



CHAPTER 3

Configuring and Using the CIM Server

This chapter provides the steps to configure the CIM server in Cisco MDS 9000 Family products and gives some sample scenarios for using CIM objects to manage your SAN. This chapter includes the following sections:

- [Configuring the CIM Server, page 3-1](#)
- [Performing Discovery and Performance Monitoring, page 3-3](#)
- [Modeling a Module Using the Blade Subprofile, page 3-4](#)
- [Configuring Zoning, page 3-4](#)



Note

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

Configuring the CIM Server

The CIM server can be configured through the CLI. To configure the CIM server, you must first enable it. For added security, you can install an SSL certificate to encrypt the login information and enable HTTPS before enabling the CIM server. The CIM server requires HTTP or HTTPS or both to be enabled. By default, HTTP is enabled and secure HTTPS is disabled. Using HTTPS encrypts all management traffic between the CIM client and the CIM server and is the recommended configuration.

Creating a Certificate Using OpenSSL

You need a valid certificate to configure the CIM server. You can use OpenSSL to create the private key and certificate needed by the CIM server. Refer to <http://www.openssl.org>.

To create a self-signed certificate and private key using OpenSSL, follow these steps:

- Step 1** Create a file called `ssl.conf` on your workstation. This is used to specify the distinguished name. Sample contents of this file are:

```
[ req ]
distinguished_name = req_distinguished_name
prompt = no
[ req_distinguished_name ]
CN = Common Name
emailAddress = test@email.address
```

Send documentation comments to mdsfeedback-doc@cisco.com

Step 2 Use the **openssl** command to create the private key and the certificate by typing the following:

```
/usr/bin/openssl req -x509 -days 365 -newkey rsa:2048 -nodes -config ./ssl.conf -keyout
./key.pem -out ./cert.pem
```

Step 3 Concatenate the private key and the certificate into a single file.

```
cat key.pem cert.pem > cimserver1.pem
```

Step 4 Copy cimserver1.pem to bootflash: on your switch. You use this as the certificate when configuring the CIM server.

Installing the Certificate and Enabling the CIM Server

To configure a CIM server using the HTTPS protocol, follow these steps:

	Command	Purpose
Step 1	switch# config t	Enters configuration mode.
Step 2	switch(config)# cimserver certificate bootflash:cimserver1.pem	Installs a Secure Socket Layer (SSL) certificate specified in the file named with a .pem extension.
	switch(config)# cimserver clearcertificate Certificate1	Optional. Clears the specified SSL certificate (Certificate1).
Step 3	switch(config)# cimserver enableHttps	Enables HTTPS (secure protocol).
	switch(config)# no cimserver enableHttps	Optional. Disables HTTPS (default).
Step 4	switch(config)# cimserver enable	Enables the CIM server.
	switch(config)# no cimserver enable	Optional. Disables the CIM server (default).

To configure a CIM server using the HTTP protocol, follow these steps:

	Command	Purpose
Step 1	switch# config t	Enters configuration mode.
Step 2	switch(config)# cimserver enable	Enables the CIM server using the default HTTP (non-secure) protocol.
	switch(config)# no cimserver enable	Optional. Disables the CIM server (default).
	switch(config)# no cimserver enableHttp	Optional. Disables HTTP.
	switch(config)# cimserver enableHttp	Optional. Enables HTTP and reverts to the switch default.

Setting CIM Server Loglevel

To configure a CIM server loglevel, follow these steps:

	Command	Purpose
Step 1	switch# show cimserver logs	Displays CIM server logs.

Send documentation comments to mdsfeedback-doc@cisco.com

	Command	Purpose
Step 2	switch# <code>conf t</code>	Configures CIM server loglevels.
Step 3	switch(config)# <code>cimserver logLevel</code>	Sets the CIM server loglevel. The loglevels range from 1-5 (1-trace, 2-information, 3-warning, 4-severe, 5-fatal).

Performing Discovery and Performance Monitoring

You can use the Fabric and Switch profiles to implement discovery and performance monitoring. See the “Fabric Profile” section on page 2-4 and the “” section on page 2-7 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, 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
<code>enumerateInstances()</code>	Enumerate instances of a CIM class.
<code>enumerateInstanceNames()</code>	Enumerate names of instances of a CIM class.
<code>getInstance()</code>	Get a CIM instance.
<code>associators()</code>	Enumerate associators of a CIM object.
<code>associatorName()</code>	Enumerate names of associators of a CIM object.
<code>references()</code>	Enumerate references to a CIM object.
<code>referenceName()</code>	Enumerate names of references to a CIM object.

The target of these methods is the location of the CIM 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 the CIM server.



Note

The namespace of the CIM server is `root/cimv2`.

[Send documentation comments to mdsfeedback-doc@cisco.com](mailto:mdsfeedback-doc@cisco.com)

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 <code>LogicalModule</code> class to the <code>PhysicalPackage</code> class. Use this class to map modules to the switch.
ModulePort	Associates the <code>FCPort</code> class to the <code>LogicalModule</code> class. Use this class to map individual ports to modules within the switch.

See the “Blade Subprofile” section on page 2-3 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
<code>CreateZoneMembershipSettingData()</code>	Creates a <code>ZoneMembershipSettingData</code> and adds it to the specified <code>Zone</code> or <code>NamedAddressCollection</code> . The <code>ConnectivityMemberID</code> is dependent upon the <code>ConnectivityMemberType</code> .
<code>CreateZone()</code>	Creates a <code>Zone</code> and associates it to <code>AdminDomain</code> where the <code>ZoneService</code> is hosted.
<code>CreateZoneAlias()</code>	Creates a <code>ZoneAlias</code> and associates it to <code>AdminDomain</code> where the <code>ZoneService</code> is hosted.
<code>CreateZoneSet()</code>	Creates a <code>ZoneSet</code> and associates it to the <code>AdminDomain</code> where the <code>ZoneService</code> is hosted.
<code>AddZone()</code>	Adds the <code>Zone</code> to the specified <code>ZoneSet</code> . Adding a <code>Zone</code> to a <code>ZoneSet</code> extends the zone enforcement definition of the <code>ZoneSet</code> to include the members of that <code>Zone</code> . If adding the <code>Zone</code> is successful, the <code>Zone</code> should be associated to the <code>ZoneSet</code> by <code>MemberOfCollection</code> .
<code>AddZoneMembershipSettingData()</code>	Adds <code>ZoneMembershipSettingData</code> to the <code>Zone</code> or <code>NamedAddressCollection</code> .

Send documentation comments to mdsfeedback-doc@cisco.com

Table 3-3 Zoning Extrinsic Methods (continued)

Extrinsic Method	How Used
AddZoneAlias()	Adds the ZoneAlias to the Zone.
ActivateZoneSet ()	Sets the ZoneSet to active.

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.

See the “Zone Control Subprofile” section on page 2-5 and the “Enhanced Zoning and Enhanced Zoning Control Subprofile” section on page 2-6 for information about the zoning subprofiles.



Note

These methods are supported for the CIM protocol only and cannot be entered as commands at the CLI. 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>.

Send documentation comments to mdsfeedback-doc@cisco.com