Configuring and Using Cisco DCNM SMI-S Server

This chapter provides the steps to configure Cisco DCNM SMI-S Server in Cisco DCNM products and provides some sample scenarios for using CIM objects to manage your SAN. This chapter includes the following sections:

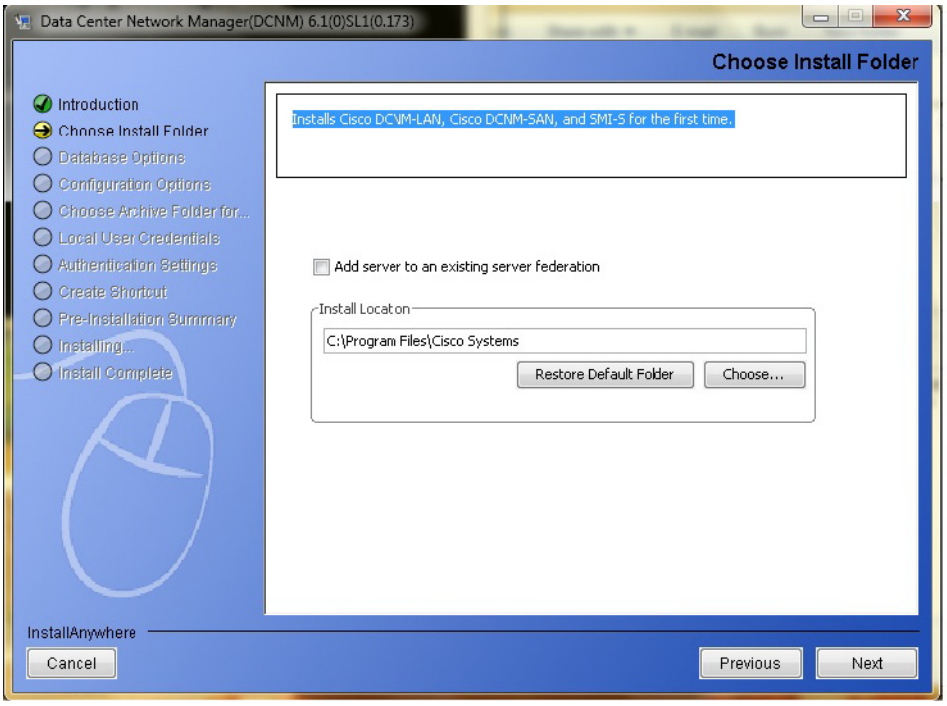
Note For information about CLI commands, refer to the *Cisco MDS 9000 Family Command Reference*.

Installing Cisco DCNM SMI-S Server

Cisco DCNM SMI-S Server is installed as part of the Cisco DCNM installation. You can use Cisco DCNM-SAN installed locally to discover the SAN fabric. For more information on discovering the fabric using Cisco DCNM-SAN client, see the *Cisco DCNM Fundamentals Guide*.

Figure 3-1 displays the SMI-S installation.

Figure 3-1 Cisco DCNM SMI-S Installation



Note All the platforms supported by Cisco DCNM are supported by SMI-S Server. SMI-S Server is configured as a startup service.

Changing the Default SMI-S Port

To change the default SMI-S port, follow these steps:

Step 1 Stop the Cisco SMI-S agent service.

Step 2 Rename **<installdir>/dcm/smis/server/jserver/bin/tcppe.properties.0** file to **<installdir>/dcm/smis/server/jserver/bin/tcppe.properties.**

Step 3 Edit **<installdir>/dcm/smis/server/jserver/bin/tcppe.properties** file to change the port by updating the property **cim-xml.http.portnumber** or **cim-xml.https.portnumber,** depending on whether DCNM is installed with http or https. Only one of this property is present in the file.

Step 4 Save the file and close it.

Step 5 Restart the Cisco SMI-S Agent service.

Step 6 SMI-S provider will use the port number assigned to the property above.

Performing Discovery and Performance Monitoring

You can use the Fabric and Switch profiles to implement discovery and performance monitoring. See the "Fan Profile" section on page 2-13 and the "FDMI Profile" section on page 2-22 for more information on these profiles.

Discovery provides information about the physical and logical entities within the SAN. This information changes dynamically as SAN entities are added, moved, or removed. Discovery also includes the discovery of object classes as well as related association classes, properties, and return status codes that are provided by servers in the managed environment.

Table 3-1 shows how to perform discovery by using the intrinsic methods defined by CIM. Use these methods to retrieve information about the switch and fabric.

Table 3-1 Performing Discovery

Method	How Used
enumerateInstances()	Enumerates instances of a CIM class.
enumerateInstanceNames()	Enumerates names of instances of a CIM class.
getInstance()	Gets a CIM instance.
associators()	Enumerates associators of a CIM object.
associatorName()	Enumerates names of associators of a CIM object.
references()	Enumerates references to a CIM object.
referenceName()	Enumerates names of references to a CIM object.

The target of these methods is the location of Cisco DCNM SMI-S Server, which is identified by the switch IP address.

Performance monitoring provides the status and statistics for the local ports. Only ports on the local switch can be monitored. You can retrieve statistics from the properties of the `FCPortStatistics` class for `FCPort` class instances on Cisco DCNM SMI-S Server.

Note The namespace of Cisco DCNM SMI-S Server is cimv2.

Modeling a Module Using the Blade Subprofile

You can use the Blade subprofile to model a supervisor module, switching module, or services module within a switch. Table 3-2 shows how to use the association classes in this subprofile to map ports to modules and modules to switches.

Table 3-2 Using the Blade Subprofile Association Classes

Class	How Used
Realizes	Associates the `LogicalModule` class to the `PhysicalPackage` class. Use this class to map modules to the switch.
ModulePort	Associates the `FCPort` class to the `LogicalModule` class. Use this class to map individual ports to modules within the switch.

See the "Blade Subprofile" section on page 2-8 for more information about the Blade subprofile.

Configuring Zoning

The zoning model in the SMI-S uses extrinsic and intrinsic methods to manage zoning within the SAN fabric. Extrinsic methods are methods specific to a particular class. Intrinsic methods are methods inherited from the CIM and present in every applicable class.

To create a zone member (referred to as `ZoneMembershipSettingData`), zone, zone alias, or zone set, use `invokeMethod(`*`operand`*`)`. The operand can be one of the extrinsic methods from the zoning subprofiles as shown in Table 3-3.

Table 3-3 Zoning Extrinsic Methods

Extrinsic Method	How Used
`CreateZoneMembershipSettingData()`	Creates a `ZoneMembershipSettingData` and adds it to the specified `Zone` or `NamedAddressCollection`. The `ConnectivityMemberID` is dependent upon the `ConnectivityMemberType`.
`CreateZone()`	Creates a `Zone` and associates it to `AdminDomain` where the `ZoneService` is hosted.
`CreateZoneAlias()`	Creates a `ZoneAlias` and associates it to `AdminDomain` where the `ZoneService` is hosted.
`CreateZoneSet()`	Creates a `ZoneSet` and associates it to the `AdminDomain` where the `ZoneService` is hosted.
`AddZone()`	Adds the `Zone` to the specified `ZoneSet`. 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`.
`AddZoneMembershipSettingData()`	Adds `ZoneMembershipSettingData` to the `Zone` or `NamedAddessCollection`.
`AddZoneAlias()`	Adds the `Zone Alias to` the `Zone`.
`ActivateZoneSet ()`	Sets the `ZoneSet` to active.
`ZoneSetDistribute()`	Distributes the full `ZoneSet` along with active zone set per VSAN in the fabric.
`CreatDeviceAlias()`	Creates a device alias with the given device alias name and PWWN.

Use the `DeleteInstance(`*`instance_name`*`)` intrinsic method to remove a zoning item from a collection or to delete the zoning item entirely. The `DeleteInstance()` method requires a reference to one of the instances shown in Table 3-4.

Table 3-4 Deleting Zoning Entities

Class	How Used
`CIM_ElementSettingData`	Removes a zone member from a zone or zone alias. Use `deleteInstance()` to delete the instance of `ElementSettingData` that associates the zone member to the zone.
`CIM_MemberOfCollection`	Removes a zone or zone alias from a zone set. Use `deleteInstance()` to delete the instance of `MemberOfCollection` that associates the zone or zone alias to the zone set.
`CIM_ZoneMembershipSettingData`	Deletes a zone member. This automatically removes it from any zone or zone alias.
`CIM_Zone`	Deletes a zone.
`CIM_ZoneAlias`	Deletes a zone alias.
`CIM_ZoneSet`	Deletes a zone set.
`RemoveDeviceAlias()`	Removes the device alias with the given device alias name.

See the "Zone Control Subprofile" section on page 2-27 and the "Enhanced Zoning and Enhanced Zoning Control Subprofile" section on page 2-25 for information about the zoning subprofiles.

Note For more information about SMI-S, refer to the SNIA website at http://www.snia.org. For more information about CIM, refer to the DMTF website at http://www.dmtf.org.