class CIM\_ElementSettingData (System to IPSettings).

**Table 159: SMI Referenced Properties/Methods for CIM\_ElementSettingData (System to IPSettings)**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| IsDefault  |       | Mandatory   |                     |

**Table 159: SMI Referenced Properties/Methods for CIM\_ElementSettingData (System to IPSettings)**

| Properties     | Flags | Requirement | Description & Notes |
|----------------|-------|-------------|---------------------|
| IsCurrent      |       | Mandatory   |                     |
| SettingData    |       | Mandatory   |                     |
| ManagedElement |       | Mandatory   |                     |

**15.8.15 CIM\_ElementSettingData (IPPE to IPSettings)**

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 160 describes class CIM\_ElementSettingData (IPPE to IPSettings).

**Table 160: SMI Referenced Properties/Methods for CIM\_ElementSettingData (IPPE to IPSettings)**

| Properties     | Flags | Requirement | Description & Notes |
|----------------|-------|-------------|---------------------|
| IsDefault      |       | Mandatory   |                     |
| IsCurrent      |       | Mandatory   |                     |
| ManagedElement |       | Mandatory   |                     |
| SettingData    |       | Mandatory   |                     |

**15.8.16 CIM\_ElementSettingData (PE to FCIPSettings)**

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 161 describes class CIM\_ElementSettingData (PE to FCIPSettings).

**Table 161: SMI Referenced Properties/Methods for CIM\_ElementSettingData (PE to FCIPSettings)**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| IsDefault  |       | Mandatory   |                     |
| IsCurrent  |       | Mandatory   |                     |



**Table 161: SMI Referenced Properties/Methods for CIM\_ElementSettingData (PE to FCIPSettings)**

| Properties     | Flags | Requirement | Description & Notes |
|----------------|-------|-------------|---------------------|
| SettingData    |       | Mandatory   |                     |
| ManagedElement |       | Mandatory   |                     |

**15.8.17 CIM\_ElementSettingData (TCPPE to TCPSettings)**

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 162 describes class CIM\_ElementSettingData (TCPPE to TCPSettings).

**Table 162: SMI Referenced Properties/Methods for CIM\_ElementSettingData (TCPPE to TCPSettings)**

| Properties     | Flags | Requirement | Description & Notes |
|----------------|-------|-------------|---------------------|
| IsDefault      |       | Mandatory   |                     |
| IsCurrent      |       | Mandatory   |                     |
| SettingData    |       | Mandatory   |                     |
| ManagedElement |       | Mandatory   |                     |

**15.8.18 CIM\_ElementStatisticalData (EthernetPort to EthernetPortStatistics)**

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 163 describes class CIM\_ElementStatisticalData (EthernetPort to EthernetPortStatistics).

**Table 163: SMI Referenced Properties/Methods for CIM\_ElementStatisticalData (EthernetPort to EthernetPortStatistics)**

| Properties     | Flags | Requirement | Description & Notes |
|----------------|-------|-------------|---------------------|
| Stats          |       | Mandatory   |                     |
| ManagedElement |       | Mandatory   |                     |

### 15.8.19 CIM\_ElementStatisticalData (FCPort to FCPortStatistics)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 164 describes class CIM\_ElementStatisticalData (FCPort to FCPortStatistics).

**Table 164: SMI Referenced Properties/Methods for CIM\_ElementStatisticalData (FCPort to FCPortStatistics)**

| Properties     | Flags | Requirement | Description & Notes |
|----------------|-------|-------------|---------------------|
| Stats          |       | Mandatory   |                     |
| ManagedElement |       | Mandatory   |                     |

### 15.8.20 CIM\_ElementStatisticalData (IPPE to IPEndpointStatistics)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 165 describes class CIM\_ElementStatisticalData (IPPE to IPEndpointStatistics).

**Table 165: SMI Referenced Properties/Methods for CIM\_ElementStatisticalData (IPPE to IPEndpointStatistics)**

| Properties     | Flags | Requirement | Description & Notes |
|----------------|-------|-------------|---------------------|
| ManagedElement |       | Mandatory   |                     |
| Stats          |       | Mandatory   |                     |

### 15.8.21 CIM\_ElementStatisticalData (TCPPE to TCPEndpointStatistics)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 166 describes class CIM\_ElementStatisticalData (TCPPE to TCPEndpointStatistics).

**Table 166: SMI Referenced Properties/Methods for CIM\_ElementStatisticalData (TCPPE to TCPEndpointStatistics)**

| Properties     | Flags | Requirement | Description & Notes |
|----------------|-------|-------------|---------------------|
| ManagedElement |       | Mandatory   |                     |
| Stats          |       | Mandatory   |                     |

#### 15.8.22 CIM\_ElementStatisticalData (System to TCPStatisticalData)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 167 describes class CIM\_ElementStatisticalData (System to TCPStatisticalData).

**Table 167: SMI Referenced Properties/Methods for CIM\_ElementStatisticalData (System to TCP-StatisticalData)**

| Properties     | Flags | Requirement | Description & Notes |
|----------------|-------|-------------|---------------------|
| ManagedElement |       | Mandatory   |                     |
| Stats          |       | Mandatory   |                     |

#### 15.8.23 CIM\_EthernetPort

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 168 describes class CIM\_EthernetPort.

**Table 168: SMI Referenced Properties/Methods for CIM\_EthernetPort**

| Properties                   | Flags | Requirement | Description & Notes |
|------------------------------|-------|-------------|---------------------|
| SystemCreationClass<br>sName |       | Mandatory   |                     |

**Table 168: SMI Referenced Properties/Methods for CIM\_EthernetPort**

| Properties        | Flags | Requirement | Description & Notes   |
|-------------------|-------|-------------|---|
| SystemName        |       | Mandatory   |   |
| CreationClassName |       | Mandatory   |   |
| DeviceID          |       | Mandatory   |   |
| ElementName       |       | Optional    | User friendly name  |
| OperationalStatus |       | Mandatory   |   |
| Speed             |       | Mandatory   |   |
| MaxSpeed          |       | Mandatory   |   |
| PortType          |       | Mandatory   | Supported port mode 10BaseT, 10-100BaseT, 100BaseT, 1000BaseT, etc. |
| PortNumber        |       | Mandatory   | System level port or bus identification number                      |
| NetworkAddresses  |       | Mandatory   | MAC addresses   |
| LinkTechnology    |       | Mandatory   | Ethernet  |

**15.8.24 CIM\_EthernetPortStatistics**

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 169 describes class CIM\_EthernetPortStatistics.

**Table 169: SMI Referenced Properties/Methods for CIM\_EthernetPortStatistics**

| Properties         | Flags | Requirement | Description & Notes |
|--------------------|-------|-------------|---------------------|
| InstanceID         |       | Mandatory   | Opaque              |
| ElementName        |       | Mandatory   |                     |
| StatisticTime      |       | Optional    |                     |
| BytesTransmitted   |       | Mandatory   |                     |
| BytesReceived      |       | Mandatory   |                     |
| PacketsTransmitted |       | Mandatory   |                     |
| PacketsReceived    |       | Mandatory   |                     |
| SymbolErrors       |       | Mandatory   |                     |
| AlignmentErrors    |       | Optional    |                     |

**Table 169: SMI Referenced Properties/Methods for CIM\_EthernetPortStatistics**

| Properties                | Flags | Requirement | Description & Notes |
|---------------------------|-------|-------------|---------------------|
| FCSErrors                 |       | Optional    |                     |
| SingleCollisionFrames     |       | Optional    |                     |
| MultipleCollisionFrames   |       | Optional    |                     |
| DeferredTransmissions     |       | Optional    |                     |
| LateCollisions            |       | Optional    |                     |
| ExcessiveCollisions       |       | Optional    |                     |
| InternalMACTransmitErrors |       | Optional    |                     |
| InternalMACReceiveErrors  |       | Optional    |                     |
| CarrierSenseErrors        |       | Mandatory   |                     |
| FrameTooLongs             |       | Optional    |                     |
| ResetSelectedStats()      |       | Optional    |                     |

**15.8.25 CIM\_FCPort**

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 170 describes class CIM\_FCPort.

**Table 170: SMI Referenced Properties/Methods for CIM\_FCPort**

| Properties              | Flags | Requirement | Description & Notes |
|-------------------------|-------|-------------|---------------------|
| SystemCreationClassName |       | Mandatory   |                     |
| SystemName              |       | Mandatory   |                     |
| CreationClassName       |       | Mandatory   |                     |
| DeviceID                |       | Mandatory   |                     |
| ElementName             |       | Optional    | User Friendly Name  |
| OperationalStatus       |       | Mandatory   |                     |

**Table 170: SMI Referenced Properties/Methods for CIM\_FCPort**

| Properties                       | Flags | Requirement | Description & Notes  |
|----------------------------------|-------|-------------|--|
| Speed                            |       | Mandatory   | Speed of zero represents a link not established 1Gb is 1062500000 bps 2Gb is 2125000000 bps 4Gb is 4250000000 bps) 10Gb single channel variants are 10518750000 bps 10Gb four channel variants are 12750000000 bps This is the raw bit rate.   |
| MaxSpeed                         |       | Mandatory   | The max speed of the Port in Bits per Second using the same algorithm as Speed.  |
| PortType                         |       | Mandatory   | FC-GS Port.Type The specific mode currently enabled for the Port. The values: "N" = Node Port, "NL" = Node Port supporting FC arbitrated loop, "E" = Expansion Port connecting fabric elements (for example, FC switches), "F" = Fabric (element) Port, "FL" = Fabric (element) Port supporting FC arbitrated loop, and "B" = Bridge Port. PortTypes are defined in the ANSI INCITS FC-GS standards. When set to 1 ("Other"), the related property OtherPortType contains a string description of the port's type. PortType is defined to force consistent naming of the 'type' property in subclasses and to guarantee unique enum values for all instances of NetworkPort. A range of values, DMTF_Reserved, has been defined that allows subclasses to override and define their specific port types. Vendor Reserved = 16000..65535 can be used if the PortType is not one already defined in the above enumerations and a vendor subclass is defined specifying the appropriate 'value' and 'valuemap'. |
| PortNumber                       |       | Mandatory   | System level port or bus identification number   |
| PermanentAddress                 |       | Mandatory   | For FibreChannel, it is the Fibre Channel Port WWN.  |
| LinkTechnology                   |       | Mandatory   | "FC"   |
| SupportedCOS                     |       | Mandatory   |  |
| ActiveCOS                        |       | Optional    |  |
| SupportedMaximumTransmissionUnit |       | Mandatory   |  |
| ActiveMaximumTransmissionUnit    |       | Optional    |  |

**15.8.26 CIM\_FCPortStatistics**

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 171 describes class CIM\_FCPortStatistics.

**Table 171: SMI Referenced Properties/Methods for CIM\_FCPortStatistics**

| Properties                       | Flags | Requirement | Description & Notes |
|----------------------------------|-------|-------------|---------------------|
| InstanceID                       |       | Mandatory   | Opaque              |
| ElementName                      |       | Optional    |                     |
| StatisticTime                    |       | Mandatory   |                     |
| BytesTransmitted                 |       | Mandatory   |                     |
| BytesReceived                    |       | Mandatory   |                     |
| PacketsTransmitted               |       | Mandatory   |                     |
| PacketsReceived                  |       | Mandatory   |                     |
| CRCErrors                        |       | Mandatory   |                     |
| LinkFailures                     |       | Mandatory   |                     |
| PrimitiveSeqProtocol<br>ErrCount |       | Mandatory   |                     |
| LossOfSignalCounter              |       | Mandatory   |                     |
| InvalidTransmission<br>Words     |       | Mandatory   |                     |
| SampleInterval                   |       | Mandatory   |                     |
| LIPCount                         |       | Mandatory   |                     |
| NOSCount                         |       | Mandatory   |                     |
| ErrorFrames                      |       | Mandatory   |                     |
| DumpedFrames                     |       | Mandatory   |                     |
| LossOfSyncCounter                |       | Mandatory   |                     |
| FramesTooShort                   |       | Mandatory   |                     |
| FramesTooLong                    |       | Mandatory   |                     |
| AddressErrors                    |       | Mandatory   |                     |
| BufferCreditNotProvi<br>ded      |       | Mandatory   |                     |
| DelimiterErrors                  |       | Mandatory   |                     |
| EncodingDisparityErr<br>ors      |       | Mandatory   |                     |
| LinkResetsReceived               |       | Mandatory   |                     |
| LinkResetsTransmitt<br>ed        |       | Mandatory   |                     |

**Table 171: SMI Referenced Properties/Methods for CIM\_FCPortStatistics**

| Properties                 | Flags | Requirement | Description & Notes |
|----------------------------|-------|-------------|---------------------|
| MulticastFramesReceived    |       | Mandatory   |                     |
| MulticastFramesTransmitted |       | Mandatory   |                     |
| RXBroadcastFrames          |       | Mandatory   |                     |
| TXBroadcastFrames          |       | Mandatory   |                     |
| FBSYFrames                 |       | Mandatory   |                     |
| PBSYFrames                 |       | Mandatory   |                     |
| FRJTFrames                 |       | Mandatory   |                     |
| PRJTFrames                 |       | Mandatory   |                     |
| RXClass1Frames             |       | Mandatory   |                     |
| TXClass1Frames             |       | Mandatory   |                     |
| RXClass2Frames             |       | Mandatory   |                     |
| TXClass2Frames             |       | Mandatory   |                     |
| Class2FBSY                 |       | Mandatory   |                     |
| Class2PBSY                 |       | Mandatory   |                     |
| Class2FRJT                 |       | Mandatory   |                     |
| Class2PRJT                 |       | Mandatory   |                     |
| RXClass3Frames             |       | Mandatory   |                     |
| TXClass3Frames             |       | Mandatory   |                     |
| Class3FramesDiscarded      |       | Mandatory   |                     |
| ResetSelectedStats()       |       | Optional    |                     |

**15.8.27 CIM\_FCIPSettings**

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory



Table 172 describes class CIM\_FCIPSettings.

**Table 172: SMI Referenced Properties/Methods for CIM\_FCIPSettings**

| Properties           | Flags | Requirement | Description & Notes  |
|----------------------|-------|-------------|--|
| InstanceID           |       | Mandatory   | Opaque   |
| ElementName          |       | Optional    | User friendly name. In addition, it can be used as a index property for a search or query. |
| ConnectionUsageFlags |       | Mandatory   |  |
| SpecialFrameTimeout  |       | Mandatory   |  |
| KeepAliveTimeout     |       | Mandatory   |  |

#### 15.8.28 CIM\_HostedAccessPoint (ComputerSystem to ProtocolEndpoint)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 173 describes class CIM\_HostedAccessPoint (ComputerSystem to ProtocolEndpoint).

**Table 173: SMI Referenced Properties/Methods for CIM\_HostedAccessPoint (ComputerSystem to ProtocolEndpoint)**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Dependent  |       | Mandatory   |                     |
| Antecedent |       | Mandatory   |                     |

#### 15.8.29 CIM\_HostedAccessPoint (ComputerSystem to TCPProtocolEndpoint)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 174 describes class CIM\_HostedAccessPoint (ComputerSystem to TCPProtocolEndpoint).

**Table 174: SMI Referenced Properties/Methods for CIM\_HostedAccessPoint (ComputerSystem to TCPProtocolEndpoint)**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Dependent  |       | Mandatory   |                     |
| Antecedent |       | Mandatory   |                     |

### 15.8.30 CIM\_HostedAccessPoint (ComputerSystem to IPProtocolEndpoint)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 175 describes class CIM\_HostedAccessPoint (ComputerSystem to IPProtocolEndpoint).

**Table 175: SMI Referenced Properties/Methods for CIM\_HostedAccessPoint (ComputerSystem to IPProtocolEndpoint)**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Dependent  |       | Mandatory   |                     |
| Antecedent |       | Mandatory   |                     |

### 15.8.31 CIM\_HostedNetworkPipe

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 176 describes class CIM\_HostedNetworkPipe.

**Table 176: SMI Referenced Properties/Methods for CIM\_HostedNetworkPipe**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Dependent  |       | Mandatory   |                     |
| Antecedent |       | Mandatory   |                     |

**15.8.32 CIM\_IPEndpointStatistics**

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 177 describes class CIM\_IPEndpointStatistics.

**Table 177: SMI Referenced Properties/Methods for CIM\_IPEndpointStatistics**

| Properties                       | Flags | Requirement | Description & Notes  |
|----------------------------------|-------|-------------|--|
| InstanceID                       |       | Mandatory   | Opaque   |
| ElementName                      |       | Optional    | User friendly name. In addition, it can be used as a index property for a search or query. |
| StatisticTime                    |       | Mandatory   |  |
| ReceivedPDUs                     |       | Mandatory   |  |
| ReceivedPDUHeaderErrors          |       | Mandatory   |  |
| ReceivedPDUAddressErrors         |       | Mandatory   |  |
| ReceivedPDUForwards              |       | Mandatory   |  |
| ReceivedPDUUnknownProtocolErrors |       | Mandatory   |  |
| ReceivedPDUDiscards              |       | Mandatory   |  |
| PDUDelivers                      |       | Mandatory   |  |
| SentPDUs                         |       | Mandatory   |  |
| SentPDUDiscards                  |       | Mandatory   |  |
| SentPDUNoRouteErrors             |       | Mandatory   |  |
| ReassemblyRequired               |       | Mandatory   |  |
| ReassembledPackets               |       | Mandatory   |  |
| ReassemblyFailed                 |       | Mandatory   |  |
| Fragmentation                    |       | Mandatory   |  |
| FragmentationFails               |       | Mandatory   |  |

**Table 177: SMI Referenced Properties/Methods for CIM\_IPEndpointStatistics**

| Properties            | Flags | Requirement | Description & Notes |
|-----------------------|-------|-------------|---------------------|
| FragmentedPDUsCreates |       | Mandatory   |                     |
| RouteEntriesDiscards  |       | Mandatory   |                     |
| ResetSelectedStats()  |       | Optional    |                     |

**15.8.33 CIM\_IPProtocolEndpoint**

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 178 describes class CIM\_IPProtocolEndpoint.

**Table 178: SMI Referenced Properties/Methods for CIM\_IPProtocolEndpoint**

| Properties              | Flags | Requirement | Description & Notes |
|-------------------------|-------|-------------|---------------------|
| SystemCreationClassName |       | Mandatory   |                     |
| SystemName              |       | Mandatory   |                     |
| CreationClassName       |       | Mandatory   |                     |
| Name                    |       | Mandatory   |                     |
| NameFormat              |       | Mandatory   |                     |
| IPv4Address             |       | Mandatory   |                     |
| IPv6Address             |       | Mandatory   |                     |
| SubnetMask              |       | Mandatory   |                     |
| PrefixLength            |       | Optional    |                     |
| ProtocolIFType          |       | Mandatory   | IPv4, IPv6, IPv4/v6 |

**15.8.34 CIM\_IPSettings**

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 179 describes class CIM\_IPSettings.

**Table 179: SMI Referenced Properties/Methods for CIM\_IPSettings**

| Properties           | Flags | Requirement | Description & Notes  |
|----------------------|-------|-------------|--|
| InstanceID           |       | Mandatory   | Opaque   |
| ElementName          |       | Optional    | User friendly name. In addition, it can be used as a index property for a search or query. |
| FragmentationTimeout |       | Mandatory   |  |
| EnableIPForwarding   |       | Mandatory   |  |

### 15.8.35 CIM\_Network

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 180 describes class CIM\_Network.

**Table 180: SMI Referenced Properties/Methods for CIM\_Network**

| Properties        | Flags | Requirement | Description & Notes |
|-------------------|-------|-------------|---------------------|
| CreationClassName |       | Mandatory   |                     |
| Name              |       | Mandatory   | IP Address          |
| NameFormat        |       | Mandatory   | IP Address          |

### 15.8.36 CIM\_NetworkPipe

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 181 describes class CIM\_NetworkPipe.

**Table 181: SMI Referenced Properties/Methods for CIM\_NetworkPipe**

| Properties          | Flags | Requirement | Description & Notes |
|---------------------|-------|-------------|---------------------|
| InstanceID          |       | Mandatory   |                     |
| Directionality      |       | Optional    |                     |
| OperationalStatus   |       | Optional    |                     |
| AggregationBehavior |       | Optional    |                     |
| EnabledState        |       | Optional    |                     |
| RequestedState      |       | Optional    |                     |

### 15.8.37 CIM\_NetworkPipeComposition

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 182 describes class CIM\_NetworkPipeComposition.

**Table 182: SMI Referenced Properties/Methods for CIM\_NetworkPipeComposition**

| Properties          | Flags | Requirement | Description & Notes |
|---------------------|-------|-------------|---------------------|
| AggregationSequence |       | Mandatory   |                     |
| GroupComponent      |       | Mandatory   |                     |
| PartComponent       |       | Mandatory   |                     |

### 15.8.38 CIM\_DeviceSAPImplementation

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 183 describes class CIM\_DeviceSAPImplementation.

**Table 183: SMI Referenced Properties/Methods for CIM\_DeviceSAPImplementation**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Dependent  |       | Mandatory   |                     |
| Antecedent |       | Mandatory   |                     |
| Dependent  |       | Mandatory   |                     |
| Antecedent |       | Mandatory   |                     |

#### 15.8.39 CIM\_ProtocolEndpoint

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 184 describes class CIM\_ProtocolEndpoint.

**Table 184: SMI Referenced Properties/Methods for CIM\_ProtocolEndpoint**

| Properties                  | Flags | Requirement | Description & Notes |
|-----------------------------|-------|-------------|---------------------|
| SystemCreationClass<br>Name |       | Mandatory   |                     |
| SystemName                  |       | Mandatory   |                     |
| CreationClassName           |       | Mandatory   |                     |
| Name                        |       | Mandatory   |                     |
| NameFormat                  |       | Mandatory   |                     |
| ProtocolIFType              |       | Mandatory   | Fibrechannel, Fcip  |

#### 15.8.40 CIM\_LANEndpoint

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 185 describes class CIM\_LANEndpoint.

**Table 185: SMI Referenced Properties/Methods for CIM\_LANEndpoint**

| Properties              | Flags | Requirement | Description & Notes |
|-------------------------|-------|-------------|---------------------|
| SystemCreationClassName |       | Mandatory   |                     |
| SystemName              |       | Mandatory   |                     |
| CreationClassName       |       | Mandatory   |                     |
| Name                    |       | Mandatory   |                     |
| NameFormat              |       | Mandatory   |                     |
| ProtocolIFType          |       | Mandatory   | Fibrechannel, Fcip  |

#### 15.8.41 CIM\_RemotePort

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Optional

Table 186 describes class CIM\_RemotePort.

**Table 186: SMI Referenced Properties/Methods for CIM\_RemotePort**

| Properties                 | Flags | Requirement | Description & Notes    |
|----------------------------|-------|-------------|------------------------|
| SystemName                 |       | Mandatory   |                        |
| CreationClassName          |       | Mandatory   |                        |
| Name                       |       | Mandatory   | Opaque                 |
| AccessInfo                 |       | Mandatory   |                        |
| InfoFormat                 |       | Mandatory   |                        |
| OtherInfoFormatDescription |       | Mandatory   | WWN                    |
| PortInfo                   |       | Optional    | WWN or TCP port number |
| PortProtocol               |       | Mandatory   |                        |
| OtherProtocolDescription   |       | Mandatory   |                        |

#### 15.8.42 CIM\_RemoteServiceAccessPoint



Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Optional

Table 187 describes class CIM\_RemoteServiceAccessPoint.

**Table 187: SMI Referenced Properties/Methods for CIM\_RemoteServiceAccessPoint**

| Properties                 | Flags | Requirement | Description & Notes          |
|----------------------------|-------|-------------|------------------------------|
| SystemCreationClassName    |       | Mandatory   |                              |
| SystemName                 |       | Mandatory   |                              |
| CreationClassName          |       | Mandatory   |                              |
| Name                       |       | Mandatory   |                              |
| AccessInfo                 |       | Mandatory   |                              |
| InfoFormat                 |       | Mandatory   | IPv4 Address OR IPv6 Address |
| OtherInfoFormatDescription |       | Optional    |                              |

#### 15.8.43 CIM\_SystemDevice (System to EthernetPort)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 188 describes class CIM\_SystemDevice (System to EthernetPort).

**Table 188: SMI Referenced Properties/Methods for CIM\_SystemDevice (System to EthernetPort)**

| Properties     | Flags | Requirement | Description & Notes |
|----------------|-------|-------------|---------------------|
| PartComponent  |       | Mandatory   |                     |
| GroupComponent |       | Mandatory   |                     |

#### 15.8.44 CIM\_SystemDevice (System to FCPort)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 189 describes class CIM\_SystemDevice (System to FCPort).

**Table 189: SMI Referenced Properties/Methods for CIM\_SystemDevice (System to FCPort)**

| Properties     | Flags | Requirement | Description & Notes |
|----------------|-------|-------------|---------------------|
| PartComponent  |       | Mandatory   |                     |
| GroupComponent |       | Mandatory   |                     |

#### 15.8.45 CIM\_TCPEndpointStatistics

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 190 describes class CIM\_TCPEndpointStatistics.

**Table 190: SMI Referenced Properties/Methods for CIM\_TCPEndpointStatistics**

| Properties               | Flags | Requirement | Description & Notes  |
|--------------------------|-------|-------------|--|
| InstanceID               |       | Mandatory   |  |
| ElementName              |       | Optional    | User friendly name. In addition, it can be used as a index property for a search or query. |
| StatisticTime            |       | Mandatory   |  |
| ReceivedSegmentsIn Error |       | Mandatory   |  |
| SentResetSegments        |       | Mandatory   |  |
| ResetSelectedStats()     |       | Optional    |  |

#### 15.8.46 CIM\_TCPProtocolEndpoint

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 191 describes class CIM\_TCPProtocolEndpoint.

**Table 191: SMI Referenced Properties/Methods for CIM\_TCPProtocolEndpoint**

| Properties              | Flags | Requirement | Description & Notes |
|-------------------------|-------|-------------|---------------------|
| SystemCreationClassName |       | Mandatory   |                     |
| SystemName              |       | Mandatory   |                     |
| CreationClassName       |       | Mandatory   |                     |
| Name                    |       | Mandatory   |                     |
| NameFormat              |       | Optional    |                     |
| PortNumber              |       | Optional    |                     |
| ProtocolIFType          |       | Mandatory   |                     |

#### 15.8.47 CIM\_TCPSettings

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 192 describes class CIM\_TCPSettings.

**Table 192: SMI Referenced Properties/Methods for CIM\_TCPSettings**

| Properties                     | Flags | Requirement | Description & Notes  |
|--------------------------------|-------|-------------|--|
| InstanceID                     |       | Mandatory   | Opaque   |
| ElementName                    |       | Optional    | User friendly name. In addition, it can be used as a index property for a search or query. |
| RetransmissionTimeoutAlgorithm |       | Mandatory   |  |
| RetransmissionTimeoutMin       |       | Mandatory   |  |
| RetransmissionTimeoutMax       |       | Mandatory   |  |

#### 15.8.48 CIM\_TCPStatisticalData

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 193 describes class CIM\_TCPStatisticalData.

**Table 193: SMI Referenced Properties/Methods for CIM\_TCPStatisticalData**

| Properties              | Flags | Requirement | Description & Notes |
|-------------------------|-------|-------------|---------------------|
| InstanceID              |       | Mandatory   | Opaque              |
| ElementName             |       | Optional    | User Friendly Name  |
| StatisticTime           |       | Mandatory   |                     |
| ActiveOpenConnections   |       | Mandatory   |                     |
| PassiveOpenConnections  |       | Mandatory   |                     |
| AttemptsFails           |       | Mandatory   |                     |
| EstablishedResets       |       | Mandatory   |                     |
| EstablishedConnections  |       | Mandatory   |                     |
| ReceivedSegments        |       | Mandatory   |                     |
| SentSegments            |       | Mandatory   |                     |
| RetransmittedSegments   |       | Mandatory   |                     |
| ReceivedSegmentsInError |       | Mandatory   |                     |
| SentResetSegments       |       | Mandatory   |                     |
| ResetSelectedStats()    |       | Mandatory   |                     |

**EXPERIMENTAL**

---

DEPRECATED

### **Clause 16: Router Profile**

The functionality of the Router Profile (Section 7.3.4.3 of SMI-S 1.0.2) modeled FC/SCSI routers used in some tape libraries. Similar functionality is included in the Storage Media Library Profile (see *Storage Management Technical Specification, Part 8 Media Libraries* Clause 4: Storage Library Profile).

DEPRECATED

---



---

---

## EXPERIMENTAL

### Clause 17: iSCSI Gateway Profile

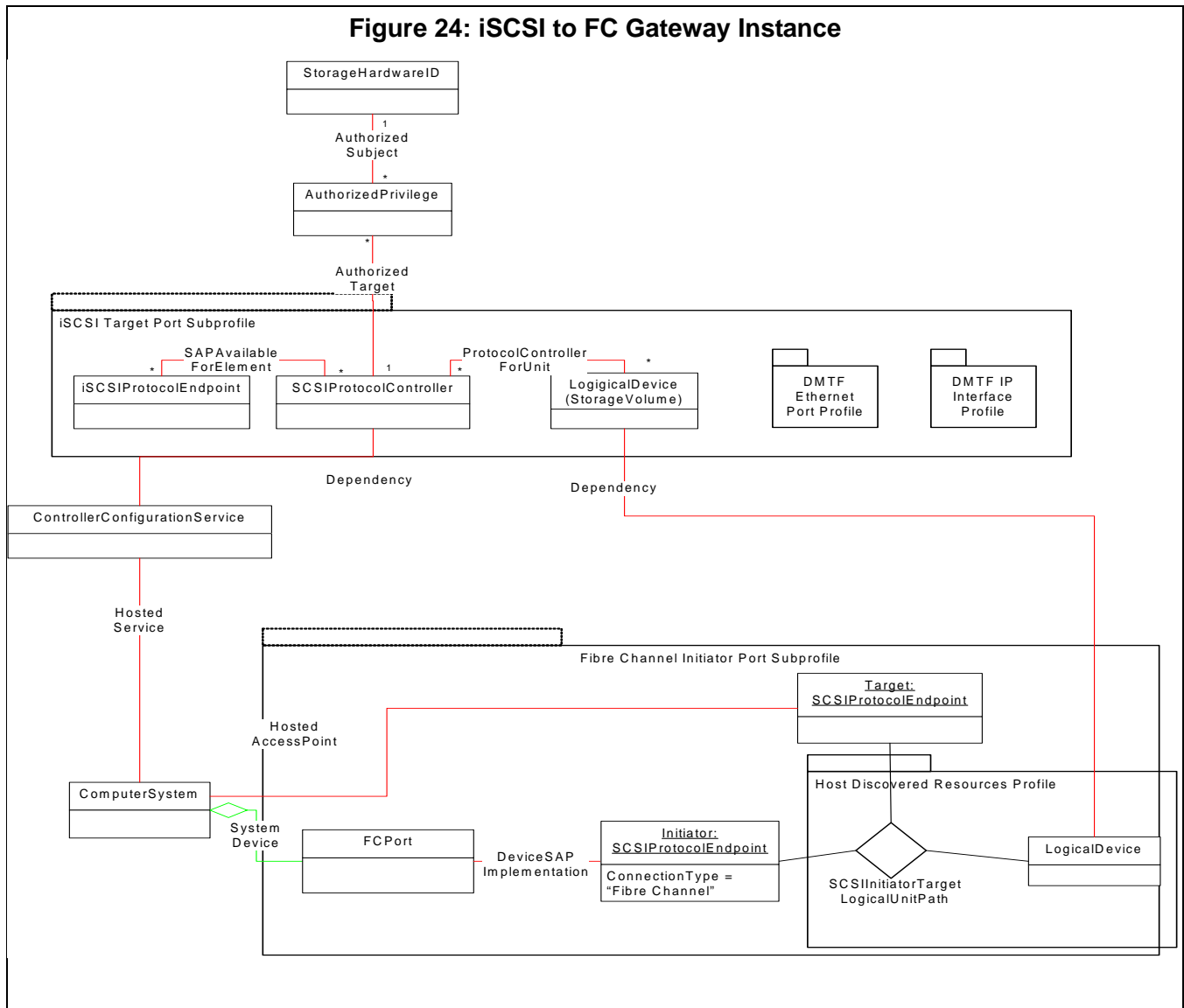
#### 17.1 Description

This profile exposes FC LUNs through an iSCSI Node. It includes the discovery of the exposed FC LUNs to the iSCSI LUN (mapping) and the configuration of the mapping. The mapping uses the methods `ExposePaths` and `HidePaths` in `StorageConfigurationService`, but instead of exposing the LUN to a Host, it is exposing a LUN from a Fibre Channel Fabric to an iSCSI Fabric.

It supports the profile of the:

- 1) iSCSI Target Ports Subprofile which presents discovery and configuration of the iSCSI nodes and iSCSI Ports (Target Portal Groups) and the discovery and configuration of exposing storage volumes through iSCSI Nodes.
- 2) Fibre Channel Initiator Port Subprofile which presents the FC Initiator and attached Volume.
- 3) Host Discovered Resources Profile which optionally presents the storage accessible through a particular FC Initiator.
- 4) DMTF Ethernet Port Profile which exposes the discovery of an Ethernet Port.
- 5) DMTF IP Interface Profile which exposes the discovery and configuration of an IP Interface.
- 6) FC Target Volume is identified as the concatenation of the FC Target Port and LUN.

In iSCSI to FC Gateway Instance (Figure 24:), the instance diagram for the iSCSI to FC Gateway is shown.

**Figure 24: iSCSI to FC Gateway Instance**

## 17.2 Health and Fault Management Consideration

None

## 17.3 Cascading Considerations

None



## 17.4 Supported Profiles, Subprofiles, and Packages

**Table 194: Supported Profiles for iSCSI to FC Gateway**

| Registered Profile Names  | Mandatory | Version |
|---------------------------|-----------|---------|
| Access Points             | No        | 1.2.0   |
| iSCSI Target Ports        | Yes       | 1.2.0   |
| FC Initiator Ports        | Yes       | 1.2.0   |
| Ethernet Port             | Yes       | 1.0     |
| IP Interface              | Yes       | 1.0     |
| Host Discovered Resources | No        | 1.2.0   |

## 17.5 Methods of the Profile

The mapping uses the methods `ExposePaths` and `HidePaths` in `ControllerConfigurationService` to expose or hide the LUN from a Fibre Channel Fabric to an iSCSI Fabric.

### 17.5.0.0.1 HidePaths

```
uint32 HidePaths (
    OUT CIM_ConcreteJob REF Job
    IN string LUNames[]
    IN string InitiatorPortIDs[]
    IN string TargetPortIDs[]
    IN/OUT CIM_SCSIProtocolController REF ProtocolControllers[] )
```

### 17.5.0.0.2 ExposePaths

```
Uint32 ExposePaths (
    OUT CIM_ConcreteJob REF Job,
    IN string LUNames[]
    IN string InitiatorPortIDs[]
    IN string TargetPortIDs[]
    IN string DeviceNumber[]
    IN uint16 DeviceAccess[]
    IN/OUT CIM_SCSIProtocolController REF ProtocolControllers[])
```

## 17.6 Client Considerations and Recipes

None

## 17.7 Registered Name and Version

iSCSI to FC Gateway version 1.2.0

## 17.8 CIM Elements

**Table 195: CIM Elements for iSCSI to FC Gateway**

| Element Name   | Requirement | Description |
|--|-------------|-------------|
| CIM_AuthorizedPrivilege (17.8.1)   | Mandatory   |             |
| CIM_AuthorizedSubject (17.8.2)   | Mandatory   |             |
| CIM_AuthorizedTarget (17.8.3)  | Mandatory   |             |
| CIM_Dependency (Associates ControllerConfigurationService and ProtocolController) (17.8.4) | Mandatory   |             |
| CIM_Dependency (Associates PrivilegeManagementService and AuthorizedPrivilege) (17.8.5)    | Mandatory   |             |
| CIM_ControllerConfigurationService (17.8.6)  | Mandatory   |             |
| CIM_HostedService (Associates ComputerSystem and ControllerConfigurationService) (17.8.7)  | Mandatory   |             |
| CIM_HostedService (Associates ComputerSystem and PrivilegeManagementService) (17.8.8)      | Mandatory   |             |
| CIM_LogicalDevice (17.8.9)   | Mandatory   |             |
| CIM_PrivilegeManagementService (17.8.10)   | Mandatory   |             |
| CIM_SCSIProtocolController (17.8.11)   | Mandatory   |             |
| CIM_ProtocolControllerForUnit (17.8.12)  | Mandatory   |             |
| CIM_SAPAvailableForElement (17.8.13)   | Mandatory   |             |
| CIM_SCSIProtocolEndpoint (17.8.14)   | Mandatory   |             |
| CIM_StorageHardwareID (17.8.15)  | Mandatory   |             |
| CIM_ControllerConfigurationService (17.8.16)   | Mandatory   |             |
| CIM_StorageHardwareIDManagementService (17.8.17)   | Mandatory   |             |

### 17.8.1 CIM\_AuthorizedPrivilege

Created By: Extrinsic: CIM\_ControllerConfigurationService.ExposePaths,  
CIM\_ControllerConfigurationService.HidePaths

Modified By: Extrinsic: CIM\_ControllerConfigurationService.ExposePaths,  
CIM\_ControllerConfigurationService.HidePaths

Deleted By: Extrinsic: CIM\_ControllerConfigurationService.ExposePaths,  
CIM\_ControllerConfigurationService.HidePaths

Class Mandatory: Mandatory

Table 196 describes class CIM\_AuthorizedPrivilege.

**Table 196: SMI Referenced Properties/Methods for CIM\_AuthorizedPrivilege**

| Properties       | Flags | Requirement | Description & Notes                          |
|------------------|-------|-------------|--|
| InstanceID       |       | Mandatory   | Opaque and unique identifier                 |
| ElementName      |       | Optional    | User friendly name                           |
| PrivilegeGranted |       | Mandatory   | Indicates if the privilege is granted or not |
| Activities       |       | Mandatory   | For SMI-S, must be "Read", "Write"           |

### 17.8.2 CIM\_AuthorizedSubject

Created By: Extrinsic: CIM\_ControllerConfigurationService.ExposePaths,  
CIM\_ControllerConfigurationService.HidePaths

Modified By: Extrinsic: CIM\_ControllerConfigurationService.ExposePaths,  
CIM\_ControllerConfigurationService.HidePaths

Deleted By: Extrinsic: CIM\_ControllerConfigurationService.ExposePaths,  
CIM\_ControllerConfigurationService.HidePaths

Class Mandatory: Mandatory

Table 197 describes class CIM\_AuthorizedSubject.

**Table 197: SMI Referenced Properties/Methods for CIM\_AuthorizedSubject**

| Properties        | Flags | Requirement | Description & Notes   |
|-------------------|-------|-------------|---|
| PrivilegedElement |       | Mandatory   | The Subject for which Privileges are granted or denied  |
| Privilege         |       | Mandatory   | The Privilege either granted or denied to an Identity or group of Identities collected by a Role. |

### 17.8.3 CIM\_AuthorizedTarget

Created By: Extrinsic: CIM\_ControllerConfigurationService.ExposePaths,  
CIM\_ControllerConfigurationService.HidePaths

Modified By: Extrinsic: CIM\_ControllerConfigurationService.ExposePaths,  
CIM\_ControllerConfigurationService.HidePaths

Deleted By: Extrinsic: CIM\_ControllerConfigurationService.ExposePaths,  
CIM\_ControllerConfigurationService.HidePaths

Class Mandatory: Mandatory

Table 198 describes class CIM\_AuthorizedTarget.

**Table 198: SMI Referenced Properties/Methods for CIM\_AuthorizedTarget**

| Properties    | Flags | Requirement | Description & Notes  |
|---------------|-------|-------------|--|
| TargetElement |       | Mandatory   | The target set of resources to which the Privilege applies |
| Privilege     |       | Mandatory   | The Privilege affecting the target resource                |

#### 17.8.4 CIM\_Dependency (Associates ControllerConfigurationService and ProtocolController)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 199 describes class CIM\_Dependency (Associates ControllerConfigurationService and ProtocolController).

**Table 199: SMI Referenced Properties/Methods for CIM\_Dependency (Associates ControllerConfigurationService and ProtocolController)**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Dependent  |       | Mandatory   |                     |
| Antecedent |       | Mandatory   |                     |

#### 17.8.5 CIM\_Dependency (Associates PrivilegeManagementService and AuthorizedPrivilege)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 200 describes class CIM\_Dependency (Associates PrivilegeManagementService and AuthorizedPrivilege).

**Table 200: SMI Referenced Properties/Methods for CIM\_Dependency (Associates PrivilegeManagementService and AuthorizedPrivilege)**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Antecedent |       | Mandatory   |                     |

**Table 200: SMI Referenced Properties/Methods for CIM\_Dependency (Associates PrivilegeManagementService and AuthorizedPrivilege)**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Dependent  |       | Mandatory   |                     |

**17.8.6 CIM\_ControllerConfigurationService**

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 201 describes class CIM\_ControllerConfigurationService.

**Table 201: SMI Referenced Properties/Methods for CIM\_ControllerConfigurationService**

| Properties                 | Flags | Requirement | Description & Notes                  |
|----------------------------|-------|-------------|--------------------------------------|
| SystemCreationClassName    |       | Mandatory   | The scoping System CreationClassName |
| SystemName                 |       | Mandatory   | The scoping System Name              |
| CreationClassName          |       | Mandatory   | The name of the concrete subclass    |
| Name                       |       | Mandatory   | Unique identifier for the Service    |
| ExposePaths()              |       | Mandatory   |                                      |
| HidePaths()                |       | Mandatory   |                                      |
| ExposeDefaultLUs()         |       | Mandatory   |                                      |
| HideDefaultLUs()           |       | Mandatory   |                                      |
| DeleteProtocolController() |       | Mandatory   |                                      |

**17.8.7 CIM\_HostedService (Associates ComputerSystem and ControllerConfigurationService)**

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 202 describes class CIM\_HostedService (Associates ComputerSystem and ControllerConfigurationService).

**Table 202: SMI Referenced Properties/Methods for CIM\_HostedService (Associates Computer-System and ControllerConfigurationService)**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Antecedent |       | Mandatory   |                     |
| Dependent  |       | Mandatory   |                     |

#### 17.8.8 CIM\_HostedService (Associates ComputerSystem and PrivilegeManagementService)

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 203 describes class CIM\_HostedService (Associates ComputerSystem and PrivilegeManagementService).

**Table 203: SMI Referenced Properties/Methods for CIM\_HostedService (Associates Computer-System and PrivilegeManagementService)**

| Properties | Flags | Requirement | Description & Notes |
|------------|-------|-------------|---------------------|
| Antecedent |       | Mandatory   |                     |
| Dependent  |       | Mandatory   |                     |

#### 17.8.9 CIM\_LogicalDevice

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 204 describes class CIM\_LogicalDevice.

**Table 204: SMI Referenced Properties/Methods for CIM\_LogicalDevice**

| Properties              | Flags | Requirement | Description & Notes                  |
|-------------------------|-------|-------------|--------------------------------------|
| SystemCreationClassName |       | Mandatory   | The scoping System CreationClassName |
| CreationClassName       |       | Mandatory   | The name of the concrete subclass    |

**Table 204: SMI Referenced Properties/Methods for CIM\_LogicalDevice**

| Properties | Flags | Requirement | Description & Notes     |
|------------|-------|-------------|-------------------------|
| SystemName |       | Mandatory   | The scoping System Name |
| DeviceID   |       | Mandatory   | Unique identifier       |

**17.8.10 CIM\_PrivilegeManagementService**

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 205 describes class CIM\_PrivilegeManagementService.

**Table 205: SMI Referenced Properties/Methods for CIM\_PrivilegeManagementService**

| Properties              | Flags | Requirement | Description & Notes                  |
|-------------------------|-------|-------------|--------------------------------------|
| SystemCreationClassName |       | Mandatory   | The scoping System CreationClassName |
| CreationClassName       |       | Mandatory   | The name of the concrete subclass    |
| SystemName              |       | Mandatory   | The scoping System Name              |
| Name                    |       | Mandatory   | Uniquely identifies the Service      |
| ElementName             |       | Mandatory   | User friendly name                   |
| AssignAccess()          |       | Mandatory   |                                      |
| RemoveAccess()          |       | Mandatory   |                                      |

**17.8.11 CIM\_SCSIProtocolController**

Created By: Extrinsic: CIM\_ControllerConfigurationService.ExposePaths,  
CIM\_ControllerConfigurationService.HidePaths

Modified By: Extrinsic: CIM\_ControllerConfigurationService.ExposePaths,  
CIM\_ControllerConfigurationService.HidePaths

Deleted By: Extrinsic: CIM\_ControllerConfigurationService.ExposePaths,  
CIM\_ControllerConfigurationService.HidePaths

Class Mandatory: Mandatory

Table 206 describes class CIM\_SCSIProtocolController.

**Table 206: SMI Referenced Properties/Methods for CIM\_SCSIProtocolController**

| Properties              | Flags | Requirement | Description & Notes                    |
|-------------------------|-------|-------------|--|
| SystemCreationClassName |       | Mandatory   | The scoping System CreationClassName   |
| CreationClassName       |       | Mandatory   | The name of the concrete subclass      |
| SystemName              |       | Mandatory   | The scoping System's Name              |
| DeviceID                |       | Mandatory   | Unique name for the ProtocolController |

#### 17.8.12 CIM\_ProtocolControllerForUnit

Created By: Extrinsic: CIM\_ControllerConfigurationService.ExposePaths, CIM\_ControllerConfigurationService.HidePaths, CIM\_ControllerConfigurationService.ExposeDefaultLUs, CIM\_ControllerConfigurationService.HideDefaultLUs

Modified By: Extrinsic: CIM\_ControllerConfigurationService.ExposePaths, CIM\_ControllerConfigurationService.HidePaths, CIM\_ControllerConfigurationService.ExposeDefaultLUs, CIM\_ControllerConfigurationService.HideDefaultLUs

Deleted By: Extrinsic: CIM\_ControllerConfigurationService.ExposePaths, CIM\_ControllerConfigurationService.HidePaths, CIM\_ControllerConfigurationService.ExposeDefaultLUs, CIM\_ControllerConfigurationService.HideDefaultLUs

Class Mandatory: Mandatory

Table 207 describes class CIM\_ProtocolControllerForUnit.

**Table 207: SMI Referenced Properties/Methods for CIM\_ProtocolControllerForUnit**

| Properties   | Flags | Requirement | Description & Notes  |
|--------------|-------|-------------|--|
| DeviceNumber |       | Mandatory   | Address (e.g. LUN) of the associated Device. Shall be formatted as unseparated uppercase hexadecimal digits, with no leading 0x. |
| DeviceAccess |       | Mandatory   | The access rights granted to the referenced logical unit as exposed through referenced ProtocolController                        |
| Antecedent   |       | Mandatory   |  |
| Dependent    |       | Mandatory   |  |

#### 17.8.13 CIM\_SAPAvailableForElement

Created By: Extrinsic: CIM\_ControllerConfigurationService.ExposePaths, CIM\_ControllerConfigurationService.HidePaths, CIM\_ControllerConfigurationService.ExposeDefaultLUs, CIM\_ControllerConfigurationService.HideDefaultLUs



Modified By: Extrinsic: CIM\_ControllerConfigurationService.ExposePaths,  
 CIM\_ControllerConfigurationService.HidePaths, CIM\_ControllerConfigurationService.ExposeDefaultLUs,  
 CIM\_ControllerConfigurationService.HideDefaultLUs

Deleted By: Extrinsic: CIM\_ControllerConfigurationService.ExposePaths,  
 CIM\_ControllerConfigurationService.HidePaths, CIM\_ControllerConfigurationService.ExposeDefaultLUs,  
 CIM\_ControllerConfigurationService.HideDefaultLUs

Class Mandatory: Mandatory

Table 208 describes class CIM\_SAPAvailableForElement.

**Table 208: SMI Referenced Properties/Methods for CIM\_SAPAvailableForElement**

| Properties     | Flags | Requirement | Description & Notes |
|----------------|-------|-------------|---------------------|
| AvailableSAP   |       | Mandatory   |                     |
| ManagedElement |       | Mandatory   |                     |

#### 17.8.14 CIM\_SCSIProtocolEndpoint

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 209 describes class CIM\_SCSIProtocolEndpoint.

**Table 209: SMI Referenced Properties/Methods for CIM\_SCSIProtocolEndpoint**

| Properties                  | Flags | Requirement | Description & Notes                  |
|-----------------------------|-------|-------------|--------------------------------------|
| SystemCreationClass<br>Name |       | Mandatory   | The scoping System CreationClassName |
| SystemName                  |       | Mandatory   | The scoping System Name              |
| CreationClassName           |       | Mandatory   | The name of the concrete subclass    |

#### 17.8.15 CIM\_StorageHardwareID

Created By: Extrinsic: CIM\_StorageHardwareIDManagementService.CreateStorageHardwareID

Modified By: Static

Deleted By: Extrinsic: CIM\_StorageHardwareIDManagementService.DeleteStorageHardwareID

Class Mandatory: Mandatory

Table 210 describes class CIM\_StorageHardwareID.

**Table 210: SMI Referenced Properties/Methods for CIM\_StorageHardwareID**

| Properties | Flags | Requirement | Description & Notes  |
|------------|-------|-------------|--|
| InstanceID |       | Mandatory   | Opaque and unique identifier   |
| StorageID  |       | Mandatory   | The worldwide unique ID  |
| IDType     |       | Mandatory   | StorageID type. Values are Other, PortWWN, NodeWWN, Hostname, and iSCSI Name |

#### 17.8.16 CIM\_ControllerConfigurationService

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 211 describes class CIM\_ControllerConfigurationService.

**Table 211: SMI Referenced Properties/Methods for CIM\_ControllerConfigurationService**

| Properties              | Flags | Requirement | Description & Notes                  |
|-------------------------|-------|-------------|--------------------------------------|
| SystemCreationClassName |       | Mandatory   | The scoping System CreationClassName |
| SystemName              |       | Mandatory   | The scoping System Name              |
| CreationClassName       |       | Mandatory   | The name of the concrete subclass    |
| Name                    |       | Mandatory   | Uniquely identifies the Service      |

#### 17.8.17 CIM\_StorageHardwareIDManagementService

Created By: Static

Modified By: Static

Deleted By: Static

Class Mandatory: Mandatory

Table 212 describes class CIM\_StorageHardwareIDManagementService.

**Table 212: SMI Referenced Properties/Methods for CIM\_StorageHardwareIDManagementService**

| Properties                   | Flags | Requirement | Description & Notes                  |
|------------------------------|-------|-------------|--------------------------------------|
| SystemCreationClassName      |       | Mandatory   | The scoping System CreationClassName |
| SystemName                   |       | Mandatory   | The scoping System Name              |
| CreationClassName            |       | Mandatory   | The name of the concrete subclass    |
| Name                         |       | Mandatory   | Uniquely identifies the Service      |
| CreateStorageHardwareID()    |       | Mandatory   |                                      |
| DeleteStorageHardwareID()    |       | Mandatory   |                                      |
| CreateHardwareIDCollection() |       | Optional    |                                      |
| AddHardwareIDsToCollection() |       | Optional    |                                      |

## EXPERIMENTAL

---



---

