CHAPTER 2

Cisco MDS 9000 Family CIM Server Support

SMI-S defines a number of profiles that specify the managed objects used to control and monitor elements of a SAN. The Cisco MDS 9000 Family CIM server supports the standard profiles listed in this chapter. The CIM server also supports extensions to these profiles to support features in Cisco MDS SAN-OS that are not available from the standard profiles.

This chapter includes the following sections:

Managing SANs Through SMI-S

SANs are created in a multivendor environment. Hosts, fabric elements (switches, directors), and data storage devices are integrated from different vendors to create an interoperable storage network. Managing these elements from different vendors is problematic to the network administrator. Each element has its own management interface that may be proprietary. A network administrator must work with these disparate management APIs to build a cohesive management application that controls and monitors the SAN.

The SMI-S addresses this management problem by creating a suite of flexible, open management API standards based on the vendor- and technology-independent CIM. Using the SMI-S APIs, collected in *profiles* of common management classes, a network administrator can create a simplified management application CIM client to control and monitor the disparate SAN elements that support SMI-S and CIM. With CIM servers either embedded on the SAN elements or supported by a proxy CIM server, these elements are accessible to the network administrator's CIM client application.

SMI-S uses the Service Location Protocol version 2 (SLPv2) to discover CIM servers. Once the CIM servers are identified, the CIM client determines the profiles supported on the CIM servers through the Server profile. This profile is mandatory on all SMI-S based CIM servers.

Besides the control and monitoring support provided by profiles, the CIM server also supports asynchronous delivery of events through CIM *indications*. Indications provide immediate notification of important occurrences such as when an interface goes down.

Service Location Protocol

The first step in managing a network of SAN elements with CIM servers is discovering the location and support available on the CIM servers. The SLPv2 provides this discovery mechanism. A CIM client uses SLPv2 to discover CIM servers, gathering generic information about what services the CIM servers provide and the URL where these services are located.

The Cisco MDS 9000 Family CIM server supports SLPv2 as defined in RFC 2608.

Note Cisco MDS SAN-OS Release 2.0(1b) and later support SLPv2 for the Cisco MDS 9000 Family CIM server.

Server Profile

Once the CIM client discovers the CIM servers within the SAN, the CIM client must determine the level of support each CIM server provides. The Server profile defines the capabilities of the CIM server. This includes providing the namespace and all profiles and subprofiles supported by the CIM server.

For each supported profile, the Server profile instantiates the `RegisteredProfile` class. Each instance of this class gives the CIM client the profile name and unique ID that is supported by the CIM server. Similarly, the CIM server lists all supported optional subprofiles, using the `RegisteredSubProfile` class and the `SubprofileRequiresProfile` association class to associate the subprofile with the profile.

Note Cisco MDS SAN-OS Release 2.0(1b) and later support the Server profile for the Cisco MDS 9000 Family CIM server.

Note For a Server profile instance diagram, refer to SMI-S at http://www.snia.org.

Switch Profile

The Switch profile models the physical and logical aspects of switches. The CIM client uses the Switch profile to identify that the CIM server is on a switch and uses classes in the Switch profile to identify and manage Fibre Channel ports on the switch.

The Switch profile also supports the optional Blade subprofile (see the "Blade Subprofile" section on page 2-3) and the optional Access Point subprofile (see the "Access Point Subprofile" section on page 2-4).

Note For a Switch profile instance diagram, refer to the SMI-S at http://www.snia.org.

Table 2-1 shows how to use the Switch profile classes and association classes to model the switch and ports.

Table 2-1 Using the Switch Profile

Class	How Used
ComputerSystem	Identifies the switch, with the Dedicated property set to Switch.
PhysicalElement	Identifies the physical aspects of a device.
FCPort	Identifies logical aspects of the port link and the data layers.
FCPortCapabilities	Defines configuration options supported by the ports.
FCPortStatistics	Identifies port statistics, showing real-time port traffic information for each instance of FCPort class.
FCSwitchCapabilities	Defines configuration options supported by the switch.
FCSwtichSettings	Requests configuration changes on the switch.
FCPortSettings	Requests configuration changes on the ports.

Blade Subprofile

The CIM client uses the optional Blade subprofile to model the physical and logical aspects of a supervisor module, switching module, or services module in a switch. Combining this with the Switch profile, the CIM client gains a chassis-level view into the switch, associating ports to modules and modules to a switch.

Table 2-2 shows how to use the classes and association classes to model a module.

Table 2-2 Using the Blade Subprofile

Class	How Used
LogicalModule	Identifies a supervisor module, switching module, or services module as an aggregation point for the switch ports.
ModulePort	Associates the ports to a module.

Note For a Blade subprofile instance diagram, refer to the SMI-S at http://www.snia.org.

Access Point Subprofile

The CIM client uses the Access Point subprofile to return the URL to access the switch and install or launch Fabric Manager or Device Manager. If Fabric Manager or Device Manager have not been installed, then the URL gives the option to install them. If Fabric Manager or Device Manager have been installed, then the URL gives the option to launch either of them.

Table 2-3 shows how to use the classes and association classes to model a module.

Table 2-3 Using the Access Point Subprofile

Class	How Used
`HostedAccessPoint`	Associates the `RemoteServiceAccessPoint` to the system on which it is hosted.
`RemoteServiceAccessPoint`	A `ServiceAccessPoint` for management tools. Returns the URL for the switch that can be used to install or launch Fabric Manager or Device Manager.
`SAPAvailableForElement`	Identifies the subset of devices in the system that are serviced by `RemoteServiceAccessPoint`.

Note For an Access Point subprofile instance diagram, refer to the SMI-S at http://www.snia.org.

Fabric Profile

A fabric is composed of one or more switches and network elements interconnected in a SAN. The Fabric profile models the physical and logical aspects of the fabric containing the SAN switches listed by the Switch profile.

Fabrics can contain one or more virtual SANs, or VSANs. See the "Cisco MDS Extensions to the Switch and Fabric Profiles" section on page 2-8 for more information on the Cisco VSAN extension. Because routing in the Cisco MDS 9000 Family is based on the VSAN, the `ConnectivityCollection` and `ProtocolEndpoint` classes must be associated with the VSAN, not the fabric.

Table 2-4 shows how to use the classes and association classes of the Fabric profile to model the fabric.

Table 2-4 Using the Fabric Profile

Class	How Used
`AdminDomain`	Identifies fabrics and VSANs.
`ContainedDomain`	Associates a VSAN to a fabric.
`ConnectivityCollection`	Groups a set of `ProtocolEndpoint` classes together that can communicate with each other directly and represents the foundation necessary for routing. Associates to a VSAN using the `Component` association class.
`ComputerSystem`	Represents the fabric elements that contain ports, such as switches, hosts, and storage systems. The `Dedicated` property is set to `Switch`. Associates to a VSAN using the `Component` association class.

Table 2-4 Using the Fabric Profile (continued)

Class	How Used
FCPort	Represents the logical aspects of the link and data layers. Associates to the ProtocolEndpoint class by the DeviceSAPImplementation association class and associates to the ComputerSystem class by the SystemDevice association class.
ProtocolEndpoint	Represents the higher network layers for routing. Associates to the ConnectivityCollection class by the ConnectivityMemberOfCollection association class.
ActiveConnection	Represents a link that associates two ProtocolEndpoint classes as a connection that is currently carrying traffic.

Note The Cisco MDS 9000 Family CIM server only provides the LogicalPortGroup class for the fabric, not for hosts or storage systems.

The CIM server requires that the name of the fabric or VSAN be unique within the same CIM namespace. Names are identified by the Name class property with an associated optional NameFormat property. A VSAN identifier is the VSAN identification followed by the world-wide name (WWN) of the principal switch, for example—"1_2001000530000A0A" (the NameFormat indicates that it is a WWN). For VSANs, the fabric identifier is a string because there is no principal switch per fabric.

Note For a Fabric profile instance diagram, refer to the SMI-S at http://www.snia.org.

The CIM server supports the following optional subprofiles from the Fabric profile:

- Zone Control subprofile (see the "Zone Control Subprofile" section on page 2-5)
- Enhanced Zoning and Enhanced Zoning Control subprofile (see the "Enhanced Zoning and Enhanced Zoning Control Subprofile" section on page 2-6)
- FDMI subprofile (see the "FDMI Subprofile" section on page 2-7)

The Fabric profile also supports a number of extensions specific to the Cisco MDS 9000 Family. See the "Cisco MDS Extensions to the Switch and Fabric Profiles" section on page 2-8.

Zone Control Subprofile

Zoning enables the CIM client to set up access control between storage devices or user groups. The Zone Control subprofile is a subprofile of the Fabric profile and models zoning information for the fabric. It incorporates read and write functionality including the following operations:

- Creating and deleting zones and zone sets
- Creating and deleting zone members (using ZoneMembershipSettingData)
- Adding and removing zone members to zones
- Adding and removing zones to zone sets
- Activating and deactivating a zone set

The CIM server supports all the CIM classes and association classes described by the SMI-S zoning model.

Enhanced Zoning and Enhanced Zoning Control Subprofile

The Enhanced Zoning and Enhanced Zoning Control subprofile is a subprofile of the Fabric profile and provides additional modeling of Cisco zoning information for management purposes. This includes support for the following:

- Creating and deleting zone aliases
- Adding and removing zone members to zone aliases

This subprofile supports all CIM classes and association classes described by the SMI-S zoning model except the concept of sessions for zoning.

Using the Zoning Subprofile

In the Cisco MDS CIM implementation, zoning occurs under the VSAN, not the fabric.

Note For Zoning subprofile instance diagrams, refer to the SMI-S at http://www.snia.org.

Table 2-5 shows how to use the classes and association classes of the Zoning subprofiles to model zoning.

Table 2-5 Using the Zoning Subprofile

Class	How Used
`ZoneMembershipSettingData`	Identifies zone members and indicates the member ID (defined in the CIM schema) and how the device was zoned.
`ZoneAlias`	Identifies zone aliases. Contains zone members (`ZoneMembershipSettingData` class) associated by the `ElementSettingData` association class.
`ZoneSets`	Identifies zone sets. Contains zones associated by the `MemberOfCollection` association class.
`AdminDomain`	Identifies VSANs. Only contains zone sets that are associated by the `HostedCollection` association class.
`ZoneControl`	Provides operations to control zone objects, such as creating, removing, and activating both zones and zone sets.
`ZoneService`	Manages the creation of zone sets, zones, zone aliases, and zone members, as well as activation of the zone set. The `ZoneService` class is hosted on the `CISCO_Vsan` class, which is a subclass of `AdminDomain`.
`ActiveConnection`	Represents a link that associates two `ProtocolEndpoint` classes as a connection that is currently carrying traffic.

Zones and zone sets that are active have the `Active property` set to `True` by the CIM server. Zones can only contain the following types of objects:

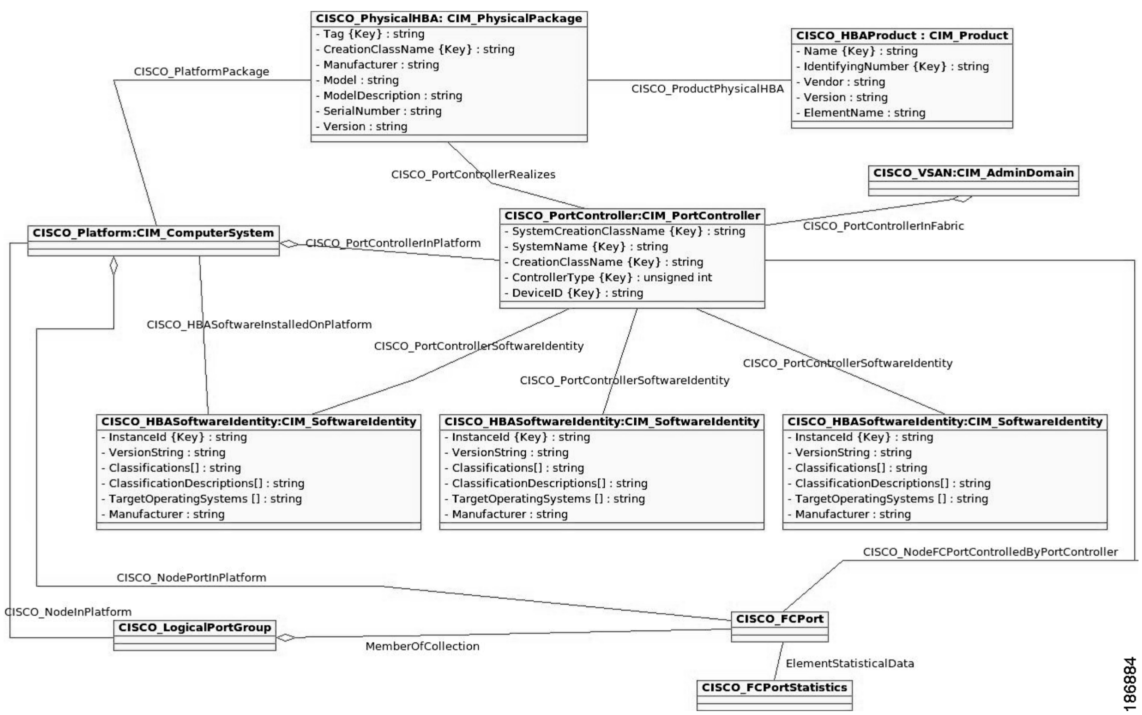
- Zone members (ZoneMembershipSettingData class) associated by the ElementSettingData association class.
- Zone aliases (ZoneAlias class; defined by SMI-S as NamedAddressCollections class) associated by the MemberOfCollection association class.

FDMI Subprofile

The Fabric Device Management Interface (FDMI) manages host bus adapters (HBA) through the Fabric and complements data in the fabric profile. It allows any entity in the fabric to expose the HBA information through the SMI without having an agent resident on the host containing the HBA. The fabric profile only addresses HBA type devices. The HBA Management Interface defined by FDMI is a subset of the interface defined by the Fibre Channel HBA API specification.

Figure 2-1 shows the FDMI subprofile instance diagram. The classes are defined in CISCO_HBA.mof. If the FDMI- enabled HBA supports the host name, then CISCO_PortController associates to a platform through CISCO_PortControllerInPlatform.If the FDMI-enabled HBA does not support the host name, then CISCO_PortController associates to a fabric, through CISCO_PortControllerInFabric.

Figure 2-1 UML Diagram for FDMI Subprofile



Using the FDMI Subprofile

In the Cisco MDS CIM implementation, the FDMI subprofile occurs under the fabric.

Note For FDMI subprofile instance diagrams, refer to the SMI-S at http://www.snia.org.

Table 2-6 shows how to use the classes and association classes of the FDMI subprofile.

Table 2-6 Using the FDMI Subprofile

Class	How Used
CISCO_PhysicalHBA	Represents FDMI enabled physical HBA card attached to a switch.
CISCO_HBAProduct	Represents product information of FDMI enabled physical HBA card attached to a switch.
CISCO_Platform	Represents a fabric-connected entity, containing one or more Node objects, that has registered with a fabric's Management Server service.
PortController	Represents the Port Controller of a FDMI enabled HBA.

Cisco MDS Extensions to the Switch and Fabric Profiles

The Cisco MDS 9000 Family CIM server supports additional classes that provide management for SAN features not covered by the standard SMI-S profiles. These extensions include:

VSAN Extensions

A VSAN is a virtual SAN that is created by partitioning the physical fabric into one or more logical fabrics. The Cisco MDS switches base routing on VSANs. The CIM client uses these VSAN extensions to identify VSANs and their associations to physical fabrics and switches.

The VSAN model in the CIM server uses the DMTF partition model. Partitioning, as defined by DMTF, is the virtual division of a single entity into multiple entities. It applies to any resource and can span namespaces and CIM object managers. Each partitioning entity manages its underlying partitions. A partitioned entity may be unaware that it is partitioned, and users may be unaware that a resource is shared. Refer to the standard partitioning model described in the CIM 2.8 schema, available from the DMTF website at http://www.dmtf.org.

Note For more information about VSANs, refer to the *Cisco MDS 9000 Family Fabric Manager Configuration Guide* or the *Cisco MDS 9000 Family CLI Configuration Guide*.

The VSAN extension provided by the Cisco MDS 9000 Family CIM server is both compatible with, and an extension of, the standard partition model. It models VSANs as a partitioned physical fabric. The E ports, F ports, PortChannels, and ports supporting FC IP and iSCSI on Cisco MDS switches all support the partitioning model.

The `HostedDependency` association class can describe the following relationships:

- Partitioning (fan in)
  - Antecedent is the partitioning entity
  - Dependent is the partitioned entity
- Clustering (fan out)

Figure 2-2 shows a UML diagram of a fabric partitioned into two VSANs. The physical switch is partitioned into two logical switches, Partitioned Switch 1 and Partitioned Switch 2. The partitions are identified as belonging to the physical switch using the `HostedDependency` association class. The VSANs are identified as belonging to the corresponding switch partitions using the `Component` association class.

Figure 2-2 UML Diagram of Fabric Partitioning

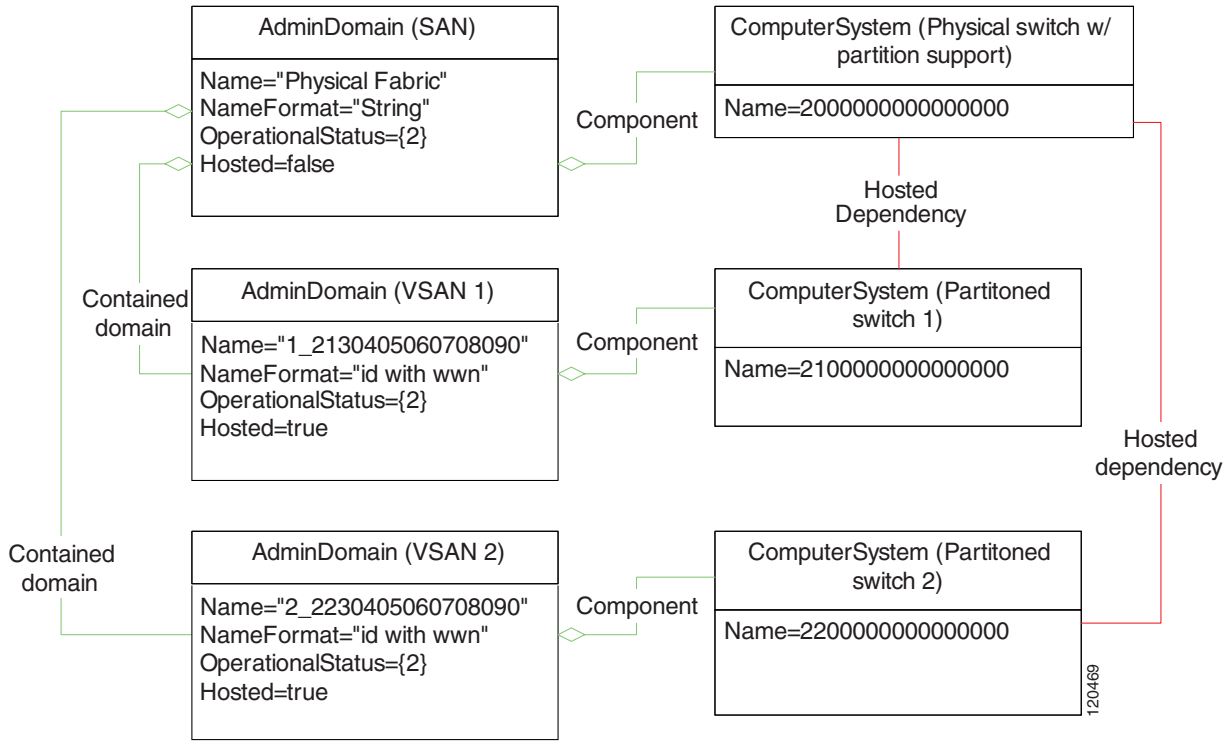
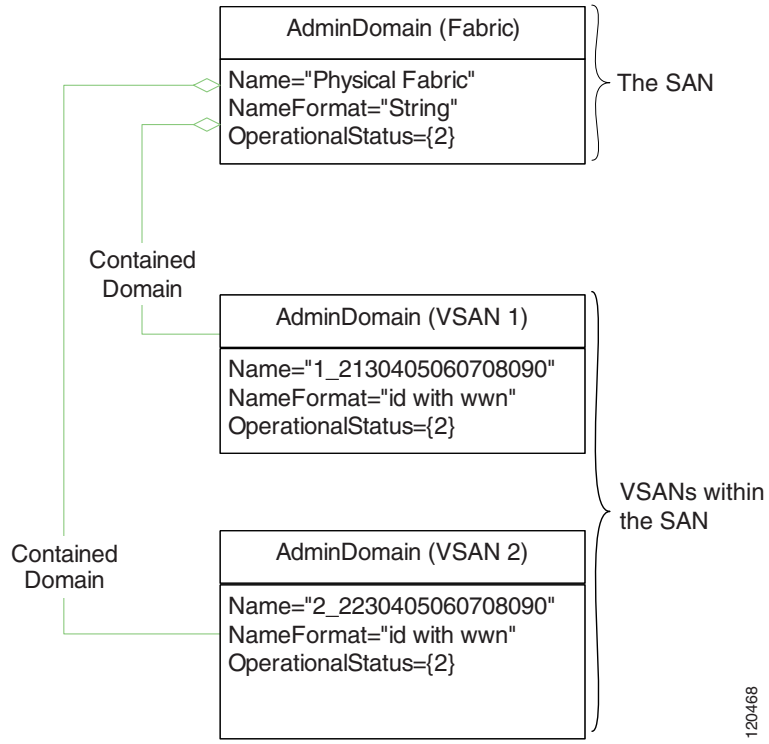


Figure 2-3 isolates the VSAN component from Figure 2-2. The physical fabric is partitioned into two VSANs, VSAN 1 and VSAN 2. Each VSAN is identified by the `AdminDomain` class. The VSANs can be identified as belonging to the physical fabric using the `ContainedDomain` association class.

Figure 2-3 VSAN Partitioning Example

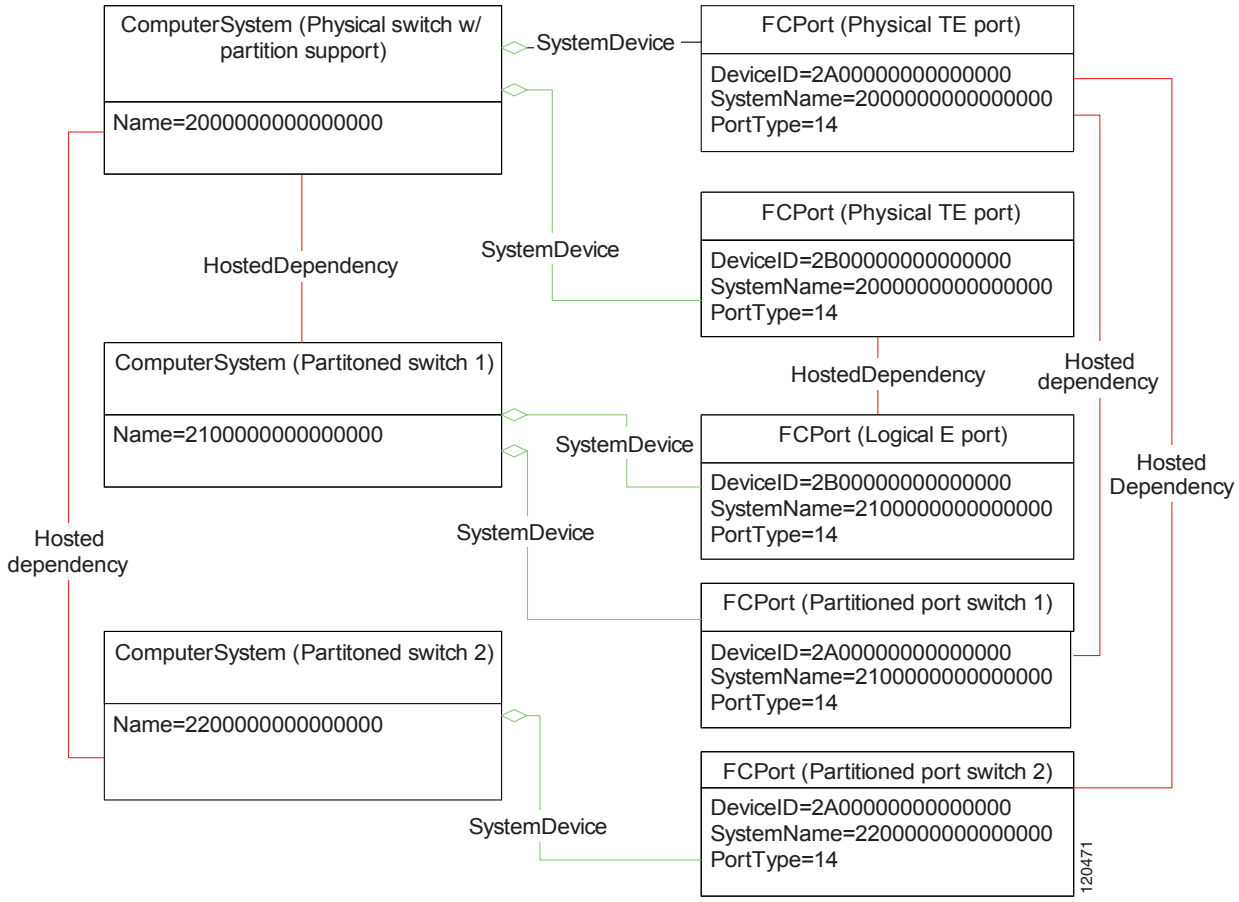


TE Port Extensions

TE ports are E ports that can carry traffic for multiple VSANs. The CIM server uses the existing fabric-to-FC port association classes to model membership of TE ports in multiple VSANs. Figure 2-4 shows the physical and logical port relationship to the switch. The two illustrated physical ports are partitioned into logical ports, and the logical ports are identified as belonging to the physical ports using the `HostedDependency` association class. A physical TE port is partitioned into two logical ports, one for Partitioned switch 1(associated to VSAN 1 in Figure 2-2) and one for Partitioned switch 2 (associated to VSAN 2 in Figure 2-2).

The physical ports are identified as components of the physical switch using the `SystemDevice` association class, and the partitioned ports are identified as components of the corresponding partitioned switch using the `SystemDevice` association class.

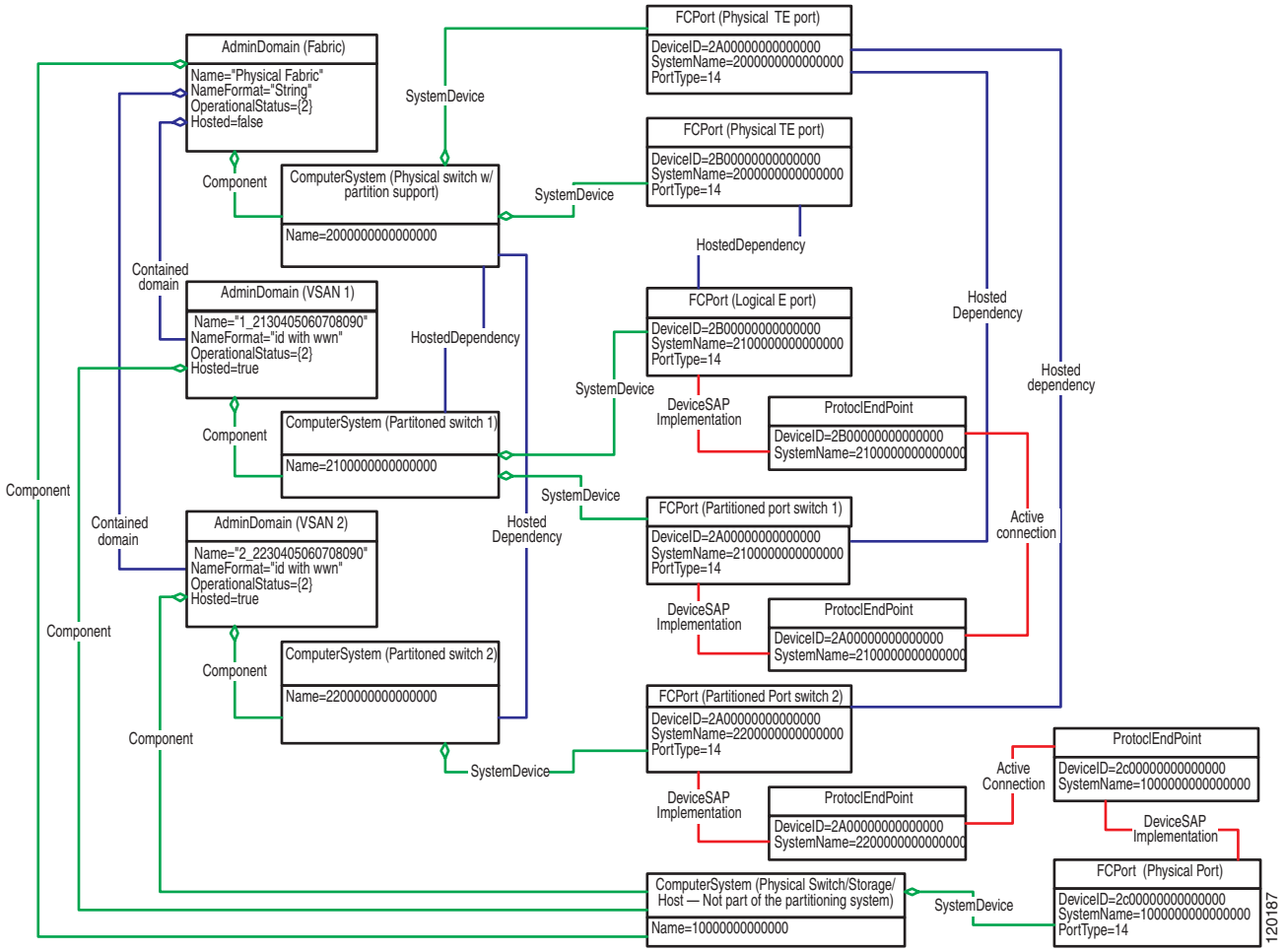
Figure 2-4 TE Port Partitioning Example



Note For more information about trunking, refer to the *Cisco MDS 9000 Family Fabric Manager Configuration Guide* or the *Cisco MDS 9000 Family CLI Configuration Guide*.

Figure 2-5 shows the full UML diagram for VSAN fabric and port partitioning in a SAN switch.

Figure 2-5 UML Diagram for VSAN Partitioning



PortChannel Extensions

A PortChannel is the aggregation of multiple physical Fibre Channel ports into one logical port to provide aggregated bandwidth, load balancing, and link redundancy. The CIM server supports a PortChannel port type in the `Cisco_FCPort` class. The `Component` association class can be used to associate individual ports with a PortChannel.

PortChannels are supported by the CIM server only for the local switch on which the CIM server is running. The CIM server also exports active connections for remote PortChannels, with two limitations:

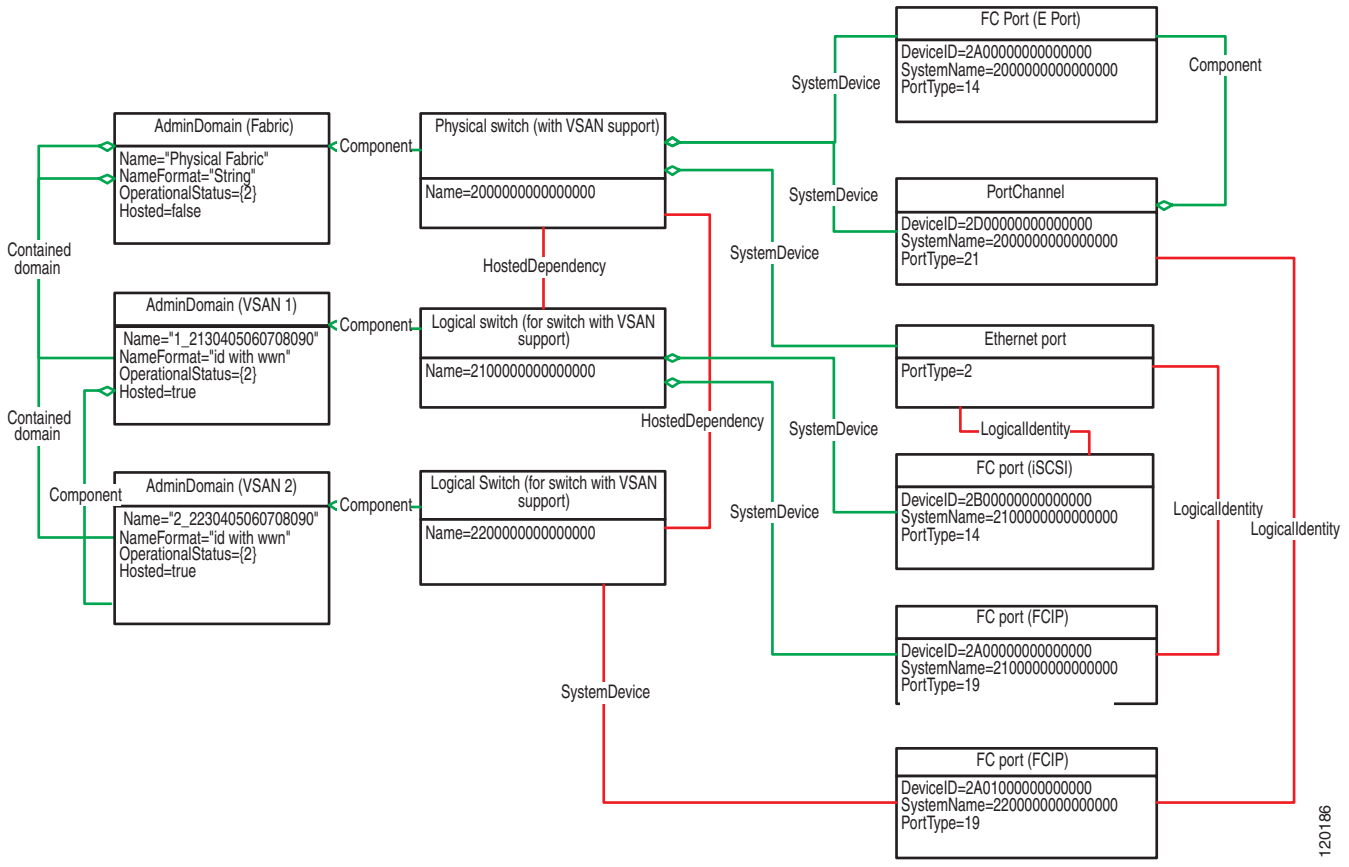
- The remote PortChannel WWN is not available; the remote switch WWN and port index are provided.
- The `Component` and `LogicalIdentity` association classes of the remote PortChannel are not available.

Note For more information about PortChannels, refer to the *Cisco MDS 9000 Family Fabric Manager Configuration Guide* or the *Cisco MDS 9000 Family CLI Configuration Guide*.

Figure 2-6 shows the relationships among ports and PortChannels on the switch that is running the CIM server. In this example:

- The PortChannels and ports are identified as belonging to the physical switch using the `SystemDevice` association class.
- The individual ports are identified as belonging to the PortChannels using the `Component` association class.

Figure 2-6 UML Instance Diagram of the Relationships Among Ports Using FCIP, PortChannels, and Ethernet Ports



FCIP Extensions

The CIM server uses the current `FCPort` class to discover information about ports supporting FCIP. For the local switch (the switch on which the CIM server is running), the CIM server uses the `LogicalIdentity` association class to link ports supporting FCIP that are on the same module.

The CIM server exports active connections for remote ports running FCIP, with two limitations:

- The WWN of the port running FCIP is not available; the remote switch WWN and port index are provided.
- The `LogicalIdentity` association class of the port running FCIP is not available.

Note For more information about FCIP, refer to the *Cisco MDS 9000 Family Fabric Manager Configuration Guide* or the *Cisco MDS 9000 Family CLI Configuration Guide*.

Figure 2-6 shows the relationships among ports running FCIP and other entities. In this example:

- The ports running FCIP are associated with other entities using the `LogicalIdentity` association class. One port running FCIP is a logical entity of an individual Ethernet port, and the other is a logical entity of a PortChannel that is comprised of Ethernet ports.
- The port running FCIP, Ethernet port, and PortChannel are identified as belonging to the physical switch using the `SystemDevice` association class.

iSCSI Extensions

You can use the current `EthernetPort` class to discover information about the port and use the `LogicalIdentity` association class to associate Gigabit Ethernet ports with iSCSI. This association class is only available for ports local to the CIM server.

Figure 2-6 shows the relationships among ports running iSCSI and other entities. In this example:

- The port running iSCSI is identified as belonging to the Ethernet port using the `LogicalIdentity` association class.
- The port running iSCSI is identified as belonging to the physical switch using the `SystemDevice` association class.

Note For more information about iSCSI, refer to the *Cisco MDS 9000 Family Fabric Manager Configuration Guide* or the *Cisco MDS 9000 Family CLI Configuration Guide*.

Fabric Profile Extensions

In addition to the standard Fabric profile, the following classes and association classes that are specific to the Cisco MDS 9000 Family are supported:

```
CISCO_ActiveConnection
CISCO_AdminDomain
CISCO_FCPort
CISCO_FCPortCapabilities
CISCO_FCPortSettings
CISCO_Vsan
CISCO_Component
CISCO_ComputerSystem
CISCO_ConnectivityCollection
CISCO_ConnectivityMemberOfCollection
CISCO_ContainedDomain
CISCO_DeviceSAPImplementation
CISCO_FCPortStatistics
CISCO_HostedAccessPoint
CISCO_HostedCollection
CISCO_ProtocolEndPoint
```

```
CISCO_PhysicalPackage
CISCO_PhysicalElement
CISCO_Product
CISCO_Realizes
CISCO_SystemDevice
CISCO_ComputerSystemPackage
CISCO_ElementStatisticalData
CISCO_LogicalPortGroup
CISCO_LogicalModule
CISCO_ModulePort
CISCO_HostedDependency
CISCO_LogicalIdentity
CISCO_PhysicalComputerSystem
CISCO_LogicalComputerSystem
CISCO_FCNodeMemberOfCollection
```

The port identifiers for the CISCO_FCPort class that are supported by the CIM server are described in Table 2-7. Port Identifiers 16004 through 16012 are Cisco extensions.

Table 2-7 Port Identifiers Supported by the Cisco MDS 9000 Family CIM Server

Port Identifier	Port Type
0	Unknown
1	Other
10	N
11	NL
12	F/NL
13	Nx
14	E
15	F
16	FL
17	B
18	G
16004(cisco specific) etc.	PortChannel
16010	FCIP
16011	iSCSI-F
16012	iSCSI-N
16000...65535	Vendor reserved

See the "Cisco Fabric MOF" section on page A-1 for the full definition of the Cisco fabric extensions.

Zoning Subprofile Extensions

In addition to the standard zoning subprofiles, the following classes and association classes that are specific to Cisco are supported:

```
CISCO_HostedService
CISCO_ZoneMemberOfCollection
CISCO_ZoneMembershipSettingData
CISCO_ZoneSet
CISCO_Zone
CISCO_ZoneCapabilities
CISCO_ZoneAlias
CISCO_ElementSettingData
CISCO_ZoneService
CISCO_SystemSpecificCollection
```

See the "Cisco Zone MOF" section on page A-4 for the full definition of the Cisco zoning extensions.

FDMI Subprofile Extensions

In addition to the standard FDMI subprofile, the following classes and association classes that are specific to the Cisco MDS 9000 Family are supported:

```
PortControllerRealizes
PlatformPackage
PortControllerSoftwareIdentity
HBASoftwareInstalledOnPlatform
NodeFCPortControlledByPortController
ProductPhysicalHBA
PlatformInFabric
NodePortInPlatform
NodeInPlatform
PortControllerInPlatform
PortControllerInFabric
```

See the "Cisco FDMI MOF" section on page A-5 for the full definition of the Cisco FDMI extensions.

CIM Indications

SMI-S provides asynchronous *indications* for changes in the CIM server or the managed elements controlled by the CIM server. These indications can inform a CIM client that:

- The SAN configuration has changed.
- The SAN switch health has degraded.
- The SAN fabric performance has degraded.
- Nameserver Database has changed.
- VSAN added/deleted/modified.
- Fan status has changed.
- Temperature status has changed.
- Power Supply status has changed.

- FRU inserted/ removed/changed.

Indications can also be used when a CIM class method is invoked that will take a long time to finish. Rather than tie up the CIM server (block) until the operation completes, the CIM server responds that the operation started, and the CIM server continues handling other requests (non-blocking). When the original, long operation completes, the CIM server sends a CIM indication asynchronously to the CIM client, showing the result of the operation. A CIM client must subscribe to indications it wants to receive from the CIM server.

The Cisco MDS 9000 Family CIM server supports the following Cisco-specific indications:

```
CISCO_LinkStateChange
CISCO_LinkUp
CISCO_Linkdown
CISCO_MediaFRUInserted
CISCO_MediaFRURemoved
CISCO_VSANChanged
CISCO_ZoneSetAlert
CISCO_EnvironmentalAlert
CISCO_FanAlert
CISCO_PowerAlert
CISCO_TempAlert
CISCO_NameServerDatabaseChanged
```

See the "Cisco Indications MOF" section on page A-15 for the Cisco Indications MOF file.