Cisco UCS Rack-Mount Servers Cisco IMC XML API Programmer's Guide for 3X60 Servers

First Published: 2015-09-17

Last Modified: 2016-09-16

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

Preface

Audience

This guide is intended for software engineers with a background in programming and the use of APIs. Engineers should have knowledge of XML, data systems, networking protocols, and storage protocols.

Document Organization

This XML API Reference Guide is organized into the following chapters:

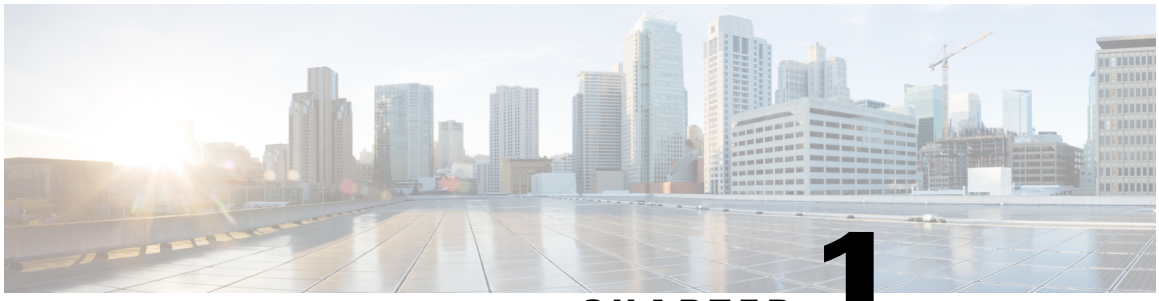
Related Documentation

For a complete list of all C-Series documentation, see the *Cisco UCS C-Series Servers Documentation Roadmap* available at the following URL: http://www.cisco.com/go/unifiedcomputing/c-series-doc.

Note The *Cisco UCS C-Series Servers Integrated Management Controller GUI Configuration Guide* and the *Cisco UCS C-Series Servers Integrated Management Controller CLI Command Reference* provide an overview of Cisco IMC. This is important background information for XML API software developers.

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to ucs-docfeedback@cisco.com. We appreciate your feedback.



CHAPTER 1

Cisco IMC XML API

This chapter includes the following sections:

About the Cisco IMC XML API

The Cisco IMC XML API is a programmatic interface to the Cisco Integrated Management Controller (Cisco IMC) software for a C-Series Rack-Mount Server. The API accepts XML documents through HTTP or HTTPS. Developers can use any programming language to generate XML documents that contain the API methods. Configuration and state information for Cisco IMC is stored in a hierarchical tree structure known as the MIT (Management Information Tree), which is completely accessible through the XML API.

The Cisco IMC XML API implements a subset of the methods and management information model available in the Cisco UCS Manager XML API. The behavior of both APIs is similar in syntax and semantics, and you can use the same client development tools and techniques for both. The scope of the Cisco IMC XML API is limited to a single C-Series Rack-Mount Server, in contrast to the Cisco UCS Manager XML API, which controls an entire Cisco UCS domain consisting of switches, FEX modules, servers, and other devices.

Using the Cisco IMC XML API, the user has programmatic access to Cisco IMC to configure, administer, and monitor the server. The API provides most of the functions that are accessible through the Cisco IMC CLI and GUI interfaces.

Operation of the API is transactional and terminates on a single data model maintained in Cisco IMC.

The API model includes the following programmatic entities:

- Classes—Define the properties and states of objects in the MIT.
- Methods—Actions that the API performs on one or more objects.

- Types—Object properties that map values to the object state (for example, `equipmentPresence`).

A typical request comes into Cisco IMC and is placed in the transactor queue in FIFO order. The transactor gets the request from the queue, interprets the request, and performs an authorization check. After the request is confirmed, the transactor updates the MIT. This complete operation is done in a single transaction.

Event subscription is supported. Up to four Cisco IMC XML API clients can subscribe to receive event notifications from Cisco IMC. The event subscription operation establishes a connection session allowing a client to receive XML-formatted event notification messages that are sent asynchronously by Cisco IMC.

Cisco UCS Management Information Model

All the physical and logical components that comprise Cisco UCS are represented in a hierarchical management information model (MIM), also referred to as the MIT. Each node in the tree represents a managed object (MO) or group of objects that contains its administrative state and its operational state.

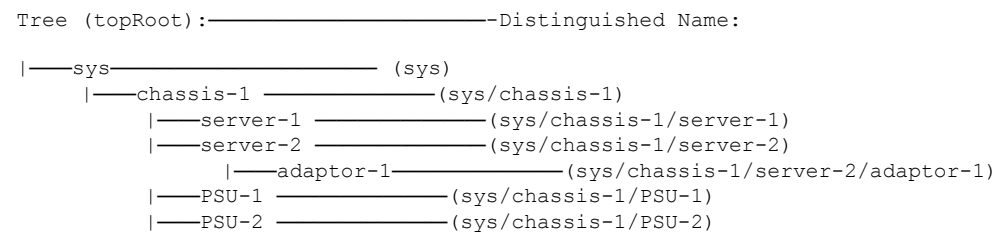
The hierarchical structure starts at the top (`sys`) and contains parent and child nodes. Each node in this tree is a managed object and each object in Cisco UCS has a unique distinguished name (DN) that describes the object and its place in the tree. Managed objects are abstractions of the Cisco UCS resources, such as CPUs, DIMMs, adapter cards, fans, and power supply units..

Configuration policies are the majority of the policies in the system and describe the configurations of different Cisco UCS components. Policies determine how the system behaves under specific circumstances. Certain managed objects are not created by users, but are automatically created by the Cisco UCS, for example, power supply objects and fan objects. By invoking the API, you are reading and writing objects to the MIM.

Cisco IMC Management Information Model

The Cisco IMC management information model is a subset of the Cisco UCS management information model. A C-Series Rack-Mount Server is modeled starting with `sys/chassis-1/server-1` in the MIT as in the following example:

Figure 1: Illustration of the Cisco IMC MIM Structure



Cisco IMC XML API Sample Flow

A typical request comes into Cisco IMC and is placed in the transactor queue in FIFO order. The transactor gets the request from the queue, interprets the request, and performs an authorization check. After the request is confirmed, the transactor updates the management information tree. This operation is done in a single transaction.

The following figure shows how Cisco IMC processes a boot server request. The following table describes the steps involved in a boot server request.

Figure 2: Sample Flow of Boot Server Request

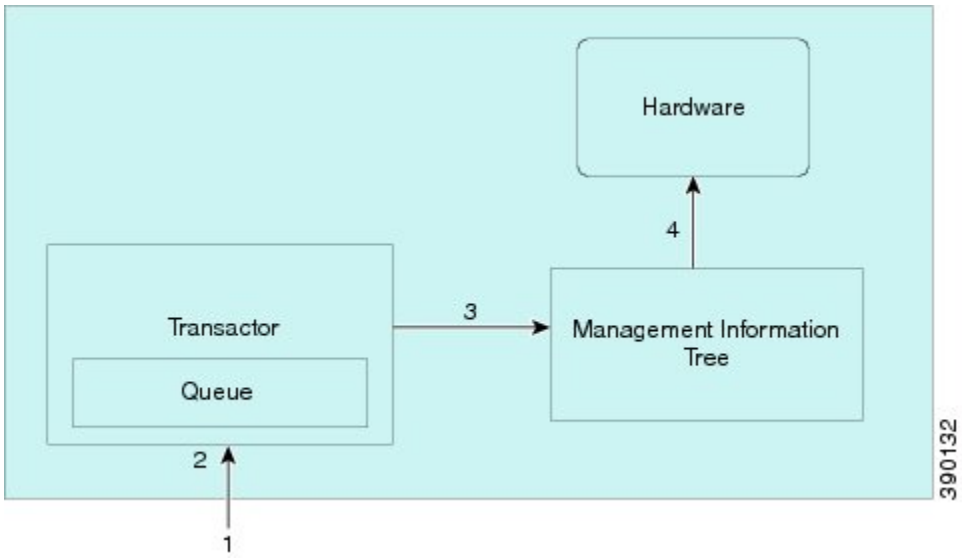


Table 1: Explanation of Boot Server Request

Step	Command/Process	Operational Power State of MO (Server)
1	CMD request: boot server	Down
2	Request queued	Down
3	State change in management information tree and make persistent the managed object (MO) state change	Down
4	Apply boot stimuli	Up

Object Naming

You can identify a specific object by its distinguished name (DN) or by its relative name (RN).

Distinguished Name

The distinguished name enables you to unambiguously identify a target object. The distinguished name has the following format consisting of a series of relative names:

```
dn = {rn}/{rn}/{rn}/{rn}...
```

In the following example, the DN provides a fully qualified path for `adaptor-1` from the top of the object tree to the object. The DN specifies the exact managed object on which the API call is operating.

```
< dn ="sys/chassis-1/server-1/adaptor-1"/>
```

Relative Name

The relative name identifies an object within the context of its parent object. The distinguished name is composed of a sequence of relative names.

For example, this distinguished name:

```
<dn = "sys/chassis-1/server-1/adaptor-1/host-eth-2"/>
```

is composed of the following relative names:

```
topSystem MO: rn="sys"
equipmentChassis MO: rn ="chassis-1"
computeServerNode MO: rn ="server-<id>"
adaptorUnit MO: rn="adaptor-<id>"
adaptorHostEthIf MO: rn="host-eth-<id>"
```

API Method Categories

Each method corresponds to an XML document.

Note Several code examples in this guide substitute the term `<real_cookie>` for an actual cookie (such as 1217377205/85f7ff49-e4ec-42fc-9437-da77a1a2c4bf). The XML API cookie is a 47-character string; it is not the type of cookie that web browsers store locally to maintain session information.

Authentication Methods

Authentication methods authenticate and maintain the session. For example:

- aaaLogin—Initial method for logging in.
- aaaRefresh—Refreshes the current authentication cookie.
- aaaLogout—Exits the current session and deactivates the corresponding authentication cookie.

Use the aaaLogin method to get a valid cookie. Use aaaRefresh to maintain the session and keep the cookie active. Use the aaaLogout method to terminate the session (also invalidates the cookie). A maximum of 4 sessions to the Cisco UCS can be opened at any one time.

Operations are performed using the HTTP post method (Cisco IMC supports both HTTP and HTTPS requests) over TCP. HTTP and HTTPS can be configured to use different port numbers, but TCP/443 (or TCP/80 for non-secure connections) is used by default. The HTTP envelope contains the XML configuration.

Tip In Cisco IMC, HTTP to HTTPS redirection is enabled by default. To capture HTTP packets between the client application and Cisco IMC, disable redirection in the Cisco IMC GUI or CLI.

Query Methods

Query methods obtain information on the current configuration state of an object. The following are query methods supported:

- configResolveDn—Retrieves objects by DN.
- configResolveClass—Retrieves objects of a given class.
- configResolveChildren—Retrieves the child objects of an object.
- configResolveParent—Retrieves the parent object of an object.

Most query methods have the argument inHierarchical (Boolean true/yes or false/no). If true, the inHierarchical argument returns all child objects.

```
<configResolveDn ... inHierarchical="false"></>
<configResolveDn ... inHierarchical="true"></>
```

Because the amount of data returned from Cisco IMC can be quite large, the inHierarchical argument should be used with care. For example, if the query method is used on a class or DN that refers to a managed object (MO) that is located high on the management information tree and inHierarchical is set to true, the response can contain almost the entire Cisco IMC configuration. The resources required for Cisco IMC to process the request can be high, causing Cisco IMC to take an extended amount of time to respond. To avoid delays, the query method should be performed on a smaller scale involving fewer MOs.

Tip If a query method does not respond or is taking a long time to respond, increase the timeout period on the client application or adjust the query method to involve fewer MOs.

The query API methods might also have an inRecursive argument to specify whether the call should be recursive (for example, follow objects that point back to other objects or the parent object).

Note Until a host is powered on at least once, Cisco IMC may not have complete inventory and status information. For example, if Cisco IMC is reset, it will not have detailed CPU, memory, or adapter inventory information until the next time the host is powered on. If a query method is performed on a MO corresponding to the unavailable data, the response will be blank.

Configuration Methods

The Cisco IMC XML API supports only a single method to make configuration changes to managed objects:

- configConfMo—Affects a single managed object (for example, a DN).

Event Subscription Methods

Applications get state change information by regular polling or event subscription. For more efficient use of resources, event subscription is the preferred method of notification. Polling should be used only under very limited circumstances.

Use eventSubscribe to register for events, as shown the following example:

```
<eventSubscribe
    cookie="<real_cookie>">
</eventSubscribe>
```

To receive notifications, open an HTTP or HTTPS session over TCP and keep the session open. On receiving eventSubscribe, starts sending all new events as they occur. You can unsubscribe from these events using the eventUnsubscribe method.

Each event has a unique event ID. Event IDs operate as counters and are included in all method responses. When an event is generated, the event ID counter increments and is assigned as the new event ID. This sequential numbering enables tracking of events and ensures that no event is missed.

An event channel connection opened by a user will be closed automatically by after 600 seconds of inactivity associated with the event channel session cookie. To prevent automatic closing of the event channel connection by , the user must either send the aaaKeepAlive request for the same event channel session cookie within 600 seconds or send any other XML API method to using the same event channel session cookie.

Success or Failure Response

When responds to an XML API request, the response indicates failure if the request is impossible to complete. A successful response indicates only that the request is valid, not that the operation is complete. For example, it may take some time for a server to finish a power-on request. The power state changes from down to up only after the server actually powers on.

Successful Response

When a request has executed successfully, Cisco IMC returns an XML document with the information requested or a confirmation that the changes were made. The following is an example of a configResolveDn query on the distinguished name `sys/chassis-1/server-1/adaptor-2/ext-eth-0`:

```
<configResolveDn
     dn="sys/chassis-1/server-1/adaptor-2/ext-eth-0"
     cookie="<real_cookie>"
     inHierarchical="false"/>
```

The response includes the following information:

```
<configResolveDn
    cookie="<real_cookie>"
    response="yes"
    dn="sys/chassis-1/server-1/adaptor-2/ext-eth-0">
    <outConfig>
        <adaptorExtEthIf
            id="0"
            ifType="physical"
            linkState="up"
            mac="00:22:BD:D6:42:DA"
```

```
            name=""
            operState="up"
            portId="0"
            purpose="general"
            transport="CE"
            type=""
            dn="sys/chassis-1/server-1/adaptor-2/ext-eth-0" >
        </adaptorExtEthIf>
    </outConfig>
</configResolveDn>
```

Failed Requests

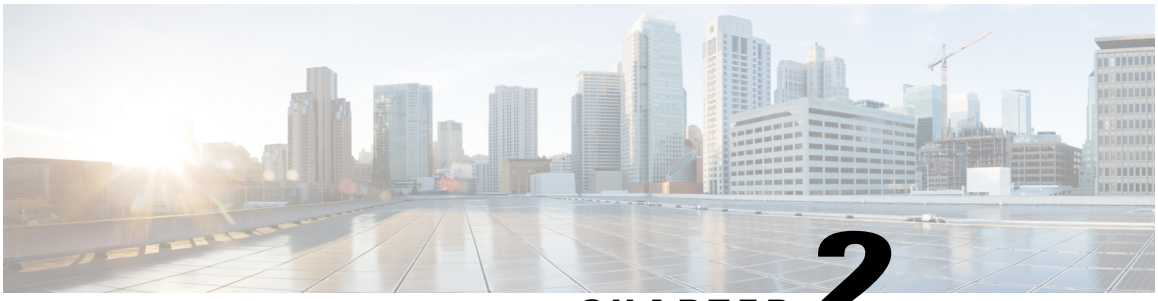
The response to a failed request includes XML attributes for errorCode and errorDescr. The following is an example of a response to a failed request:

```
<configConfMo dn="sys/chassis-1/server-1/adaptor-1/ext-eth-0"
     cookie="<real_cookie>"
     response="yes"
     errorCode="103"
     invocationResult="unidentified-fail"
     errorDescr="can't create; object already exists.">
</configConfMo>
```

Empty Results

A query request for a nonexistent object is not treated as a failure by Cisco IMC. If the object does not exist, Cisco IMC returns a success message, but the XML document contains an empty data field (`<outConfig> </outConfig>`) to indicate that the requested object was not found. The following example shows the response to an attempt to resolve the distinguished name on a nonexistent rack-mount server:

```
<configResolveDn
    cookie="<real_cookie>"
    response="yes"
    dn="sys/chassis-1/server-1/adaptor-9999">
    <outConfig>
    </outConfig>
</configResolveDn>
```



CHAPTER 2

Using the Cisco IMC XML API Methods

This chapter includes the following sections:

Authentication Methods

Authentication allows XML API interaction with the Cisco IMC. It provides a way to set permissions and control the operations that can be performed.

Note Most code examples in this guide substitute the term `<real_cookie>` for an actual cookie (such as 1217377205/85f7ff49-e4ec-42fc-9437-da77a1a2c4bf). The Cisco UCS cookie is a 47-character string; it is not the type of cookie that web browsers store locally to maintain session information.

Login

To log in, the XML API client establishes a TCP connection to the Cisco IMC HTTP (or HTTPS) server and posts an XML document containing the aaaLogin method.

In the following example, the Telnet utility is used to establish a TCP connection to port 80 of the Cisco IMC with IP address 192.0.20.72. The path used is /nuova.

```
$ telnet 192.0.20.72 80
POST /nuova HTTP/1.1
USER-Agent: lwp-request/2.06
HOST: 192.0.20.72
Content-Length: 62
Content-Type: application/x-www-form-urlencoded
```

Next, the client specifies the aaaLogin method and provides a user name and password:

```
<aaaLogin
    inName='admin'
```

```
    inPassword='password'>
</aaaLogin>
```

Note Do not include XML version or DOCTYPE lines in the XML API document. The inName and inPassword attributes are parameters.

Each XML API document represents an operation to be performed. When the request is received as an XML API document, Cisco IMC reads the request and performs the actions as provided in the method. Cisco IMC responds with a message in XML document format and indicates success or failure of the request.

The following is a typical successful response:

```
1 <aaaLogin
2    response="yes"
3    outCookie="<real_cookie>"
4       outRefreshPeriod="600"
5       outPriv="admin">
6 </aaaLogin>
```

Each line in the response should be interpreted as follows:

1 Specifies the method used to login.

2 Confirms that this is a response.

3 Provides the session cookie.

4 Specifies the recommended cookie refresh period. The default login session length is 600 seconds.

5 Specifies the privilege level assigned to the user account (this can be admin, user, or readonly).

6 Closing tag.

Alternatively, you can use the cURL utility to log in to the XML API, as shown in the following example:

```
curl -d "<aaaLogin inName='admin' inPassword='password'></aaaLogin>" http://192.0.20.72/nuova
```

If HTTPS is enabled, you must use HTTPS in the cURL command, as shown in the following example:

```
curl -d "<aaaLogin inName='admin' inPassword='password'></aaaLogin>" https://192.0.20.72/nuova
```

Refreshing the Session

Sessions are refreshed with the aaaRefresh method, using the 47-character cookie obtained either from the aaaLogin response or a previous refresh.

```
<aaaRefresh
    cookie="<real_cookie>"
    inCookie="<real_cookie>"
    inName='admin'
    inPassword='password'>
</aaaRefresh>
```

Logging Out of the Session

Use the following method to log out of a session:

```
<aaaLogout
    cookie="<real_cookie>"
    inCookie="<real_cookie>"
</aaaLogout>
```

Unsuccessful Responses

Failed login:

```
<aaaLogin
    cookie=""
    response="yes"
    errorCode="551"
    invocationResult="unidentified-fail"
    errorDescr="Authentication failed">
</aaaLogin>
```

Nonexistent object (blank return indicates no object with the specified DN):

```
<configResolveDn
    cookie="<real_cookie>"
    response="yes"
    dn="sys/chassis-1/server-1/adaptor-9999">
    <outConfig>
    </outConfig>
</configResolveDn>
```

Bad request:

```
<configConfMo
    cookie="<real_cookie>"
    response="yes"
    dn="sys/chassis-1/server-1/adaptor-1/ext-eth-0">
    errorCode="103“
    invocationResult="unidentified-fail“
    errorDescr="can't create; object already exists.">
</configConfMo>
```

Query Methods

Using configResolveChildren

When resolving children of objects in the MIT, note the following:

- This method obtains all child objects of a named object that are instances of the named class. If a class name is omitted, all child objects of the named object are returned.
- inDn attribute specifies the named object from which the child objects are retrieved (required).
- classId attribute specifies the name of the child object class to return (optional).

- Authentication cookie (from aaaLogin or aaaRefresh) is required.
- inHierarchical attribute (default = false) if true, specifies that results are hierarchical.
- Enumerated values, `classIds`, and bit masks are displayed as strings.

See the example request/response in configResolveChildren, on page 22.

Using configResolveClass

When resolving a class, note the following:

- All objects of the specified class type are retrieved.
- classId specifies the object class name to return (required).
- Authentication cookie (from aaaLogin or aaaRefresh) is required.
- inHierarchical attribute (default = false) if true, specifies that results are hierarchical.
- Enumerated values, `classIds`, and bit masks are displayed as strings.

Result sets can be large. Be precise when defining result sets. For example, to obtain only a list of adapters, use adaptorUnit as the attribute value for classId in the query. This example queries for all instances of the `adaptorUnit` class:

```
<configResolveClass
 cookie="real_cookie"
 inHierarchical="false"
 classId="adaptorUnit"/>
```

See the example request/response in configResolveClass, on page 23.

Using configResolveDn

When resolving a DN, note the following:

- The object specified by the DN is retrieved.
- Specified DN identifies the object instance to be resolved (required).
- Authentication cookie (from aaaLogin or aaaRefresh) is required.
- inHierarchical attribute (default = false) if true, specifies that results are hierarchical.
- Enumerated values, `classIds`, and bit masks are displayed as strings.

See the example request/response in configResolveDn, on page 24.

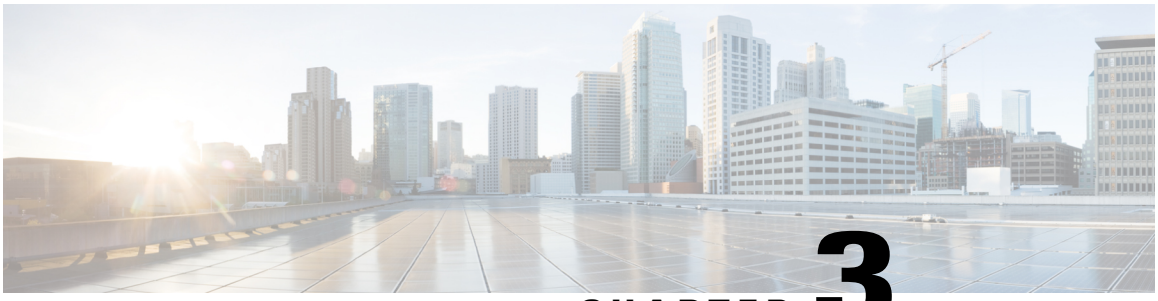
Using configResolveParent

When resolving the parent object of an object, note the following:

- This method retrieves the parent object of a specified DN.

- dn attribute is the DN of the child object (required).
- Authentication cookie (from aaaLogin or aaaRefresh) is required.
- inHierarchical attribute (default = false) if true, specifies that results are hierarchical.
- Enumerated values, `classIds`, and bit masks are displayed as strings.

See the example request/response in configResolveParent, on page 25.



CHAPTER 3

Cisco IMC XML API Method Descriptions

This chapter includes the following sections:

aaaGetComputeAuthTokens

The aaaGetComputeAuthTokens method returns authentication tokens that are used to launch the KVM. This generates two temporary authentication tokens that are valid for 60 seconds. The first is the KVM user name and the second token is the password. Using the authorization tokens as credentials, you can access the URL from where you can download the Java Network Launch Protocol (JNLP) file. You can download the JNLP file from the URL and launch it to start a KVM session.

Note

- You cannot obtain tokens if the vKVM option is disabled on the Cisco IMC.
- You must have user or admin privileges to the Cisco IMC to obtain the authentication tokens. Users with read-only privileges will not be able to obtain the tokens.
- The authorization tokens expire is 60 seconds; you cannot use the tokens after 60 seconds to access the URL. If you try to access after 60 seconds, the login fails and you get a authentication failure or timeout message.

Request Syntax

```
<xs:element name="aaaGetComputeAuthTokens" type="aaaGetComputeAuthTokens"
substitutionGroup="externalMethod"/>
     <xs:complexType name="aaaGetComputeAuthTokens" mixed="true">
         <xs:attribute name="cookie" type="stringMin0Max47" use="required"/>
         <xs:attribute name="response" type="YesOrNo"/>
     </xs:complexType>
```

Response Syntax

```
<xs:element name="aaaGetComputeAuthTokens" type="aaaGetComputeAuthTokens"
substitutionGroup="externalMethod"/>
        <xs:complexType name="aaaGetComputeAuthTokens" mixed="true">
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="outTokens">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                         <xs:minLength value="0"/>
                         <xs:maxLength value="510"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
        </xs:complexType>
```

Examples

Request:

```
aaaGetComputeAuthTokens
cookie="<real_cookie>" />
```

Response:

```
<aaaGetComputeAuthTokens cookie="<real_cookie>" outTokens="1804289383,846930886"
response="yes"> </aaaGetComputeAuthTokens>
```

aaaKeepAlive

The aaaKeepAlive method keeps the session active until the default session time expires, using the same cookie after the method call.

Request Syntax

```
<xs:element name="aaaKeepAlive" type="aaaKeepAlive" substitutionGroup="externalMethod"/>
        <xs:complexType name="aaaKeepAlive" mixed="true">
            <xs:attribute name="cookie" type="stringMin0Max47" use="required"/>
            <xs:attribute name="response" type="YesOrNo"/>
        </xs:complexType>
```

Response Syntax

```
<xs:element name="aaaKeepAlive" type="aaaKeepAlive" substitutionGroup="externalMethod"/>
        <xs:complexType name="aaaKeepAlive" mixed="true">
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
        </xs:complexType>
```

Examples

Request

```
<aaaKeepAlive
cookie="<real_cookie>">
</aaaKeepAlive>
```

Response

```
<aaaKeepAlive cookie="<real_cookie>" response="yes"> </aaaKeepAlive>
```

aaaLogin

The aaaLogin method is the login process and is required to begin a session. This action establishes the HTTP (or HTTPS) session between the client and Cisco IMC.

Request Syntax

```
<xs:element name="aaaLogin" type="aaaLogin" substitutionGroup="externalMethod"/>
        <xs:complexType name="aaaLogin" mixed="true">
            <xs:attribute name="inName" use="required">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                         <xs:pattern value="[\-\.:_a-zA-Z0-9]{0,16}"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="inPassword" use="required">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                         <xs:minLength value="0"/>
                         <xs:maxLength value="510"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="cookie" type="stringMin0Max47"/>
            <xs:attribute name="response" type="YesOrNo"/>
        </xs:complexType>
```

Response Syntax

```
<xs:element name="aaaLogin" type="aaaLogin" substitutionGroup="externalMethod"/>
        <xs:complexType name="aaaLogin" mixed="true">
            <xs:attribute name="outCookie" type="xs:string"/>
            <xs:attribute name="outRefreshPeriod" type="xs:unsignedInt"/>
        <xs:attribute name="outPriv">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:pattern value="(read-only|admin|user){0,1}"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
            <xs:attribute name="outDomains" type="xs:string"/>
            <xs:attribute name="outChannel">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                         <xs:enumeration value="fullssl"/>
                         <xs:enumeration value="noencssl"/>
                         <xs:enumeration value="plain"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="outEvtChannel">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                         <xs:enumeration value="fullssl"/>
                         <xs:enumeration value="noencssl"/>
                         <xs:enumeration value="plain"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="outSessionId">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                         <xs:minLength value="0"/>
                         <xs:maxLength value="32"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="outVersion" type="xs:string"/>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
        </xs:complexType>
```

Examples

Request

```
<aaaLogin inName='admin' inPassword='password'/>
```

Response

```
<aaaLogin cookie="" response="yes" outCookie="<real_cookie>" outRefreshPeriod="600"
outPriv="admin" outSessionId="17" outVersion="3.0(0.149)"> </aaaLogin>
```

aaaLogout

The aaaLogout method is a process to close a web session by passing the session cookie as input. It is not automatic; the user has to explicitly invoke the aaaLogout method to terminate the session.

Request Syntax

```
<xs:element name="aaaLogout" type="aaaLogout" substitutionGroup="externalMethod"/>
        <xs:complexType name="aaaLogout" mixed="true">
            <xs:attribute name="inCookie" type="stringMin0Max47" use="required"/>
            <xs:attribute name="cookie" type="stringMin0Max47"/>
            <xs:attribute name="response" type="YesOrNo"/>
        </xs:complexType>
```

Response Syntax

```
<xs:element name="aaaLogout" type="aaaLogout" substitutionGroup="externalMethod"/>
        <xs:complexType name="aaaLogout" mixed="true">
            <xs:attribute name="outStatus">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                         <xs:enumeration value="success"/>
                         <xs:enumeration value="failure"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
        </xs:complexType>
```

Examples

Request

```
<aaaLogout cookie="<real_cookie>" inCookie="<real_cookie>"></aaaLogout>
```

Response

```
<aaaLogout cookie="<real_cookie>" response="yes" outStatus="success"> </aaaLogout>
```

aaaRefresh

The aaaRefresh method keeps sessions active (within the default session time frame) by user activity. There is a default of 600 seconds that counts down when inactivity begins. If the 600 seconds expire, Cisco IMC enters a sleep mode. It requires signing back in, which restarts the countdown. It continues using the same session ID.

Note Using this method expires the previous cookie and issues a new cookie.

Request Syntax

```
<xs:element name="aaaRefresh" type="aaaRefresh" substitutionGroup="externalMethod"/>
        <xs:complexType name="aaaRefresh" mixed="true">
            <xs:attribute name="inName" use="required">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
```

```
                        <xs:pattern value="[\-\.:_a-zA-Z0-9]{0,16}"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="inPassword" use="required">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:minLength value="0"/>
                        <xs:maxLength value="510"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="inCookie" type="stringMin0Max47" use="required"/>
            <xs:attribute name="cookie" type="stringMin0Max47"/>
            <xs:attribute name="response" type="YesOrNo"/>
        </xs:complexType>
```

Response Syntax

```
<xs:element name="aaaRefresh" type="aaaRefresh" substitutionGroup="externalMethod"/>
        <xs:complexType name="aaaRefresh" mixed="true">
            <xs:attribute name="outCookie" type="xs:string"/>
            <xs:attribute name="outRefreshPeriod" type="xs:unsignedInt"/>
            <xs:attribute name="outPriv">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:pattern value="(read-only|admin|user){0,1}"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="outDomains" type="xs:string"/>
            <xs:attribute name="outChannel">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="fullssl"/>
                        <xs:enumeration value="noencssl"/>
                        <xs:enumeration value="plain"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="outEvtChannel">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="fullssl"/>
                        <xs:enumeration value="noencssl"/>
                        <xs:enumeration value="plain"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
        </xs:complexType>
```

Examples

Request

```
<aaaRefresh
    cookie="<real_cookie>"
    inCookie="<real_cookie>"
    inName='admin'
    inPassword='password'>
</aaaRefresh>

<aaaRefresh cookie="<real_cookie>" inCookie="<real_cookie>" inName="admin"
```

```
inPassword="password">
</aaaRefresh>
```

Response

```
<aaaRefresh
    cookie="<real_cookie>"
    response="yes"
    outCookie="<real_cookie>"
    outRefreshPeriod="600"
    outPriv="admin">
</aaaRefresh>
```

configConfMo

The configConfMo method configures the specified managed object in a single subtree (for example, DN).

Request Syntax

```
<xs:element name="configConfMo" type="configConfMo" substitutionGroup="externalMethod"/>
        <xs:complexType name="configConfMo" mixed="true">
            <xs:all>
                <xs:element name="inConfig" type="configConfig" minOccurs="1"/>
            </xs:all>
            <xs:attribute name="inHierarchical">
                <xs:simpleType>
                    <xs:union memberTypes="xs:boolean YesOrNo"/>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="cookie" type="stringMin0Max47" use="required"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="dn" type="referenceObject" use="required"/>
        </xs:complexType>
```

Response Syntax

```
<xs:element name="configConfMo" type="configConfMo" substitutionGroup="externalMethod"/>
        <xs:complexType name="configConfMo" mixed="true">
            <xs:all>
                <xs:element name="outConfig" type="configConfig" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
            <xs:attribute name="dn" type="referenceObject"/>
        </xs:complexType>
```

Examples

Request

```
<configConfMo
 cookie="<real_cookie>"
 dn='sys/rack-unit-1/locator-led'>
 <inConfig>
 <equipmentLocatorLed
 adminState='on' dn='sys/rack-unit-1/locator-led'>
 </equipmentLocatorLed>
```

```
 </inConfig>
 </configConfMo>
```

Response

```
<configConfMo dn="sys/rack-unit-1/locator-led"
cookie="1461754266/2f609b81-3176-1176-8007-4cc92474a254" response="yes">
<outConfig>
<equipmentLocatorLed dn="sys/rack-unit-1/locator-led" adminState="inactive" color="unknown"
 id="1" name="" operState="off" status="modified"/></outConfig>
</configConfMo>
```

configResolveChildren

The configResolveChildren method retrieves children of managed objects under a specific DN in the managed information tree.

Request Syntax

```
<xs:element name="configResolveChildren" type="configResolveChildren"
substitutionGroup="externalMethod"/>
        <xs:complexType name="configResolveChildren" mixed="true">
            <xs:attribute name="inDn" type="referenceObject" use="required"/>
            <xs:attribute name="inHierarchical">
                <xs:simpleType>
                    <xs:union memberTypes="xs:boolean YesOrNo"/>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="cookie" type="stringMin0Max47" use="required"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="classId" type="namingClassId"/>
        </xs:complexType>
```

Response Syntax

```
<xs:element name="configResolveChildren" type="configResolveChildren"
substitutionGroup="externalMethod"/>
        <xs:complexType name="configResolveChildren" mixed="true">
            <xs:all>
                <xs:element name="outConfigs" type="configSet" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
            <xs:attribute name="classId" type="namingClassId"/>
        </xs:complexType>
```

Examples

Request

```
<configResolveChildren cookie="<real_cookie>" inHierarchical="false" inDn="sys"/>
```

Response

```
<configResolveChildren cookie="1461754266/2f609b81-3176-1176-8007-4cc92474a254"
response="yes"> <outConfigs> <computeRackUnit dn="sys/rack-unit-1" adminPower="policy"
availableMemory="16384" model="UCSC-Cxxx-M3S" memorySpeed="1333" name="UCS Cxxx M3S"
numOfAdaptors="1" numOfCores="8" numOfCoresEnabled="8" numOfCpus="2" numOfEthHostIfs="2"
```

```
numOfFcHostIfs="2" numOfThreads="8" operPower="on"
originalUuid="3FDC58B1-26CF-4CFA-BFA9-B028047280B1" presence="equipped" serverId="1"
serial="FCH1917V0P1" totalMemory="16384" usrLbl="" uuid="3FDC58B1-26CF-4CFA-BFA9-B028047280B1"
 vendor="Cisco Systems Inc" cimcResetReason="graceful-reboot " assetTag="in demo"
></computeRackUnit><aaaUserEp dn="sys/user-ext" ></aaaUserEp><aaaLdap dn="sys/ldap-ext"
adminState="disabled" basedn="" domain="" filter="sAMAccountName" attribute="CiscoAvPair"
timeout="60" encryption="enabled" locateDirectoryUsingDNS="no"
dnsDomainSource="extracted-domain" dnsSearchDomain="" dnsSearchForest="" ldapServer1=""
ldapServerPort1="389" ldapServer2="" ldapServerPort2="389" ldapServer3=""
ldapServerPort3="389" ldapServer4="" ldapServerPort4="3268" ldapServer5=""
ldapServerPort5="3268" ldapServer6="" ldapServerPort6="3268" bindMethod="login-credentials"
 bindDn="" password="" groupAuth="disabled" groupAttribute="memberOf" groupNestedSearch="128"
 ></aaaLdap><commSvcEp dn="sys/svc-ext" ></commSvcEp><certificateManagement dn="sys/cert-mgmt"
 description="Certificate Management" ></certificateManagement><mgmtImporter
dn="sys/import-config" adminState="disabled" fsmStageDescr="" fsmRmtInvErrCode=""
fsmRmtInvErrDescr="" fsmDescr="import-config" proto="none" hostname="" remoteFile="" user=""
 pwd="" passphrase="" ></mgmtImporter><mgmtBackup dn="sys/export-config" adminState="disabled"
 fsmStageDescr="" fsmRmtInvErrCode="" fsmRmtInvErrDescr="" fsmDescr="export-config"
proto="none" hostname="" remoteFile="" user="" pwd="" passphrase=""
></mgmtBackup><mgmtInventory dn="sys/inventory" adminState="triggered" proto="none"
hostname="" remoteFile="" user="" pwd="" fsmStatus="COMPLETED" progress="100%"
></mgmtInventory><huuController dn="sys/huu" description="Host Upgrade Utility (HUU)"
></huuController><iodController dn="sys/iod" description="Non-Interactive Offline Diagnostics
 (IOD)" ></iodController></outConfigs> </configResolveChildren>
```

configResolveClass

The configResolveClass method returns requested managed object in a given class. If inHierarchical=true, the results contain children.

Request Syntax

```
<xs:element name="configResolveClass" type="configResolveClass"
substitutionGroup="externalMethod"/>
        <xs:complexType name="configResolveClass" mixed="true">
            <xs:attribute name="inHierarchical">
                <xs:simpleType>
                    <xs:union memberTypes="xs:boolean YesOrNo"/>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="cookie" type="stringMin0Max47" use="required"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="classId" type="namingClassId" use="required"/>
        </xs:complexType>
```

Response Syntax

```
<xs:element name="configResolveClass" type="configResolveClass"
substitutionGroup="externalMethod"/>
        <xs:complexType name="configResolveClass" mixed="true">
            <xs:all>
                <xs:element name="outConfigs" type="configSet" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
            <xs:attribute name="classId" type="namingClassId"/>
        </xs:complexType>
```

Examples

Request

```
<configResolveClass cookie="<real_cookie>" inHierarchical="false" classId="topSystem"/>
```

Response

```
<configResolveClass cookie="<real_cookie>" response="yes" classId="topSystem">
 <outConfigs> <topSystem dn="sys" address="10.10.10.10" currentTime="Wed Apr 27 10:51:08
2016 "
  localTime="Wed Apr 27 13:51:08 2016 EAT +0300" timeZone="Africa/Addis Ababa"
mode="stand-alone"
  name="Cxxx-FCH1917V0P1" >
 </topSystem>
</outConfigs>
 </configResolveClass>
```

configResolveDn

The configResolveDn method retrieves a single managed object for a specified DN.

Request Syntax

```
<xs:element name="configResolveDn" type="configResolveDn" substitutionGroup="externalMethod"/>

        <xs:complexType name="configResolveDn" mixed="true">
            <xs:attribute name="inHierarchical">
                <xs:simpleType>
                    <xs:union memberTypes="xs:boolean YesOrNo"/>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="cookie" type="stringMin0Max47" use="required"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="dn" type="referenceObject" use="required"/>
        </xs:complexType>
```

Response Syntax

```
<xs:element name="configResolveDn" type="configResolveDn"
substitutionGroup="externalMethod"/>
        <xs:complexType name="configResolveDn" mixed="true">
            <xs:all>
                <xs:element name="outConfig" type="configConfig" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
            <xs:attribute name="dn" type="referenceObject"/>
        </xs:complexType>
```

Examples

Request

```
<configResolveDn cookie="1461754266/2f609b81-3176-1176-8007-4cc92474a254"
inHierarchical="false" dn="sys/rack-unit-1"/>
```

Response

```
<configResolveDn cookie="<real_cookie>" response="yes" dn="sys/rack-unit-1">
<outConfig> <computeRackUnit dn="sys/rack-unit-1" adminPower="policy" availableMemory="16384"

 model="UCSC-Cxxx-M3S" memorySpeed="1333" name="UCS Cxxx M3S" numOfAdaptors="1" numOfCores="8"

 numOfCoresEnabled="8" numOfCpus="2" numOfEthHostIfs="2" numOfFcHostIfs="2" numOfThreads="8"

 operPower="on" originalUuid="3FDC58B1-26CF-4CFA-BFA9-B028047280B1" presence="equipped"
 serverId="1" serial="FCH1917V0P1" totalMemory="16384" usrLbl=""
 uuid="3FDC58B1-26CF-4CFA-BFA9-B028047280B1" vendor="Cisco Systems Inc"
 cimcResetReason="graceful-reboot " assetTag="in demo" >
 </computeRackUnit>
</outConfig>
</configResolveDn>
```

configResolveParent

For a specified DN, the configResolveParent method retrieves the parent of the managed object.

Request Syntax

```
<xs:element name="configResolveParent" type="configResolveParent"
substitutionGroup="externalMethod"/>
        <xs:complexType name="configResolveParent" mixed="true">
            <xs:attribute name="inHierarchical">
                <xs:simpleType>
                    <xs:union memberTypes="xs:boolean YesOrNo"/>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="cookie" type="stringMin0Max47" use="required"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="dn" type="referenceObject" use="required"/>
        </xs:complexType>
```

Response Syntax

```
<xs:element name="configResolveParent" type="configResolveParent"
substitutionGroup="externalMethod"/>
        <xs:complexType name="configResolveParent" mixed="true">
            <xs:all>
                <xs:element name="outConfig" type="configConfig" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
            <xs:attribute name="dn" type="referenceObject"/>
        </xs:complexType>
```

Examples

Request

```
<configResolveParent cookie="<real_cookie>" inHierarchical="false" dn="sys/rack-unit-1"/>
```

Response

```
<configResolveParent cookie="<real_cookie>" response="yes" dn="sys/rack-unit-1">
 <outConfig>
```

```
<topSystem dn="sys" address="10.197.125.42" currentTime="Wed Apr 27 10:53:26 2016 "
  localTime="Wed Apr 27 13:53:26 2016 EAT +0300" timeZone="Africa/Addis Ababa"
mode="stand-alone"
  name="Cxxx-FCH1917V0P1" >
 </topSystem>
</outConfig>
</configResolveParent>
```

eventSubscribe

The eventSubscribe method allows a client to subscribe to asynchronous System Event Log (SEL) events generated by Cisco IMC.

Event subscription allows a client application to register for event notification from Cisco IMC. When an event occurs, Cisco IMC informs the client application of the event and its type. Only the actual change information is sent. The object's unaffected attributes are not included.

Use eventSubscribe to register for events as shown in the following example:

```
<eventSubscribe
    cookie="<real_cookie>">
</eventSubscribe>
```

Request Syntax

```
<xs:element name="eventSubscribe" type="eventSubscribe" substitutionGroup="externalMethod"/>

        <xs:complexType name="eventSubscribe" mixed="true">
            <xs:attribute name="cookie" type="stringMin0Max47" use="required"/>
            <xs:attribute name="response" type="YesOrNo"/>
        </xs:complexType>
```

Response Syntax

```
<xs:element name="eventSubscribe" type="eventSubscribe"
substitutionGroup="externalMethod"/>
        <xs:complexType name="eventSubscribe" mixed="true">
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
        </xs:complexType>
```

Examples

Request

```
<eventSubscribe
    cookie="<real_cookie>">
</eventSubscribe>
```

Response

```
NO RESPONSE OR ACKNOWLEDGMENT.
```

eventUnsubscribe

The eventUnsubscribe method allows a client to unsubscribe from asynchronous System Event Log (SEL) events generated by Cisco IMC, reversing event subscriptions that resulted from eventUnsubscribe.

Use eventUnsubscribe to unsubscribe from events as shown in the following example:

```
<eventUnsubscribe
    cookie="<real_cookie>">
</eventUnsubscribe>
```

Request Syntax

```
<xs:element name="eventUnsubscribe" type="eventUnsubscribe"
substitutionGroup="externalMethod"/>
        <xs:complexType name="eventUnsubscribe" mixed="true">
            <xs:attribute name="cookie" type="stringMin0Max47" use="required"/>
            <xs:attribute name="response" type="YesOrNo"/>
        </xs:complexType>
```

Response Syntax

```
<xs:element name="eventUnsubscribe" type="eventUnsubscribe"
substitutionGroup="externalMethod"/>
        <xs:complexType name="eventUnsubscribe" mixed="true">
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
        </xs:complexType>
```

Examples

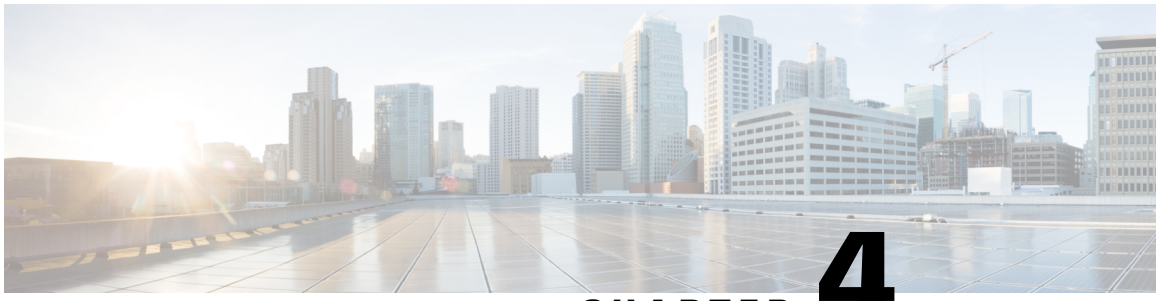
Request

```
<eventUnsubscribe
    cookie="<real_cookie>">
</eventUnsubscribe>
```

Response

```
NO RESPONSE OR ACKNOWLEDGMENT.
```

eventUnsubscribe



CHAPTER 4

Cisco IMC XML Schema Definition Files

This chapter includes the following sections:

About the Cisco IMC XML Schema Definition Files

The C-Series XML API provides users with input XML Schema Definition (xsd) files for every model and a schema definition file for the output:

- RACK-IN.xsd — This document defines the XML document structure for a valid XML request that the Cisco IMC XML API accepts. It also specifies the classes and attributes that you can provide to the XML API configConfMo (Set) requests.
- RACK-OUT.xsd — This document defines the XML document structure for a valid XML response that the Cisco IMC XML API displays. It also specifies the classes and attributes that must appear in the XML API responses.

You can obtain these files from the Cisco IMC at:

RACK-IN.xsd — **https://<Cisco IMC-IP>/visore/RACK-IN.xsd**

RACK-OUT — **https://<Cisco IMC-IP>/visore/RACK-OUT.xsd**

You also can download these files from the Cisco Developer Network, at: http://developer.cisco.com/web/unifiedcomputing/c-series-cimc-xml-api.

You can use one of the available open source tools to validate the XML document against the XML schema files. In the examples used in this section, xmllint available for download at: www.xmlsoft.org is used as the tool for validation. You also can use the xml schema validation feature of a programming language, for example xerces in Java, for this validation.

Examples of RACK-IN.xsd Usage

Validating an XML Request Using RACK-IN.xsd

Example of Use of the RACK-IN.xsd for an Invalid configResolveClass Request

Request:

```
$cat myXMLRequest.xml
<configResolveClass cookie="1360626069/7189c2b0-d57b-157b-8002-f4759de53d50"
inHierarchical="false"/>
```

Validating the request:

```
/usr/bin/xmllint –schema  ./RACK-IN.xsd  myXMLRequest.xml
<configResolveClass cookie="1360626069/7189c2b0-d57b-157b-8002-f4759de53d50"
inHierarchical="false"/>
```

Response:

```
myXMLRequest.xml:1: element configResolveClass:
Schemas validity error : Element 'configResolveClass': The attribute 'classId' is required
 but missing.
```

In the preceding example, validation of the XML request fails and displays an error because the 'classId' is missing in the request.

Example of Use of the RACK-IN.xsd for a Valid configResolveClass Request

Request:

```
$cat myXMLRequest.xml
<configResolveClass cookie="1360626069/7189c2b0-d57b-157b-8002-f4759de53d50"
inHierarchical="false" classId="topSystem"/>
```

Request:

```
/usr/bin/xmllint -schema ./RACK-IN.xsd  myXMLRequest.xml
<configResolveClass cookie="1360626069/7189c2b0-d57b-157b-8002-f4759de53d50"
inHierarchical="false" classId="topSystem"/>
```

In the preceding example, validation of the XML request for a classResolveClass is successful and the response is displayed.

Example of Use of the RACK-IN.xsd for an Invalid configConfMo Request

Request:

```
$cat setRackUnit.xml
<configConfMo cookie="1300242644/ad04d239-d1aa-498d-b074-ccb923066003"
dn="sys/chassis-1/server-1" inHierarchical="false">
  <inConfig>
     <computeServerNode adminPower="down" usrLbl="UCS Cxxx server For Demo"
      availableMemory="16384" dn="sys/ chassis-1/server-1"/>
  </inConfig>
</configConfMo>
```

Validating the request:

```
/usr/bin/xmllint -schema ./RACK-IN.xsd  /setRackUnit.xml
<configConfMo  cookie="1300242644/ad04d239-d1aa-498d-b074-ccb923066003"
dn="sys/chassis-1/server-1"inHierarchical="false">
  <inConfig>
    <computeServerNode adminPower="down" usrLbl="UCS Cxxx server For Demo"
     availableMemory="16384" dn="sys/chassis-1/server-1"/>
  </inConfig>
</configConfMo>
```

Response:

```
/setRackUnit.xml:3: element computeServerNode: Schemas validity error :
Element 'computeServerNode', attribute 'availableMemory': The attribute 'availableMemory'
is not allowed.
/setRackUnit.xml fails to validate
```

The availableMemory attribute is read-only in the computeServerNodeclass. You can view the read/write attributes in computeServerNode that can be set using configCongMo XML by looking at the computeServerNode definition in RACK-IN.xsd. A sample snippet is as follows:

```
<!--computeServerNode-->
        <xs:element name="computeServerNode" type=" computeServerNode"
substitutionGroup="managedObject"/>
        <xs:complexType name="computeServerNode" mixed="true">

            <xs:attribute name="adminPower">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                         <xs:enumeration value="up"/>
                         <xs:enumeration value="down"/>
                         <xs:enumeration value="soft-shut-down"/>
                         <xs:enumeration value="cycle-immediate"/>
                         <xs:enumeration value="hard-reset-immediate"/>
                         <xs:enumeration value="bmc-reset-immediate"/>
                         <xs:enumeration value="bmc-reset-default"/>
                         <xs:enumeration value="cmos-reset-immediate"/>
                         <xs:enumeration value="diagnostic-interrupt"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>

            <xs:attribute name="usrLbl">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                         <xs:pattern value="[
!#$%&amp;\(\)\*\+,\-\./:;\?@\[\]_\{\|\}~a-zA-Z0-9]{0,64}"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>

            <xs:attribute name="dn" type="referenceObject"/>
            <xs:attribute name="rn" type="referenceRn"/>
            <xs:attribute name="status" type="objectStatus"/>
        </xs:complexType>
```

Example of Use of the RACK-IN.xsd for an Invalid configConfMo Request

Request:

```
$ cat setBootOrder.xml
<configConfMo cookie="1360205300/79c672f0-d519-1519-8004-30339ee53d50"
inHierarchical="true" dn="sys/chassis-1/server-1/boot-policy">
  <inConfig>
    <lsbootDef dn="sys/chassis-1/server-1/boot-policy"   name="boot-policy"
purpose="operational" rebootOnUpdate="no">
      <lsbootVirtualMedia access="read-only" order="2" type="virtual-media"
rn="vm-read-only"/>
      <lsbootVirtualMedia access="read-write" order="3" type="virtual-media"
rn="vm-read-write"/>
      <lsbootLan rn="lan-read-only" access="read-only" order="4" prot="pxe" type="lan"/>
      <lsbootStorage rn="storage-read-write" access="read-write" order="1" type="storage">
       <lsbootLocalStorage rn="local-storage"/>
      </lsbootStorage>
      <lsbootEfi rn="efi-read-only" access="read-only" order="5" type="efi"/>
    </lsbootDef>
  </inConfig>
</configConfMo>
```

Validating the request:

```
/usr/bin/xmllint -schema ./RACK-IN.xsd ./setBootOrder.xml
<configConfMo cookie="1360205300/79c672f0-d519-1519-8004-30339ee53d50"
inHierarchical="true" dn="sys/chassis-1/server-1/boot-policy">
  <inConfig>
    <lsbootDef dn="sys/chassis-1/server-1/boot-policy" name="boot-policy"
purpose="operational" rebootOnUpdate="no">
      <lsbootVirtualMedia access="read-only" order="2" type="virtual-media"
rn="vm-read-only"/>
      <lsbootVirtualMedia access="read-write" order="3" type="virtual-media"
rn="vm-read-write"/>
      <lsbootLan rn="lan-read-only" access="read-only" order="4" prot="pxe" type="lan"/>
      <lsbootStorage rn="storage-read-write" access="read-write" order="1" type="storage">
        <lsbootLocalStorage rn="local-storage"/>
      </lsbootStorage>
      <lsbootEfi rn="efi-read-only" access="read-only" order="5" type="efi"/>
    </lsbootDef>
  </inConfig>
</configConfMo>
```

Response:

```
./setBootOrder.xml:3: element lsbootDef: Schemas validity error :
Element 'lsbootDef', attribute 'name': The attribute 'name' is not allowed.
./setBootOrder.xml:3: element lsbootDef: Schemas validity error :
Element 'lsbootDef', attribute 'purpose': The attribute 'purpose' is not allowed.
./setBootOrder.xml fails to validate
```

Name and purpose attributes of class lsBootDef are read-only and cannot be used in the configConMo/set request.

Example of Use of the RACK-IN.xsd for a Valid configConfMo Request

Request:

```
usr/bin/xmllint -schema ./RACK-IN.xsd  /setRackUnit.xml
<configConfMo  cookie="1300242644/ad04d239-d1aa-498d-b074-ccb923066003"
 dn="sys/chassis-1/server-1" inHierarchical="false">
  <inConfig>
    <computeServerNode adminPower="down" usrLbl="UCS Cxxx server For Demo"
dn="sys/chassis-1/server-1"/>
</inConfig>
</configConfMo>
```

xsd file validates the request and completes the configuration.

Examples of RACK-OUT.xsd Usage

Validating an XML Request Using RACK_OUT.xsd

Example of Use of the RACK-OUT.xsd for a configResolveClass Request

Request:

```
<configResolveClass cookie="1360632361/e892fa10-d57c-157c-8003-f4759de53d50"
inHierarchical="false" classId="computeServerNode"/>'
https://172.xx.219.xx/nuova | xmllint -format -  |  xmllint -schema ./RACK-OUT.xsd -
```

Response:

```
<configResolveClass cookie="1360632361/e892fa10-d57c-157c-8003-f4759de53d50"
response="yes" classId="computeServerNode">
  <outConfigs>
    <computeServerNode dn="sys/chassis-1/server-1" adminPower="policy" availableMemory="16384"
 model="UCSC-Cxxx-M3L"
     memorySpeed="1333" name="UCS Cxxx M3L" numOfAdaptors="1" numOfCores="16"
numOfCoresEnabled="16" numOfCpus="2"
```

```
     numOfEthHostIfs="2" numOfFcHostIfs="3" numOfThreads="32" operPower="on"
     originalUuid="2E5D2295-F32D-48C9-BE8E-BAD36BE174FB" presence="equipped" serverId="1"
serial="FCH1551V030"
    totalMemory="16384" usrLbl="SL2_=+@#$-;,./\" uuid="2E5D2295-F32D-48C9-BE8E-BAD36BE174FB"

     vendor="Cisco Systems Inc"/>
  </outConfigs>
</configResolveClass>
```

Rack-out.xsd validates the output successfully.

Example of Use of the RACK-OUT.xsd for a configResolveClass Request

Request:

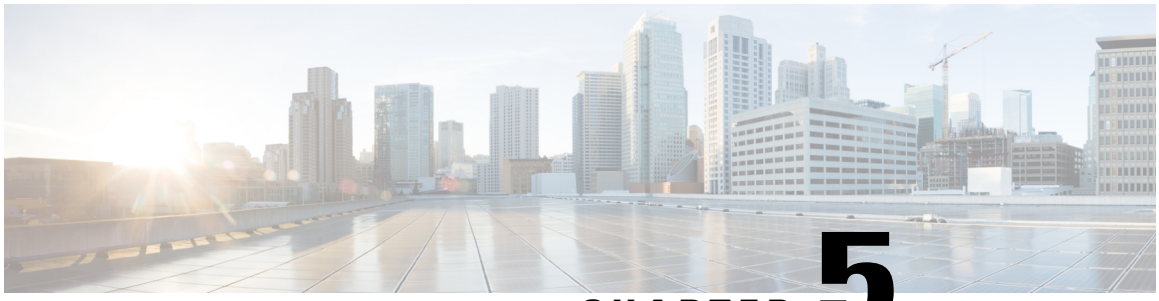
```
$ /usr/bin/curl -k -d'<configResolveClass
cookie="1361150931/a5bacff0-d5f5-15f5-8007-f4759de53d50"
inHierarchical="true" classId="topSystem"/>'  https://172.29.219.74/nuova >
Cxxx_complete_MIT.xml
real    0m35.065s
user    0m0.016s
sys     0m0.044s
```

Validating the response:

```
$ ls -l Cxxx_complete_MIT.xml
-rw-r--r--. 1 sajaffer eng58 64905 Feb 17 17:46 Cxxx_complete_MIT.xml

$ /usr/bin/xmllint -schema   RACK-OUT.xsd   Cxxx_complete_MIT.xml 1>/dev/null
Cxxx_complete_MIT.xml validates
```

The preceding XML requests retrieve the complete management information tree of a C-Series server and validate the response against RACK-OUT.xsd



CHAPTER 5

Cisco IMC XML Object-Access Privileges

This chapter includes the following sections:

Privileges Summary Table

When users are assigned to a role, that role allows certain privileges. Those privileges allow the user access to specific system resources and authorize permission to perform tasks on those resources. The following table lists each privilege and the initial default user role that has been given that privilege.

Internal Name	Label	Description
admin, on page 35	ADMIN	Access to everything
read-only, on page 36	READ_ONLY	Read-only access
user, on page 36	USER	Limited configuration access

Privileges

admin

Purpose

System administration

Responsible Role

Administrator

Controlled Objects

This role is system level. The administrator controls all objects.

read-only

Purpose

Read-only access

Responsible Role

This is not a selectable privilege. All roles have read-only access to all objects. Roles that have read-write privileges on some objects also have read-only access to all other objects.

user

Purpose

Restricted configuration

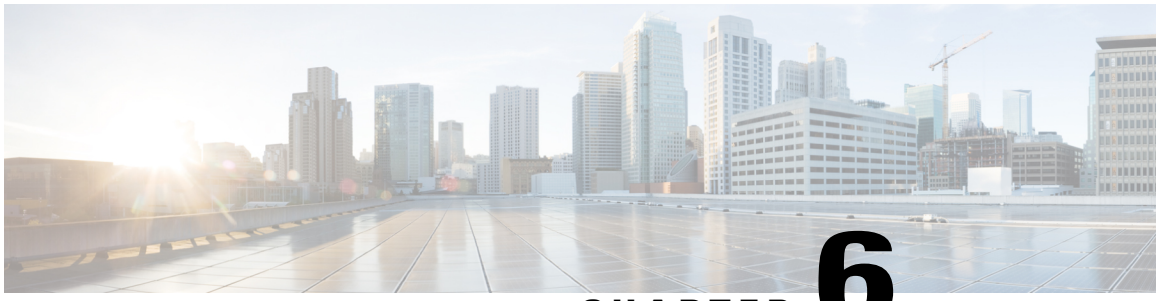
Responsible Role

User

Controlled Objects

This role can perform the following tasks:

- View all information
- Manage the power control options such as power on, power cycle, and power off
- Launch the KVM console and virtual media
- Clear all logs
- Toggle the locator LED
- Set time zone
- Ping



CHAPTER 6

Managing Chassis and Dynamic Storage

This chapter includes the following sections:

Chassis Inventory Tasks

The examples in this section show how to use the Cisco IMC XML API to retrieve chassis inventory details. Each example shows the XML API request followed by the response from Cisco IMC.

This section includes the following examples:

Retrieving Power Supply Properties

Request:

```
<configResolveClass cookie="1256511180/f78548f2-ca76-16ca-8004-aaec921b0ff4"
inHierarchical="false" classId="equipmentPsu"/>
```

Response:

```
<configResolveClass cookie="1473455878/95c48a0a-1a3c-1c1a-8003-be18652a6ca4"
 response="yes" classId="equipmentPsu">
  <outConfigs>
    <equipmentPsu id="1" pid="UCSC-PSU1-1050W" model="PS-2112-9S-LF"
     operability="operable" power="on" presence="equipped" serial="LIT18520J9K"
     thermal="ok" vendor="Cisco Systems Inc" voltage="ok" input="0" maxOutput="0"
     fwVersion="10062013" dn="sys/chassis-1/psu-1"/>
    <equipmentPsu id="2" pid="UCSC-PSU1-1050W" model="PS-2112-9S-LF" operability="operable"

     power="on" presence="equipped" serial="LIT18520JCA" thermal="ok" vendor="Cisco Systems
 Inc"
     voltage="ok" input="0" maxOutput="0" fwVersion="10062013" dn="sys/chassis-1/psu-2"/>
    <equipmentPsu id="3" pid="UCSC-PSU1-1050W" model="PS-2112-9S-LF" operability="operable"

     power="on" presence="equipped" serial="LIT18520JBZ" thermal="ok" vendor="Cisco Systems
 Inc"
     voltage="ok" input="0" maxOutput="0" fwVersion="10062013" dn="sys/chassis-1/psu-3"/>
    <equipmentPsu id="4" pid="UCSC-PSU1-1050W" model="PS-2112-9S-LF" operability="operable"

     power="on" presence="equipped" serial="LIT18520JC1" thermal="ok" vendor="Cisco Systems
 Inc"
     voltage="ok" input="452" maxOutput="404" fwVersion="10062013" dn="sys/chassis-1/psu-4"/>

  </outConfigs>
</configResolveClass>
```

Retrieving Power Supply Properties Using DN

Request:

```
<configResolveDn cookie="1256511180/f78548f2-ca76-16ca-8004-aaec921b0ff4"
inHierarchical="true" dn='sys/chassis-1/psu-1'/>
```

Response:

```
<configResolveDn cookie="1473455878/95c48a0a-1a3c-1c1a-8003-be18652a6ca4"
response="yes" dn="sys/chassis-1/psu-1">
  <outConfig>
    <equipmentPsu id="1" pid="UCSC-PSU1-1050W" model="PS-2112-9S-LF" operability="operable"

     power="on" presence="equipped" serial="LIT18520J9K" thermal="ok" vendor="Cisco Systems
 Inc"
     voltage="ok" input="0" maxOutput="0" fwVersion="10062013" dn="sys/chassis-1/psu-1"/>
  </outConfig>
</configResolveDn>
```

Retrieving Fan Properties

Request:

```
<configResolveClass cookie="1256511180/f78548f2-ca76-16ca-8004-aaec921b0ff4"
inHierarchical="true" classId="equipmentFanModule"/>
```

Response:

```
<configResolveClass cookie="1256511180/f78548f2-ca76-16ca-8004-aaec921b0ff4"
     response="yes" classId="equipmentFanModule">
  <outConfigs>
     <equipmentFanModule id="1" model="" operability="unknown" power="on"
          presence="equipped" serial="" thermal="not-supported" tray="1" vendor=""
          voltage="not-supported" dn="sys/chassis-1/fan-module-1-1">
     <equipmentFan id="1" model="" module="1" operability="unknown" power="on"
          presence="equipped" serial="" thermal="not-supported" tray="1" vendor=""
          voltage="not-supported" rn="fan-1"/>
     <equipmentFan id="2" model="" module="1" operability="unknown"
```

```
      power="on" presence="equipped" serial="" thermal="not-supported"
          tray="1" vendor="" voltage="not-supported" rn="fan-2"/>
    </equipmentFanModule>
     <equipmentFanModule id="2" model="" operability="unknown"
     power="on" presence="equipped" serial="" thermal="not-supported"
          tray="1" vendor="" voltage="not-supported" dn="sys/chassis-1/fan-module-1-2">
     <equipmentFan id="1" model="" module="2" operability="unknown"
     power="on" presence="equipped" serial="" thermal="not-supported"
          tray="1" vendor="" voltage="not-supported" rn="fan-1"/>
     <equipmentFan id="2" model="" module="2" operability="unknown"
     power="on" presence="equipped" serial="" thermal="not-supported"
          tray="1" vendor="" voltage="not-supported" rn="fan-2"/>
    </equipmentFanModule>
     <equipmentFanModule id="3" model="" operability="unknown" power="on"
          presence="equipped" serial="" thermal="not-supported" tray="1"
          vendor="" voltage="not-supported" dn="sys/chassis-1/fan-module-1-3">
     <equipmentFan id="1" model="" module="3" operability="unknown"  power="on"
          presence="equipped" serial="" thermal="not-supported" tray="1"
          vendor="" voltage="not-supported" rn="fan-1"/>
     <equipmentFan id="2" model="" module="3" operability="unknown" power="on"
          presence="equipped" serial="" thermal="not-supported" tray="1"
          vendor="" voltage="not-supported" rn="fan-2"/>
    </equipmentFanModule>
     <equipmentFanModule id="4" model="" operability="unknown" power="on"
          presence="equipped" serial="" thermal="not-supported" tray="1"
          vendor="" voltage="not-supported" dn="sys/chassis-1/fan-module-1-4">
     <equipmentFan id="1" model="" module="4" operability="unknown" power="on"
          presence="equipped" serial="" thermal="not-supported" tray="1" vendor=""
          voltage="not-supported" rn="fan-1"/>
     <equipmentFan id="2" model="" module="4" operability="unknown" power="on"
          presence="equipped" serial="" thermal="not-supported" tray="1" vendor=""
          voltage="not-supported" rn="fan-2"/>
    </equipmentFanModule>
  </outConfigs>
</configResolveClass>
```

Retrieving Fan Properties Using DN

Request:

```
<configResolveDn cookie="1256514416/cc6493cf-cb76-16cb-8005-aaec921b0ff4"
inHierarchical="false" dn='sys/chassis-1/fan-module-1-3/fan-2'/>
```

Response:

```
<configResolveDn cookie="1256514416/cc6493cf-cb76-16cb-8005-aaec921b0ff4"
response="yes" dn="sys/chassis-1/fan-module-1-3/fan-2">
  <outConfig>
    <equipmentFan id="2" model="" module="3" operability="unknown"
     power="on" presence="equipped" serial="" thermal="not-supported"
         tray="1" vendor="" voltage="not-supported"
         dn="sys/chassis-1/fan-module-1-3/fan-2"/>
  </outConfig>
</configResolveDn>
```

Retrieving Indicator LED Status

Request:

```
<configResolveClass cookie="1256514920/8c2af8ff-cb76-16cb-8006-aaec921b0ff4"
  inHierarchical="true" classId="equipmentIndicatorLed"/>
```

Response:

```
<configResolveClass cookie="1256514920/8c2af8ff-cb76-16cb-8006-aaec921b0ff4"
     response="yes" classId="equipmentIndicatorLed">
```

```
  <outConfigs>
    <equipmentIndicatorLed color="green" id="1" name="LED_HLTH_STATUS"
      operState="off" dn="sys/chassis-1/indicator-led-1"/>
    <equipmentIndicatorLed color="green" id="2" name="LED_PSU_STATUS"
      operState="on" dn="sys/chassis-1/indicator-led-2"/>
    <equipmentIndicatorLed color="green" id="3" name="LED_TEMP_STATUS"
      operState="on" dn="sys/chassis-1/indicator-led-3"/>
    <equipmentIndicatorLed color="green" id="4" name="LED_FAN_STATUS"
      operState="on" dn="sys/chassis-1/indicator-led-4"/>
  </outConfigs>
</configResolveClass>
```

Retrieving Indicator LED Status using DN

Request:

```
<configResolveDn cookie="1256515596/c4969312-cc76-16cc-8007-aaec921b0ff4"
     inHierarchical="true" dn='sys/chassis-1/indicator-led-3'/>
```

Response:

```
<configResolveDn cookie="1256515596/c4969312-cc76-16cc-8007-aaec921b0ff4"
    response="yes" dn="sys/chassis-1/indicator-led-3">
  <outConfig>
    <equipmentIndicatorLed color="green" id="3" name="LED_TEMP_STATUS"
      operState="on" dn="sys/chassis-1/indicator-led-3"/>
  </outConfig>
</configResolveDn>
```

Retrieving System Input Output Controller Properties

Request:

```
<configResolveClass cookie="1256515596/c4969312-cc76-16cc-8007-aaec921b0ff4"
     inHierarchical="true" classId="equipmentIOCard"/>
```

Response:

```
<configResolveClass cookie="1256515596/c4969312-cc76-16cc-8007-aaec921b0ff4"
     response="yes" classId="equipmentIOCard">
  <outConfigs>
    <equipmentIOCard id="1" description="SIOC - System Input Output Controller"
      dn="sys/chassis-1/slot-1">
    <mgmtController rn="mgmt" model="NA" serial="NA" subject="sioc"
      vendor="Cisco Systems Inc">
    <firmwareRunning rn="fw-system"
      description="System IO Controller currently running firmware version"
      deployment="system" type="sioc" version="0.0(4.r202233)"/>
    <firmwareUpdatable rn="fw-updatable"
      description="System IO Controller backup firmware version"
      adminState="triggered" deployment="backup" operState="ready"
      version="2.0(6.11)" protocol="none" remoteServer="" remotePath="" user=""
      pwd="" progress="Success" type="sioc"/>
    <firmwareBootDefinition rn="fw-boot-def" type="sioc">
     <firmwareBootUnit rn="bootunit-combined"
      description="System Input Out Controller startup firmware version"
      adminState="triggered" image="running" resetOnActivate="no"
      type="combined" version="0.0(4.r202233)"/>
    </firmwareBootDefinition>
        <mgmtIf rn="if-1" description="Chassis Management Interface Network Settings"
         id="1" extEnabled="yes" extIp="10.106.145.78" extMask="255.255.255.0"
         extGw="10.106.145.1" ifType="physical" mac="F4:0F:1B:92:EC:AA"
         hostname="UCS-C3260-1" dhcpEnable="yes" dnsUsingDhcp="yes" ddnsEnable="yes"
         ddnsDomain="" dnsPreferred="171.70.168.183" dnsAlternate="0.0.0.0"
         nicMode="dedicated" vicSlot="1" nicRedundancy="none"
         vlanEnable="no" vlanId="1" vlanPriority="0" portProfile=""
         v6extEnabled="no" v6extIp="::" v6extGw="::" v6prefix="64"
```

```
          v6linkLocal="::" v6SlaacIp="::" v6dhcpEnable="no" v6dnsUsingDhcp="no"
          v6dnsPreferred="::" v6dnsAlternate="::" subject="blade"/>
      </mgmtController>
    </equipmentIOCard>
    <equipmentIOCard id="2" description="SIOC - System Input Output Controller"
     dn="sys/chassis-1/slot-2">
      <mgmtController rn="mgmt" model="NA" serial="NA" subject="sioc"
       vendor="Cisco Systems Inc">
        <firmwareRunning rn="fw-system"
         description="System IO Controller currently running firmware version"
         deployment="system" type="sioc" version="0.0(4.r202233)"/>
        <firmwareUpdatable rn="fw-updatable"
         description="System IO Controller backup firmware version"
         adminState="triggered" deployment="backup" operState="ready"
         version="0.0(4.r202050)" protocol="none" remoteServer=""
         remotePath="" user="" pwd="" progress="Success" type="sioc"/>
        <firmwareBootDefinition rn="fw-boot-def" type="sioc">
           <firmwareBootUnit rn="bootunit-combined"
            description="System Input Out Controller startup firmware version"
            adminState="triggered" image="running" resetOnActivate="no"
            type="combined" version="0.0(4.r202233)"/>
        </firmwareBootDefinition>
        <mgmtIf rn="if-1" description="Chassis Management Interface Network Settings"
         id="1" extEnabled="yes" extIp="10.106.145.253" extMask="255.255.255.0"
         extGw="10.106.145.1" ifType="physical" mac="F4:0F:1B:92:7A:C4"
         hostname="UCS-C3260-2" dhcpEnable="yes" dnsUsingDhcp="yes"
         ddnsEnable="yes" ddnsDomain="" dnsPreferred="171.70.168.183"
         dnsAlternate="0.0.0.0" nicMode="dedicated" vicSlot="1" nicRedundancy="none"
         vlanEnable="no" vlanId="1" vlanPriority="0" portProfile="" v6extEnabled="no"
         v6extIp="::" v6extGw="::" v6prefix="64" v6linkLocal="::" v6SlaacIp="::"
         v6dhcpEnable="no" v6dnsUsingDhcp="no" v6dnsPreferred="::"
         v6dnsAlternate="::" subject="blade"/>
      </mgmtController>
    </equipmentIOCard>
  </outConfigs>
</configResolveClass>
```

Retrieving System Input Output Controller Properties using DN

Request:

```
<configResolveDn cookie="1256516771/c56962d3-cb76-16cb-8008-aaec921b0ff4"
      inHierarchical="true" dn='sys/chassis-1/slot-2'/>
```

Response:

```
<configResolveDn cookie="1256516771/c56962d3-cb76-16cb-8008-aaec921b0ff4"
response="yes" dn="sys/chassis-1/slot-2">
  <outConfig>
    <equipmentIOCard id="2" description="SIOC - System Input Output Controller"
     dn="sys/chassis-1/slot-2">
      <mgmtController rn="mgmt" model="NA" serial="NA"
       subject="sioc" vendor="Cisco Systems Inc">
        <firmwareRunning rn="fw-system"
         description="System IO Controller currently running firmware version"
         deployment="system" type="sioc" version="0.0(4.r202233)"/>
        <firmwareUpdatable rn="fw-updatable"
         description="System IO Controller backup firmware version"
         adminState="triggered" deployment="backup" operState="ready"
         version="0.0(4.r202050)" protocol="none" remoteServer=""
         remotePath="" user="" pwd="" progress="Success" type="sioc"/>
        <firmwareBootDefinition rn="fw-boot-def" type="sioc">
          <firmwareBootUnit rn="bootunit-combined"
           description="System Input Out Controller startup firmware version"
           adminState="triggered" image="running" resetOnActivate="no"
           type="combined" version="0.0(4.r202233)"/>
        </firmwareBootDefinition>
        <mgmtIf rn="if-1" description="Chassis Management Interface Network Settings"
         id="1" extEnabled="yes" extIp="10.106.145.253" extMask="255.255.255.0"
         extGw="10.106.145.1" ifType="physical" mac="F4:0F:1B:92:7A:C4"
         hostname="UCS-C3260-2" dhcpEnable="yes" dnsUsingDhcp="yes"
```

```
        ddnsEnable="yes" ddnsDomain="" dnsPreferred="171.70.168.183"
        dnsAlternate="0.0.0.0" nicMode="dedicated" vicSlot="1"
        nicRedundancy="none" vlanEnable="no" vlanId="1"
        vlanPriority="0" portProfile="" v6extEnabled="no"
        v6extIp="::" v6extGw="::" v6prefix="64" v6linkLocal="::"
        v6SlaacIp="::" v6dhcpEnable="no" v6dnsUsingDhcp="no" v6dnsPreferred="::"
        v6dnsAlternate="::" subject="blade"/>
      </mgmtController>
    </equipmentIOCard>
  </outConfig>
</configResolveDn>
```

Dynamic Storage Tasks

The examples in this section show how to use the Cisco IMC XML API to retrieve and perform dynamic storage tasks. Each example shows the XML API request followed by the response from Cisco IMC.

This section includes the following examples:

Retrieving SAS Expander Properties

Request:

```
<configResolveClass cookie="0960329953/d2022d82-6a69-196a-8002-82f92a02b884"
 inHierarchical="true" classId="chassisStorage"/>
```

Response:

```
<configResolveClass cookie="0960329953/d2022d82-6a69-196a-8002-82f92a02b884" res
ponse="yes" classId="chassisStorage">
 <outConfigs>
  <chassisStorage dn="sys/chassis-1/storage" description="Chassis scope dynamic storage
management">
   <storageChassisDisk slot="1" diskstate="done" vendor="WD" productId="WD400
    1FYYG-01SL3" revision="VR07" serial="WMC1F1927484" blocksize="512" blockcount="7
    814037168" size="3.63 TB" expander1linkspeed="6.0 Gb" expander2linkspeed="6.0 Gb
    " sasaddress1="50000c0f01e85d26" sasaddress2="50000c0f01e85d27" ownership="none"
    health="good" rn="disk-1"/>
  <storageChassisDisk slot="2" diskstate="done" vendor="WD" productId="WD400
   1FYYG-01SL3" revision="VR07" serial="WMC1F1930758" blocksize="512" blockcount="7
   814037168" size="3.63 TB" expander1linkspeed="6.0 Gb" expander2linkspeed="6.0 Gb
   " sasaddress1="50000c0f01e85bd6" sasaddress2="50000c0f01e85bd7" ownership="none"
   health="good" rn="disk-2"/>
  <storageChassisDisk slot="3" diskstate="done"
   vendor="WD" productId="WD4001FYYG-01SL3" revision="VR07"
   serial="WMC1F1930961" blocksize="512" blockcount="7814037168"
   size="3.63 TB" expander1linkspeed="6.0 Gb" expander2linkspeed="6.0 Gb"
   sasaddress1="50000c0f01e85a0a" sasaddress2="50000c0f01e85a0b"
   ownership="none" health="good" rn="disk-3"/>
```

```
...
...
...

     <storageChassisDisk slot="60" diskstate="done"
      vendor="WD" productId="WD4001FYYG-01SL3" revision="VR07"
      serial="WMC1F1926558" blocksize="512" blockcount="7814037168"
      size="3.63 TB" expander1linkspeed="6.0 Gb" expander2linkspeed="6.0 Gb"
      sasaddress1="50000c0f012eb1f6" sasaddress2="50000c0f012eb1f7"
      ownership="server1" health="good" rn="disk-60"/>
      <storageChassisDiskSlotEp slot="1" health="good" presence="done" rn="diskslot-1">
       <computeServerRef slot="1" diskstate="done"
       ownership="none" rn="server-ref-none"/>
      </storageChassisDiskSlotEp>
      <storageChassisDiskSlotEp slot="2"
       health="good" presence="done" rn="diskslot-2">
       <computeServerRef slot="2" diskstate="done"
        ownership="none" rn="server-ref-none"/>
      </storageChassisDiskSlotEp>
      <storageChassisDiskSlotEp slot="54" health="fault"
        presence="Drive Discovery Error" rn="diskslot-54">
        <computeServerRef slot="54" diskstate="Drive Discovery Error"
         ownership="none" rn="server-ref-none"/>
        <faultInst ack="yes" cause="equipment-inoperable"
         code="F0181" created="2000-06-06T07:51:39"
         descr="Storage Local disk 54 is inoperable: reseat or replace the storage drive
54"
         affectedDN="sys/chassis-1/storage/diskslot-54"
         highestSeverity="critical" id="3506451968"
         lastTransition="2000-06-06T07:51:28" lc="flapping"
         occur="4" origSeverity="cleared" prevSeverity="cleared"
         rule="fltStorageLocalDiskInoperable"
         severity="major" tags="storage" type="server" rn="fault-F0181"/>
         </storageChassisDiskSlotEp>
....
....
....

    </chassisStorage>
  </outConfigs>
</configResolveClass>
```

Retrieving Chassis Storage Disk Slot

Request:

```
<configResolveClass cookie="0960331076/16b9fadd-6a69-196a-8003-82f92a02b884"
 inHierarchical="true" classId="storageChassisDiskSlotEp"/>
```

Response:

```
<configResolveClass cookie="0960331076/16b9fadd-6a69-196a-8003-82f92a02b884"
 response="yes" classId="storageChassisDiskSlotEp">
  <outConfigs>
    <storageChassisDiskSlotEp slot="1" health="good"
     presence="done" dn="sys/chassis-1/storage/diskslot-1">
      <computeServerRef slot="1" diskstate="done"
       ownership="none" rn="server-ref-none"/>
    </storageChassisDiskSlotEp>
    <storageChassisDiskSlotEp slot="2" health="good"
     presence="done" dn="sys/chassis-1/storage/diskslot-2">
      <computeServerRef slot="2" diskstate="done"
       ownership="none" rn="server-ref-none"/>
    </storageChassisDiskSlotEp>
...
...
...
    <storageChassisDiskSlotEp slot="54" health="fault" presence="Drive Discovery
     Error" dn="sys/chassis-1/storage/diskslot-54">
```

```
      <computeServerRef slot="54" diskstate="Drive Discovery Error"
        ownership="none" rn="server-ref-none"/>
      <faultInst ack="yes" cause="equipment-inoperable"
       code="F0181" created="2000-06-06T07:51:39"
       descr="Storage Local disk 54 is inoperable: reseat or replace the storage drive 54"
       affectedDN="sys/chassis-1/storage/diskslot-54"
       highestSeverity="critical" id="3506451968"
       lastTransition="2000-06-06T07:51:28" lc="flapping"
       occur="4" origSeverity="cleared" prevSeverity="cleared"
       rule="fltStorageLocalDiskInoperable" severity="major"
       tags="storage" type="server" rn="fault-F0181"/>
    </storageChassisDiskSlotEp>
...
...
...
  </outConfigs>
</configResolveClass>
```

Retrieving Chassis Storage Disk Slot Using DN

Request:

```
<configResolveDn cookie="0960344358/94c74f08-6e69-196e-8005-82f92a02b884"
 inHierarchical="true" dn='sys/chassis-1/storage/diskslot-12'/>
```

Response:

```
<configResolveDn cookie="0960344358/94c74f08-6e69-196e-8005-82f92a02b884"
 response="yes" dn="sys/chassis-1/storage/diskslot-12">
  <outConfig>
    <storageChassisDiskSlotEp slot="12" health="good"
     presence="done" dn="sys/chassis-1/storage/diskslot-12">
      <computeServerRef slot="12" diskstate="done"
       ownership="server1" rn="server-ref-server1"/>
    </storageChassisDiskSlotEp>
  </outConfig>
</configResolveDn>
>
```

Retrieving Chassis Storage Disk

Request:

```
<configResolveClass cookie="0960344358/94c74f08-6e69-196e-8005-82f92a02b884"
 inHierarchical="true" classId="storageChassisDisk"/>
```

Response:

```
<configResolveClass cookie="0960344358/94c74f08-6e69-196e-8005-82f92a02b884"
  response="yes" classId="storageChassisDisk">
  <outConfigs>
    <storageChassisDisk slot="1" diskstate="done" vendor="WD"
      productId="WD4001FYYG-01SL3" revision="VR07"
      serial="WMC1F1927484" blocksize="512" blockcount="7814037168"
      size="3.63 TB" expander1linkspeed="6.0 Gb" expander2linkspeed="6.0 Gb"
      sasaddress1="50000c0f01e85d26" sasaddress2="50000c0f01e85d27"
      ownership="none" health="good" dn="sys/chassis-1/storage/disk-1"/>
    <storageChassisDisk slot="2" diskstate="done"
     vendor="WD" productId="WD4001FYYG-01SL3" revision="VR07"
     serial="WMC1F1930758" blocksize="512" blockcount="7814037168"
     size="3.63 TB" expander1linkspeed="6.0 Gb" expander2linkspeed="6.0 Gb"
     sasaddress1="50000c0f01e85bd6" sasaddress2="50000c0f01e85bd7" ownership="none"
     health="good" dn="sys/chassis-1/storage/disk-2"/>

...
...
...

     <storageChassisDisk slot="59" diskstate="done" vendor="WD"
```

```
      productId="WD4001FYYG-01SL3" revision="VR07"
      serial="WMC1F1928804" blocksize="512" blockcount="7814037168"
      size="3.63 TB" expander1linkspeed="6.0 Gb" expander2linkspeed="6.0 Gb"
      sasaddress1="50000c0f01e85532" sasaddress2="50000c0f01e85533"
      ownership="server1" health="good" dn="sys/chassis-1/storage/disk-59"/>
    <storageChassisDisk slot="60" diskstate="done" vendor="WD"
      productId="WD4001FYYG-01SL3" revision="VR07"
      serial="WMC1F1926558" blocksize="512" blockcount="7814037168"
      size="3.63 TB" expander1linkspeed="6.0 Gb" expander2linkspeed="6.0 Gb"
      sasaddress1="50000c0f012eb1f6" sasaddress2="50000c0f012eb1f7"
      ownership="server1" health="good" dn="sys/chassis-1/storage/disk-60"/>
  </outConfigs>
</configResolveClass>
```

Retrieving Chassis Storage Disk Using DN

Request:

```
<configResolveDn cookie="0955686332/993f316f-3165-1531-8002-a35a6a1b0ff4"
inHierarchical="false" dn='sys/chassis-1/storage/disk-1'/>
```

Response:

```
<configResolveDn cookie="0960347338/0cad041e-6f69-196f-8007-82f92a02b884"
 response="yes" dn="sys/chassis-1/storage/disk-23">
  <outConfig>
    <storageChassisDisk slot="23" diskstate="done"
     vendor="TOSHIBA" productId="MG03SCA400" revision="5702"
     serial="14O0A082FVU4" blocksize="512" blockcount="7814037168"
     size="3.63 TB" expander1linkspeed="6.0 Gb" expander2linkspeed="6.0 Gb"
     sasaddress1="5000039548e05956" sasaddress2="5000039548e05957"
     ownership="server1" health="good" dn="sys/chassis-1/storage/disk-23"/>
  </outConfig>
</configResolveDn>
```

Assigning Physical Drives to Servers

Request:

```
<configConfMo cookie='0955691347/295e610a-3365-1533-8002-a35a6a1b0ff4'
 dn='sys/chassis-1/storage/zone-drive'>
<inConfig>
        <storageChassisDiskSlotZoneHelper
         dn='sys/chassis-1/storage/zone-drive' slotList="1,2,3"
         ownership="server2" adminState="trigger"/>
</inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/storage/zone-drive"
    cookie="0955692275/224d31b6-3265-1532-8002-a35a6a1b0ff4" response="yes">
  <outConfig>
    <storageChassisDiskSlotZoneHelper dn="sys/chassis-1/storage/zone-drive"
    adminState="triggered" status="modified"/>
  </outConfig>
</configConfMo>
```

Unassigning Physical Drives

Request:

```
<configConfMo cookie='0955692275/224d31b6-3265-1532-8002-a35a6a1b0ff4'
 dn='sys/chassis-1/storage/zone-drive'>
<inConfig>
        <storageChassisDiskSlotZoneHelper
         dn='sys/chassis-1/storage/zone-drive' slotList="1,2,3"
```

```
        ownership="none" adminState="trigger"/>
</inConfig>
</configConfMo>
```

Response:

```
<configConfMo cookie="0955692275/224d31b6-3265-1532-8002-a35a6a1b0ff4"
 response="yes" dn="sys/chassis-1/storage/zone-drive">
  <outConfig>
    <storageChassisDiskSlotZoneHelper
     dn="sys/chassis-1/storage/zone-drive"
     adminState="triggered" status="modified"/>
  </outConfig>
</configConfMo>
```

Managing PID Catalog

The examples in this section show how to use the Cisco IMC XML API to retrieve and upload PID catalog. Each example shows the XML API request followed by the response from Cisco IMC.

This section includes the following examples:

Uploading PID Catalog

Request:

```
<configConfMo cookie="1470041969/37207236-ff38-18ff-8003-4d32145daae4"
dn="sys/chassis-1/pid/upload-catalog" inHierarchical="false">
<inConfig>
 <uploadPIDCatalog protocol="tftp"  remoteServer="10.104.236.99"
  remoteFile="latestpid.tar.gz" dn="sys/chassis-1/pid/upload-catalog"/>
  </inConfig>
</configConfMo>
```

TFTP used in the preceding example is the default protocol. You can also download the LDAP CA certificate using the other available protocols such as the FTP, SFTP, SCP and HTTP.

Response:

```
<configConfMo dn="sys/chassis-1/pid/upload-catalog"
cookie="1470041969/37207236-ff38-18ff-8003-4d32145daae4" response="yes">
<outConfig>
 <uploadPIDCatalog dn="sys/chassis-1/pid/upload-catalog" protocol="none" remoteServer=""
  remoteFile="" user="" pwd="" pidUploadStatus="Upload in Progress" status="modified" >
 </uploadPIDCatalog>
 </outConfig>
```

```
</configConfMo>
```

Activating PID Catalog on Server Node 1

Request:

```
<configConfMo cookie="1462971767/5e320cff-9032-1290-8009-b1f9cc1a3500"
  dn="sys/chassis-1/server-1/board/pid/activate-catalog" inHierarchical="false">
 <inConfig>
  <activatePIDCatalog  adminState="trigger"
   dn="sys/chassis-1/server-1/board/pid/activate-catalog"/>
  </inConfig>
 </configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/server-1/board/pid/activate-catalog"
 cookie="1470739643/d6acc6cc-a139-19a1-8002-5a45145daae4" response="yes">
<outConfig>
 <activatePIDCatalog dn="sys/chassis-1/server-1/board/pid/activate-catalog"
  adminState="triggered" pidActivationStatus="Activation Successful" status="modified">
 </activatePIDCatalog>
</outConfig>
</configConfMo>
```

Activating PID Catalog on Server Node 2

Request:

```
<configConfMo cookie="1462971767/5e320cff-9032-1290-8009-b1f9cc1a3500"
  dn="sys/chassis-1/server-2/board/pid/activate-catalog" inHierarchical="false">
 <inConfig>
  <activatePIDCatalog  adminState="trigger"
   dn="sys/chassis-1/server-2/board/pid/activate-catalog"/>
  </inConfig>
 </configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/server-2/board/pid/activate-catalog"
 cookie="1470739643/d6acc6cc-a139-19a1-8002-5a45145daae4" response="yes">
<outConfig>
 <activatePIDCatalog dn="sys/chassis-1/server-2/board/pid/activate-catalog"
  adminState="triggered" pidActivationStatus="Activation Successful" status="modified">
 </activatePIDCatalog>
</outConfig>
</configConfMo>
```

Viewing PID Catalog

Request:

```
<configResolveClass cookie="1473455878/95c48a0a-1a3c-1c1a-8003-be18652a6ca4"
inHierarchical="false" classId="pidCatalog"/>
```

Response:

```
<configResolveClass cookie="1473455878/95c48a0a-1a3c-1c1a-8003-be18652a6ca4"
 response="yes" classId="pidCatalog">
  <outConfigs>
    <pidCatalog dn="sys/chassis-1/server-1/board/pid" name="Cisco
     Product Identifiers (PID)" version="2.0(13a)09"/>
    <pidCatalog dn="sys/chassis-1/server-2/board/pid" name="Cisco
     Product Identifiers (PID)" version="2.0(13a)09"/>
```

```
  </outConfigs>
</configResolveClass>
```

Retrieving PID Catalog of CPUs

Request:

```
<configResolveClass cookie='1282522267/f91bd068-8e72-1e72-8002-7e573e8f835c'
inHierarchical='false' classId='pidCatalogCpu'/>
```

Response:

```
<configResolveClass cookie="1470739643/d6acc6cc-a139-19a1-8002-5a45145daae4"
response="yes" classId="pidCatalogCpu">
 <outConfigs>
  <pidCatalogCpu id="1" socketdesignation="CPU1" description="Intel(R)
   Xeon(R) CPU E5-2620 v4 @ 2.10GHz/85W 8C/20MB Cache/DDR4 2133MHz"
   pid="UCS-CPU-E52620E" model="Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz"
   signature="Type 0, Family 6, Model 79, Stepping 1" currentspeed="2100"
   operState="Enabled" dn="sys/chassis-1/server-1/board/pid/pid-cpu-1" >
  </pidCatalogCpu>
  <pidCatalogCpu id="2" socketdesignation="CPU2" description="Intel(R)
   Xeon(R) CPU E5-2620 v4 @ 2.10GHz/85W 8C/20MB Cache/DDR4 2133MHz"
   pid="UCS-CPU-E52620E" model="Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz"
   signature="Type 0, Family 6, Model 79, Stepping 1" currentspeed="2100"
   operState="Enabled" dn="sys/chassis-1/server-1/board/pid/pid-cpu-2" >
  </pidCatalogCpu>
  <pidCatalogCpu id="1" socketdesignation="CPU1" description="Intel(R)
   Xeon(R) CPU E5-2695 v4 @ 2.10GHz/120W 18C/45MB Cache/DDR4 2400MHz"
   pid="UCS-CPU-E52695E" model="Intel(R) Xeon(R) CPU E5-2695 v4 @ 2.10GHz"
   signature="Type 0, Family 6, Model 79, Stepping 1" currentspeed="2100"
   operState="Enabled" dn="sys/chassis-1/server-2/board/pid/pid-cpu-1" >
  </pidCatalogCpu>
  <pidCatalogCpu id="2" socketdesignation="CPU2" description="Intel(R)
   Xeon(R) CPU E5-2695 v4 @ 2.10GHz/120W 18C/45MB Cache/DDR4 2400MHz"
   pid="UCS-CPU-E52695E" model="Intel(R) Xeon(R) CPU E5-2695 v4 @ 2.10GHz"
   signature="Type 0, Family 6, Model 79, Stepping 1" currentspeed="2100"
   operState="Enabled" dn="sys/chassis-1/server-2/board/pid/pid-cpu-2" >
  </pidCatalogCpu>
 </outConfigs>
</configResolveClass>
```

Retrieving PID Catalog of DIMMs

Request:

```
<configResolveClass cookie='1282522267/f91bd068-8e72-1e72-8002-7e573e8f835c'
inHierarchical='false' classId='pidCatalogDimm'/>
```

Response:

```
<configResolveClass cookie="1470739643/d6acc6cc-a139-19a1-8002-5a45145daae4"
response="yes" classId="pidCatalogDimm">
<outConfigs>
 </pidCatalogDimm>
 <pidCatalogDimm name="DIMM_G2" description="32GB DDR4-2400-MHz RDIMM/PC4-19200/dual
rank/x4/1.2v"
  pid="UCS-MR-1X322RV-A" mfgid="0xCE00" manufacturer="Samsung" serialnumber="02EB3481"
  model="M393A4K40BB1-CRC" capacity="32768 MB" speed="2400" datawidth="64 bits"
  operability="Operable" dn="sys/chassis-1/server-2/board/pid/pid-dimm-DIMM_G2" >
 </pidCatalogDimm>
   .
   .
   .
   .
  <pidCatalogDimm name="DIMM_G2" description="32GB DDR4-2400-MHz RDIMM/PC4-19200/dual
rank/x4/1.2v"
   pid="UCS-MR-1X322RV-A" mfgid="0xCE00" manufacturer="Samsung" serialnumber="02EB33FD"
```

```
   model="M393A4K40BB1-CRC" capacity="32768 MB" speed="2400" datawidth="64 bits"
   operability="Operable" dn="sys/chassis-1/server-2/board/pid/pid-dimm-DIMM_G1" >
  </pidCatalogDimm>
  <pidCatalogDimm name="DIMM_G2" description="32GB DDR4-2400-MHz RDIMM/PC4-19200/dual
rank/x4/1.2v"
   pid="UCS-MR-1X322RV-A" mfgid="0xCE00" manufacturer="Samsung" serialnumber="02EB3481"
   model="M393A4K40BB1-CRC" capacity="32768 MB" speed="2400" datawidth="64 bits"
   operability="Operable" dn="sys/chassis-1/server-2/board/pid/pid-dimm-DIMM_G2" >
  </pidCatalogDimm>
 </outConfigs>
</configResolveClass>
```

Retrieving PID Catalog of HDDs

Request:

```
<configResolveClass cookie='1282522267/f91bd068-8e72-1e72-8002-7e573e8f835c'
inHierarchical='false' classId='pidCatalogHdd'/>
```

Response:

```
<configResolveClass cookie="1470739643/d6acc6cc-a139-19a1-8002-5a45145daae4"
response="yes" classId="pidCatalogHdd">
 <outConfigs>
  <pidCatalogHdd disk="8" controller="SBMezz1" description="4TB 6Gb SAS 7.2K RPM 3.5"
   pid="UCSC-C3X60-HD4TB" vendor="WD" model="WD4001FYYG-01SL3"
   serialnumber="WMC1F1927555"  dn="sys/chassis-1/server-1/board/pid/pid-hdd-8" >
  </pidCatalogHdd>
  <pidCatalogHdd disk="9" controller="SBMezz1" description="4TB 6Gb SAS 7.2K RPM 3.5"
   pid="UCSC-C3X60-HD4TB" vendor="WD" model="WD4001FYYG-01SL3"
   serialnumber="WMC1F1930050" dn="sys/chassis-1/server-1/board/pid/pid-hdd-9" >
  </pidCatalogHdd>
  <pidCatalogHdd disk="6" controller="SBMezz1" description="4TB 6Gb SAS 7.2K RPM 3.5"
   pid="UCSC-C3X60-HD4TB" vendor="WD" model="WD4001FYYG-01SL3"
   serialnumber="WMC1F1927672" dn="sys/chassis-1/server-2/board/pid/pid-hdd-6" >
  </pidCatalogHdd>
  <pidCatalogHdd disk="7" controller="SBMezz1" description="4TB 6Gb SAS 7.2K RPM 3.5"
   pid="UCSC-C3X60-HD4TB" vendor="WD" model="WD4001FYYG-01SL3" serialnumber="WMC1F1928331"

   dn="sys/chassis-1/server-2/board/pid/pid-hdd-7" >
  </pidCatalogHdd>
  <pidCatalogHdd disk="10" controller="SBMezz1" description="4TB 6Gb SAS 7.2K RPM 3.5"
   pid="UCSC-C3X60-HD4TB" vendor="WD" model="WD4001FYYG-01SL3" serialnumber="WMC1F1928305"

  dn="sys/chassis-1/server-2/board/pid/pid-hdd-10" >
  </pidCatalogHdd>
 <pidCatalogHdd disk="11" controller="SBMezz1" description="4TB 6Gb SAS 7.2K RPM 3.5"
  pid="UCSC-C3X60-HD4TB" vendor="TOSHIBA" model="MG03SCA400" serialnumber="14P0A0E8FVU4"
   dn="sys/chassis-1/server-2/board/pid/pid-hdd-11" >
  </pidCatalogHdd>
  <pidCatalogHdd disk="12" controller="SBMezz1" description="4TB 6Gb SAS 7.2K RPM 3.5"
   pid="UCSC-C3X60-HD4TB" vendor="TOSHIBA" model="MG03SCA400" serialnumber="14P0A069FVU4"
   dn="sys/chassis-1/server-2/board/pid/pid-hdd-12" >
  </pidCatalogHdd>
  <pidCatalogHdd disk="13" controller="SBMezz1" description="4TB 6Gb SAS 7.2K RPM 3.5"
   pid="UCSC-C3X60-HD4TB" vendor="TOSHIBA" model="MG03SCA400" serialnumber="14P0A07BFVU4"
   dn="sys/chassis-1/server-2/board/pid/pid-hdd-13" >
  </pidCatalogHdd>
  <pidCatalogHdd disk="14" controller="SBMezz1" description="4TB 6Gb SAS 7.2K RPM 3.5"
   pid="UCSC-C3X60-HD4TB" vendor="TOSHIBA" model="MG03SCA400" serialnumber="14O0A06EFVU4"
   dn="sys/chassis-1/server-2/board/pid/pid-hdd-14" >
  </pidCatalogHdd>
  <pidCatalogHdd disk="28" controller="SBMezz1" description="4TB 6Gb SAS 7.2K RPM 3.5"
   pid="UCSC-C3X60-HD4TB" vendor="TOSHIBA" model="MG03SCA400" serialnumber="14O0A08MFVU4"
  dn="sys/chassis-1/server-2/board/pid/pid-hdd-28" >
  </pidCatalogHdd>
 </outConfigs>
</configResolveClass>
```

Retrieving PID Catalog of PCI Adapters

Request:

```
<configResolveClass cookie='1282522267/f91bd068-8e72-1e72-8002-7e573e8f835c'
inHierarchical='false' classId='pidCatalogPCIAdapter'/>
```

Response:

```
<configResolveClass cookie="1470739643/d6acc6cc-a139-19a1-8002-5a45145daae4"
response="yes" classId="pidCatalogPCIAdapter">
<outConfigs>
 <pidCatalogPCIAdapter slot="SBNVMe1" description="800GB 2.5in NVMe based
  PCIe SSD drive" pid="UCSC-C3K-NV8" vendor="0x1c58" device="0x0003" subvendor="0x1137"
  subdevice="0x019e" dn="sys/chassis-1/server-1/board/pid/pid-pciadapter-SBNVMe1" >
 </pidCatalogPCIAdapter>
 <pidCatalogPCIAdapter slot="SIOC1" description="Cisco UCS 40Gb Ethernet"
  pid="UCSC-C3260-SIOC" vendor="0x1137" device="0x0042" subvendor="0x1137"
  subdevice="0x0157" dn="sys/chassis-1/server-1/board/pid/pid-pciadapter-SIOC1" >
 </pidCatalogPCIAdapter>
 <pidCatalogPCIAdapter slot="SBMezz1" description="Cisco UCS C3000 RAID controller
  for M4 Server Blade with 4G RAID Cache" pid="UCS-C3K-M4RAID" vendor="0x1000"
  device="0x00ce" subvendor="0x1137" subdevice="0x0197"
 dn="sys/chassis-1/server-1/board/pid/pid-pciadapter-SBMezz1" >
 </pidCatalogPCIAdapter>
 <pidCatalogPCIAdapter slot="SIOC2" description="Cisco UCS 40Gb Ethernet" pid="N/A"
  vendor="0x1137" device="0x0042" subvendor="0x1137" subdevice="0x0157"
 dn="sys/chassis-1/server-2/board/pid/pid-pciadapter-SIOC2" >
 </pidCatalogPCIAdapter>
<pidCatalogPCIAdapter slot="SBMezz1" description="Cisco UCS C3000 RAID controller
 for M4 Server Blade with 4G RAID Cache" pid="UCSC-C3K-M4RAID" vendor="0x1000"
 device="0x00ce" subvendor="0x1137" subdevice="0x0197"
 dn="sys/chassis-1/server-2/board/pid/pid-pciadapter-SBMezz1" >
 </pidCatalogPCIAdapter>
</outConfigs>
</configResolveClass>
```

Managing PID Catalog

The examples in this section show how to use the Cisco IMC XML API to retrieve and upload PID catalog. Each example shows the XML API request followed by the response from Cisco IMC.

This section includes the following examples:

Uploading PID Catalog

Request:

```
<configConfMo cookie="1470041969/37207236-ff38-18ff-8003-4d32145daae4"
dn="sys/chassis-1/pid/upload-catalog" inHierarchical="false">
<inConfig>
 <uploadPIDCatalog protocol="tftp"  remoteServer="10.104.236.99"
  remoteFile="latestpid.tar.gz" dn="sys/chassis-1/pid/upload-catalog"/>
  </inConfig>
</configConfMo>
```

TFTP used in the preceding example is the default protocol. You can also download the LDAP CA certificate using the other available protocols such as the FTP, SFTP, SCP and HTTP.

Response:

```
<configConfMo dn="sys/chassis-1/pid/upload-catalog"
cookie="1470041969/37207236-ff38-18ff-8003-4d32145daae4" response="yes">
<outConfig>
 <uploadPIDCatalog dn="sys/chassis-1/pid/upload-catalog" protocol="none" remoteServer=""
  remoteFile="" user="" pwd="" pidUploadStatus="Upload in Progress" status="modified" >
 </uploadPIDCatalog>
 </outConfig>
</configConfMo>
```

Activating PID Catalog on Server Node 1

Request:

```
<configConfMo cookie="1462971767/5e320cff-9032-1290-8009-b1f9cc1a3500"
  dn="sys/chassis-1/server-1/board/pid/activate-catalog" inHierarchical="false">
 <inConfig>
  <activatePIDCatalog  adminState="trigger"
   dn="sys/chassis-1/server-1/board/pid/activate-catalog"/>
  </inConfig>
 </configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/server-1/board/pid/activate-catalog"
 cookie="1470739643/d6acc6cc-a139-19a1-8002-5a45145daae4" response="yes">
<outConfig>
 <activatePIDCatalog dn="sys/chassis-1/server-1/board/pid/activate-catalog"
  adminState="triggered" pidActivationStatus="Activation Successful" status="modified">
 </activatePIDCatalog>
</outConfig>
</configConfMo>
```

Activating PID Catalog on Server Node 2

Request:

```
<configConfMo cookie="1462971767/5e320cff-9032-1290-8009-b1f9cc1a3500"
  dn="sys/chassis-1/server-2/board/pid/activate-catalog" inHierarchical="false">
 <inConfig>
  <activatePIDCatalog  adminState="trigger"
   dn="sys/chassis-1/server-2/board/pid/activate-catalog"/>
  </inConfig>
 </configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/server-2/board/pid/activate-catalog"
 cookie="1470739643/d6acc6cc-a139-19a1-8002-5a45145daae4" response="yes">
```

```
<outConfig>
 <activatePIDCatalog dn="sys/chassis-1/server-2/board/pid/activate-catalog"
  adminState="triggered" pidActivationStatus="Activation Successful" status="modified">
 </activatePIDCatalog>
</outConfig>
</configConfMo>
```

Viewing PID Catalog

Request:

```
<configResolveClass cookie="1473455878/95c48a0a-1a3c-1c1a-8003-be18652a6ca4"
inHierarchical="false" classId="pidCatalog"/>
```

Response:

```
<configResolveClass cookie="1473455878/95c48a0a-1a3c-1c1a-8003-be18652a6ca4"
 response="yes" classId="pidCatalog">
  <outConfigs>
    <pidCatalog dn="sys/chassis-1/server-1/board/pid" name="Cisco
     Product Identifiers (PID)" version="2.0(13a)09"/>
    <pidCatalog dn="sys/chassis-1/server-2/board/pid" name="Cisco
     Product Identifiers (PID)" version="2.0(13a)09"/>
  </outConfigs>
</configResolveClass>
```

Retrieving PID Catalog of CPUs

Request:

```
<configResolveClass cookie='1282522267/f91bd068-8e72-1e72-8002-7e573e8f835c'
inHierarchical='false' classId='pidCatalogCpu'/>
```

Response:

```
<configResolveClass cookie="1470739643/d6acc6cc-a139-19a1-8002-5a45145daae4"
response="yes" classId="pidCatalogCpu">
 <outConfigs>
  <pidCatalogCpu id="1" socketdesignation="CPU1" description="Intel(R)
   Xeon(R) CPU E5-2620 v4 @ 2.10GHz/85W 8C/20MB Cache/DDR4 2133MHz"
   pid="UCS-CPU-E52620E" model="Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz"
   signature="Type 0, Family 6, Model 79, Stepping 1" currentspeed="2100"
   operState="Enabled" dn="sys/chassis-1/server-1/board/pid/pid-cpu-1" >
  </pidCatalogCpu>
  <pidCatalogCpu id="2" socketdesignation="CPU2" description="Intel(R)
   Xeon(R) CPU E5-2620 v4 @ 2.10GHz/85W 8C/20MB Cache/DDR4 2133MHz"
   pid="UCS-CPU-E52620E" model="Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz"
   signature="Type 0, Family 6, Model 79, Stepping 1" currentspeed="2100"
   operState="Enabled" dn="sys/chassis-1/server-1/board/pid/pid-cpu-2" >
  </pidCatalogCpu>
  <pidCatalogCpu id="1" socketdesignation="CPU1" description="Intel(R)
   Xeon(R) CPU E5-2695 v4 @ 2.10GHz/120W 18C/45MB Cache/DDR4 2400MHz"
   pid="UCS-CPU-E52695E" model="Intel(R) Xeon(R) CPU E5-2695 v4 @ 2.10GHz"
   signature="Type 0, Family 6, Model 79, Stepping 1" currentspeed="2100"
   operState="Enabled" dn="sys/chassis-1/server-2/board/pid/pid-cpu-1" >
  </pidCatalogCpu>
  <pidCatalogCpu id="2" socketdesignation="CPU2" description="Intel(R)
   Xeon(R) CPU E5-2695 v4 @ 2.10GHz/120W 18C/45MB Cache/DDR4 2400MHz"
   pid="UCS-CPU-E52695E" model="Intel(R) Xeon(R) CPU E5-2695 v4 @ 2.10GHz"
   signature="Type 0, Family 6, Model 79, Stepping 1" currentspeed="2100"
   operState="Enabled" dn="sys/chassis-1/server-2/board/pid/pid-cpu-2" >
  </pidCatalogCpu>
 </outConfigs>
</configResolveClass>
```

Retrieving PID Catalog of DIMMs

Request:

```
<configResolveClass cookie='1282522267/f91bd068-8e72-1e72-8002-7e573e8f835c'
inHierarchical='false' classId='pidCatalogDimm'/>
```

Response:

```
<configResolveClass cookie="1470739643/d6acc6cc-a139-19a1-8002-5a45145daae4"
response="yes" classId="pidCatalogDimm">
<outConfigs>
 </pidCatalogDimm>
 <pidCatalogDimm name="DIMM_G2" description="32GB DDR4-2400-MHz RDIMM/PC4-19200/dual
rank/x4/1.2v"
  pid="UCS-MR-1X322RV-A" mfgid="0xCE00" manufacturer="Samsung" serialnumber="02EB3481"
  model="M393A4K40BB1-CRC" capacity="32768 MB" speed="2400" datawidth="64 bits"
  operability="Operable" dn="sys/chassis-1/server-2/board/pid/pid-dimm-DIMM_G2" >
 </pidCatalogDimm>
   .
   .
   .
   .
  <pidCatalogDimm name="DIMM_G2" description="32GB DDR4-2400-MHz RDIMM/PC4-19200/dual
rank/x4/1.2v"
   pid="UCS-MR-1X322RV-A" mfgid="0xCE00" manufacturer="Samsung" serialnumber="02EB33FD"
   model="M393A4K40BB1-CRC" capacity="32768 MB" speed="2400" datawidth="64 bits"
   operability="Operable" dn="sys/chassis-1/server-2/board/pid/pid-dimm-DIMM_G1" >
  </pidCatalogDimm>
  <pidCatalogDimm name="DIMM_G2" description="32GB DDR4-2400-MHz RDIMM/PC4-19200/dual
rank/x4/1.2v"
   pid="UCS-MR-1X322RV-A" mfgid="0xCE00" manufacturer="Samsung" serialnumber="02EB3481"
   model="M393A4K40BB1-CRC" capacity="32768 MB" speed="2400" datawidth="64 bits"
   operability="Operable" dn="sys/chassis-1/server-2/board/pid/pid-dimm-DIMM_G2" >
  </pidCatalogDimm>
 </outConfigs>
</configResolveClass>
```

Retrieving PID Catalog of HDDs

Request:

```
<configResolveClass cookie='1282522267/f91bd068-8e72-1e72-8002-7e573e8f835c'
inHierarchical='false' classId='pidCatalogHdd'/>
```

Response:

```
<configResolveClass cookie="1470739643/d6acc6cc-a139-19a1-8002-5a45145daae4"
response="yes" classId="pidCatalogHdd">
 <outConfigs>
  <pidCatalogHdd disk="8" controller="SBMezz1" description="4TB 6Gb SAS 7.2K RPM 3.5"
   pid="UCSC-C3X60-HD4TB" vendor="WD" model="WD4001FYYG-01SL3"
   serialnumber="WMC1F1927555"  dn="sys/chassis-1/server-1/board/pid/pid-hdd-8" >
  </pidCatalogHdd>
  <pidCatalogHdd disk="9" controller="SBMezz1" description="4TB 6Gb SAS 7.2K RPM 3.5"
   pid="UCSC-C3X60-HD4TB" vendor="WD" model="WD4001FYYG-01SL3"
   serialnumber="WMC1F1930050" dn="sys/chassis-1/server-1/board/pid/pid-hdd-9" >
  </pidCatalogHdd>
  <pidCatalogHdd disk="6" controller="SBMezz1" description="4TB 6Gb SAS 7.2K RPM 3.5"
   pid="UCSC-C3X60-HD4TB" vendor="WD" model="WD4001FYYG-01SL3"
   serialnumber="WMC1F1927672" dn="sys/chassis-1/server-2/board/pid/pid-hdd-6" >
  </pidCatalogHdd>
  <pidCatalogHdd disk="7" controller="SBMezz1" description="4TB 6Gb SAS 7.2K RPM 3.5"
   pid="UCSC-C3X60-HD4TB" vendor="WD" model="WD4001FYYG-01SL3" serialnumber="WMC1F1928331"

   dn="sys/chassis-1/server-2/board/pid/pid-hdd-7" >
  </pidCatalogHdd>
  <pidCatalogHdd disk="10" controller="SBMezz1" description="4TB 6Gb SAS 7.2K RPM 3.5"
   pid="UCSC-C3X60-HD4TB" vendor="WD" model="WD4001FYYG-01SL3" serialnumber="WMC1F1928305"
```

```
 dn="sys/chassis-1/server-2/board/pid/pid-hdd-10" >
 </pidCatalogHdd>
<pidCatalogHdd disk="11" controller="SBMezz1" description="4TB 6Gb SAS 7.2K RPM 3.5"
 pid="UCSC-C3X60-HD4TB" vendor="TOSHIBA" model="MG03SCA400" serialnumber="14P0A0E8FVU4"
  dn="sys/chassis-1/server-2/board/pid/pid-hdd-11" >
 </pidCatalogHdd>
 <pidCatalogHdd disk="12" controller="SBMezz1" description="4TB 6Gb SAS 7.2K RPM 3.5"
  pid="UCSC-C3X60-HD4TB" vendor="TOSHIBA" model="MG03SCA400" serialnumber="14P0A069FVU4"
  dn="sys/chassis-1/server-2/board/pid/pid-hdd-12" >
 </pidCatalogHdd>
 <pidCatalogHdd disk="13" controller="SBMezz1" description="4TB 6Gb SAS 7.2K RPM 3.5"
  pid="UCSC-C3X60-HD4TB" vendor="TOSHIBA" model="MG03SCA400" serialnumber="14P0A07BFVU4"
  dn="sys/chassis-1/server-2/board/pid/pid-hdd-13" >
 </pidCatalogHdd>
 <pidCatalogHdd disk="14" controller="SBMezz1" description="4TB 6Gb SAS 7.2K RPM 3.5"
  pid="UCSC-C3X60-HD4TB" vendor="TOSHIBA" model="MG03SCA400" serialnumber="14O0A06EFVU4"
  dn="sys/chassis-1/server-2/board/pid/pid-hdd-14" >
 </pidCatalogHdd>
 <pidCatalogHdd disk="28" controller="SBMezz1" description="4TB 6Gb SAS 7.2K RPM 3.5"
  pid="UCSC-C3X60-HD4TB" vendor="TOSHIBA" model="MG03SCA400" serialnumber="14O0A08MFVU4"
 dn="sys/chassis-1/server-2/board/pid/pid-hdd-28" >
 </pidCatalogHdd>
</outConfigs>
</configResolveClass>
```

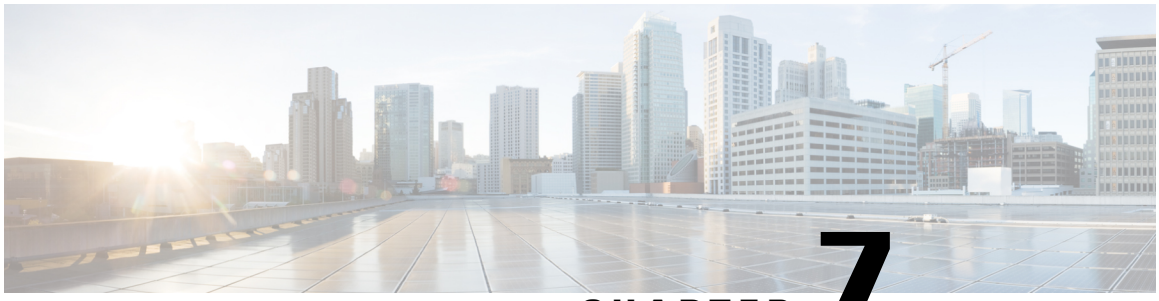
Retrieving PID Catalog of PCI Adapters

Request:

```
<configResolveClass cookie='1282522267/f91bd068-8e72-1e72-8002-7e573e8f835c'
inHierarchical='false' classId='pidCatalogPCIAdapter'/>
```

Response:

```
<configResolveClass cookie="1470739643/d6acc6cc-a139-19a1-8002-5a45145daae4"
response="yes" classId="pidCatalogPCIAdapter">
<outConfigs>
 <pidCatalogPCIAdapter slot="SBNVMe1" description="800GB 2.5in NVMe based
  PCIe SSD drive" pid="UCSC-C3K-NV8" vendor="0x1c58" device="0x0003" subvendor="0x1137"
  subdevice="0x019e" dn="sys/chassis-1/server-1/board/pid/pid-pciadapter-SBNVMe1" >
 </pidCatalogPCIAdapter>
 <pidCatalogPCIAdapter slot="SIOC1" description="Cisco UCS 40Gb Ethernet"
  pid="UCSC-C3260-SIOC" vendor="0x1137" device="0x0042" subvendor="0x1137"
  subdevice="0x0157" dn="sys/chassis-1/server-1/board/pid/pid-pciadapter-SIOC1" >
 </pidCatalogPCIAdapter>
 <pidCatalogPCIAdapter slot="SBMezz1" description="Cisco UCS C3000 RAID controller
  for M4 Server Blade with 4G RAID Cache" pid="UCS-C3K-M4RAID" vendor="0x1000"
  device="0x00ce" subvendor="0x1137" subdevice="0x0197"
 dn="sys/chassis-1/server-1/board/pid/pid-pciadapter-SBMezz1" >
 </pidCatalogPCIAdapter>
 <pidCatalogPCIAdapter slot="SIOC2" description="Cisco UCS 40Gb Ethernet" pid="N/A"
  vendor="0x1137" device="0x0042" subvendor="0x1137" subdevice="0x0157"
 dn="sys/chassis-1/server-2/board/pid/pid-pciadapter-SIOC2" >
 </pidCatalogPCIAdapter>
<pidCatalogPCIAdapter slot="SBMezz1" description="Cisco UCS C3000 RAID controller
 for M4 Server Blade with 4G RAID Cache" pid="UCSC-C3K-M4RAID" vendor="0x1000"
 device="0x00ce" subvendor="0x1137" subdevice="0x0197"
 dn="sys/chassis-1/server-2/board/pid/pid-pciadapter-SBMezz1" >
 </pidCatalogPCIAdapter>
</outConfigs>
</configResolveClass>
```



CHAPTER 7

Managing the Server

This chapter includes the following sections:

Managing Host Power

The examples in this section show how to use the Cisco IMC XML API to manage host power. Each example shows the XML API request followed by the response from Cisco IMC.

This section includes the following examples:

Powering On the Server

Request:

```
<configConfMo cookie='1418463333/5d69498a-160a-1a16-8002-91fabb1b0ff4'
        dn='sys/chassis-1/server-1'><inConfig><computeServerNode
        dn='sys/chassis-1/server-1' adminPower='up'/>
       </inConfig>
          </configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/server-1"
cookie="1418463333/5d69498a-160a-1a16-8002-91fabb1b0ff4"
 response="yes">
  <outConfig>
    <computeServerNode serverId="1" adminPower="policy" availableMemory="262144"
     model="UCSC-C3X60-SVRNB" memorySpeed="1866" name="UCS C3160" numOfCores="24"
```

```
     numOfCoresEnabled="24" numOfCpus="2" numOfThreads="48" operPower="on"
     originalUuid="1C0C4600-671D-4B53-A06E-590CD0FEBC85" presence="equipped"
     serial="FCH1821JAV4" totalMemory="262144" usrLbl=""
uuid="1C0C4600-671D-4B53-A06E-590CD0FEBC85"
     vendor="Cisco Systems Inc" dn="sys/chassis-1/server-1" status="modified"/>
  </outConfig>
</configConfMo>
```

Powering Off the Server

Request:

```
<configConfMo cookie='1418463333/5d69498a-160a-1a16-8002-91fabb1b0ff4'
 dn='sys/chassis-1/server-1'><inConfig><computeServerNode
dn='sys/chassis-1/server-1' adminPower='down'/>
</inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/server-1"
cookie="1418463333/5d69498a-160a-1a16-8002-91fabb1b0ff4"
 response="yes">
  <outConfig>
    <computeServerNode serverId="1" adminPower="policy" availableMemory="262144"
    model="UCSC-C3X60-SVRNB" memorySpeed="1866" name="UCS C3160" numOfCores="24"
    numOfCoresEnabled="24" numOfCpus="2" numOfThreads="48" operPower="off"
    originalUuid="1C0C4600-671D-4B53-A06E-590CD0FEBC85" presence="equipped"
    serial="FCH1821JAV4" totalMemory="262144" usrLbl=""
uuid="1C0C4600-671D-4B53-A06E-590CD0FEBC85"
    vendor="Cisco Systems Inc" dn="sys/chassis-1/server-1" status="modified"/>
  </outConfig>
</configConfMo>
```

Power Cycling the Server

Request:

```
<configConfMo cookie='1418463333/5d69498a-160a-1a16-8002-91fabb1b0ff4'
dn='sys/chassis-1/server-1'><inConfig><computeServerNode dn='sys/chassis-1/server-1'
adminPower='cycle-immediate'/></inConfig></configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/server-1"
cookie="1418463333/5d69498a-160a-1a16-8002-91fabb1b0ff4"
response="yes">
  <outConfig>
    <computeServerNode serverId="1" adminPower="policy" availableMemory="262144"
     model="UCSC-C3X60-SVRNB" memorySpeed="1866" name="UCS C3160" numOfCores="24"
     numOfCoresEnabled="24" numOfCpus="2" numOfThreads="48" operPower="off"
     originalUuid="1C0C4600-671D-4B53-A06E-590CD0FEBC85" presence="equipped"
serial="FCH1821JAV4"
     totalMemory="262144" usrLbl="" uuid="1C0C4600-671D-4B53-A06E-590CD0FEBC85"
     vendor="Cisco Systems Inc" dn="sys/chassis-1/server-1" status="modified"/>
  </outConfig>
</configConfMo>
```

Power Management

The examples in this section show how to use the Cisco IMC XML API to retrieve and perform power management tasks. Each example shows the XML API request followed by the response from Cisco IMC.

This section includes the following examples:

Disabling Power Capping

Request:

```
<configConfMo cookie='1472715713/d2304f05-6e3b-1b6e-8002-ddf0c322e470'
dn='sys/chassis-1/budget'>
<inConfig>
  <chassisPowerBudget dn='sys/chassis-1/budget' adminState='disabled'/>
 </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/budget"
cookie="1472715713/d2304f05-6e3b-1b6e-8002-ddf0c322e470"
response="yes">
<outConfig>
 <chassisPowerBudget dn="sys/chassis-1/budget" adminState="disabled" minPower="712"
  maxPower="1023" budget="1023" capBudget="551" powerCharStatus="Completed successfully"
  autoMinBudget="901" adminAction="no-op" status="modified" >
  </chassisPowerBudget>
 </outConfig>
</configConfMo>
```

Enabling Power Capping and Setting Chassis Budget

Request:

```
<configConfMo cookie='1472715713/d2304f05-6e3b-1b6e-8002-ddf0c322e470'
dn='sys/chassis-1/budget'>
<inConfig>
 <chassisPowerBudget dn='sys/chassis-1/budget' adminState='enabled' budget='900'/>
</inConfig></configConfMo>"
```

Response:

```
<configConfMo dn="sys/chassis-1/budget"
cookie="1472715713/d2304f05-6e3b-1b6e-8002-ddf0c322e470"
response="yes">
<outConfig>
 <chassisPowerBudget dn="sys/chassis-1/budget" adminState="enabled" minPower="712"
  maxPower="1023" budget="900" capBudget="551" powerCharStatus="Completed successfully"
  autoMinBudget="901" adminAction="no-op" status="modified" >
 </chassisPowerBudget>
</outConfig>
</configConfMo>
```

Running Power Characterization

Request:

```
<configConfMo cookie='1452562357/3e4c4620-1929-1919-800f-00f8c322e470'
 dn='sys/chassis-1/budget'>
<inConfig>
  <chassisPowerBudget dn='sys/chassis-1/budget' adminAction='start-power-char'
   status='modified'/>
 </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/budget"
cookie="1452562357/3e4c4620-1929-1919-800f-00f8c322e470"
response="yes">
<outConfig>
 <chassisPowerBudget dn="sys/chassis-1/budget" adminState="disabled" minPower="421"
  maxPower="421" budget="421" capBudget="-1" runPowCharAtBoot="yes"
  powerCharStatus="Completed successfully" adminAction="no-op" status="modified" >
 </chassisPowerBudget>
 </outConfig>
</configConfMo>
```

Resetting Power Profiles to Default

Request:

```
<configConfMo cookie='1452562357/3e4c4620-1929-1919-800f-00f8c322e470'
dn='sys/chassis-1/budget'>
 <inConfig>
 <chassisPowerBudget dn='sys/chassis-1/budget'
   adminAction='reset-power-profile-default' status='modified'/>
 </inCo nfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/budget"
cookie="1453845128/57ec2969-442a-1a44-8003-6b47145daae4"
response="yes">
<outConfig>
  <chassisPowerBudget dn="sys/chassis-1/budget" adminState="enabled" minPower="N/A"
   maxPower="N/A" budget="N/A" capBudget="N/A" runPowCharAtBoot="yes"
   powerCharStatus="Running" adminAction="no-op" status="modified">
  </chassisPowerBudget>
 </outConfig>
</configConfMo>
```

Retrieving Global Power Capping Details

Request:

```
<configResolveDn cookie='1453596884/fd2963e6-092a-1a09-8009-6b47145daae4'
inHierarchical='false' dn='sys/chassis-1/budget'/>
```

Response:

```
<configResolveDn cookie="1472719025/9c73f130-6e3b-1b6e-8003-ddf0c322e470"
 response="yes" dn="sys/chassis-1/budget">
<outConfig>
<chassisPowerBudget dn="sys/chassis-1/budget" adminState="enabled" minPower="712"
 maxPower="1023" budget="900" capBudget="435" powerCharStatus="Completed successfully"
 autoMinBudget="901" adminAction="no-op" >
 </chassisPowerBudget>
</outConfig>
</configResolveDn>
```

Enabling Custom Profiles

Request:

```
<configConfMo cookie='1452505509/a4e30f9d-0c29-190c-8008-00f8c322e470'
dn='sys/chassis-1/server-1/budget/cust-prof'>
 <inConfig>
  <customPowerProfile dn='sys/chassis-1/server-1/budget/cust-prof' profileEnabled='yes'
  allowThrottle='yes' suspendPeriod='10:30-22:30|Mo,Tu' corrAction='alert' corrTime='3'
  powerLimit='250'/>
 </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/server-1/budget/cust-prof"
 cookie="1452505509/a4e30f9d-0c29-190c-8008-00f8c322e470" response="yes">
<outConfig>
 <customPowerProfile dn="sys/chassis-1/server-1/budget/cust-prof"
  profileType="custom" profileEnabled="yes" allowThrottle="yes" powerLimit="250"
  suspendPeriod="10:30-22:30|Mo,Tu" corrAction="alert" corrTime="3" status="modified" >
 </customPowerProfile>
</outConfig>
</configConfMo>
```

Retrieving Custom Profile Details

Request:

```
<configResolveClass cookie='1452506619/f5fd0476-0c29-190c-8009-00f8c322e470'
inHierarchical='true' classId='customPowerProfile'/>
```

Response:

```
 <configResolveClass cookie="1452506619/f5fd0476-0c29-190c-8009-00f8c322e470"
response="yes" classId="customPowerProfile">
 <customPowerProfile dn="sys/chassis-1/server-1/budget/cust-prof" profileType="custom"
  profileEnabled="yes" allowThrottle="yes" powerLimit="250" suspendPeriod="10:30-22:30|Mo,Tu"

  corrAction="alert" corrTime="3" >
 </customPowerProfile>
 </outConfigs>
</configResolveClass>
```

Disabling Custom Profile

Request:

```
<configConfMo cookie='1452505509/a4e30f9d-0c29-190c-8008-00f8c322e470'
dn='sys/chassis-1/server-1/budget/cust-prof'>
<inConfig>
  <customPowerProfile dn='sys/chassis-1/server-1/budget/cust-prof' profileEnabled='no'/>
</inConfig>
</configConfMo>
```

Response:

```
 <configConfMo dn="sys/chassis-1/server-1/budget/cust-prof"
  cookie="1452505509/a4e30f9d-0c29-190c-8008-00f8c322e470" response="yes">
<outConfig>
 <customPowerProfile dn="sys/chassis-1/server-1/budget/cust-prof" profileType="custom"
  profileEnabled="no" allowThrottle="yes" powerLimit="250" suspendPeriod="10:30-22:30|Mo,Tu"

  corrAction="alert" corrTime="3" status="modified" >
 </customPowerProfile>
 </outConfig>
</configConfMo>
```

Enabling Auto-Balance Profile with Priority Type Dynamic

Request:

```
<configConfMo cookie='1453492545/55992822-f229-19f2-8003-6b47145daae4'
dn='sys/chassis-1/budget/auto-prof'>
<inConfig>
 <autoPowerProfile dn='sys/chassis-1/budget/auto-prof' profileEnabled='yes'
  priority='dynamic' allowThrottle='yes' corrAction='alert' corrTime='1'
  suspendPeriod='10:30-22:30|Mo,Tu'/>
 </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/budget/auto-prof"
 cookie="1472719802/6298eaac-6e3b-1b6e-8005-ddf0c322e470" response="yes">
 <outConfig>
 <autoPowerProfile dn="sys/chassis-1/budget/auto-prof" profileType="auto_balance"
  profileEnabled="yes" priority="dynamic" priorServerId="1" powerLimitServer1="223"
  powerLimitServer2="1163" allowThrottle="yes" suspendPeriod="10:30-22:30|Mo,Tu"
  corrAction="alert" corrTime="1" status="modified" >
 </autoPowerProfile>
 </outConfig>
</configConfMo>
```

Enabling Auto-Balance Profile with Priority Type Manual

Request:

```
<configConfMo cookie='1472719802/6298eaac-6e3b-1b6e-8005-ddf0c322e470'
dn='sys/chassis-1/budget/auto-prof'>
<inConfig>
 <autoPowerProfile dn='sys/chassis-1/budget/auto-prof' profileEnabled='yes'
  priority='manual' priorServerId='2' allowThrottle='yes' corrAction='alert'
  corrTime='1' suspendPeriod='10:30-22:30|Mo,Tu'/>
 </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/budget/auto-prof"
cookie="1472719802/6298eaac-6e3b-1b6e-8005-ddf0c322e470" response="yes">
<outConfig>
 <autoPowerProfile dn="sys/chassis-1/budget/auto-prof" profileType="auto_balance"
```

```
  profileEnabled="yes" priority="manual" priorServerId="2" powerLimitServer1="201"
  powerLimitServer2="329" allowThrottle="yes" suspendPeriod="10:30-22:30|Mo,Tu"
  corrAction="alert" corrTime="1" status="modified" >
 </autoPowerProfile>
</outConfig>
</configConfMo>
```

Disabling Auto-Balance Profile

Request:

```
<configConfMo cookie='1472719802/6298eaac-6e3b-1b6e-8005-ddf0c322e470'
 dn='sys/chassis-1/budget/auto-prof'>
<inConfig>
 <autoPowerProfile dn='sys/chassis-1/budget/auto-prof' profileEnabled='no'/>
 </inConfig>
</configConfMo>
```

Response:

```
 <configConfMo dn="sys/chassis-1/budget/auto-prof"
cookie="1472719802/6298eaac-6e3b-1b6e-8005-ddf0c322e470" response="yes">
<outConfig>
<autoPowerProfile dn="sys/chassis-1/budget/auto-prof" profileType="auto_balance"
 profileEnabled="no" priority="manual" priorServerId="2" powerLimitServer1="201"
 powerLimitServer2="329" allowThrottle="yes" suspendPeriod="10:30-22:30|Mo,Tu"
 corrAction="alert" corrTime="1" status="modified" >
 </autoPowerProfile>
 </outConfig>
</configConfMo>
```

Enabling Thermal Profile

Request:

```
<configConfMo cookie='1452550587/731da901-1729-1917-800b-00f8c322e470'
dn='sys/chassis-1/server-2/budget/thermal-prof'>
 <inConfig>
 <thermalPowerProfile dn='sys/chassis-1/server-2/budget/thermal-prof'
  profileEnabled='yes' temperature ='25'/>
 </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/server-2/budget/thermal-prof"
cookie="1472719802/6298eaac-6e3b-1b6e-8005-ddf0c322e470" response="yes">
<outConfig>
<thermalPowerProfile dn="sys/chassis-1/server-2/budget/thermal-prof"
 profileType="thermal" profileEnabled="yes" temperature="25" powerLimit="150"
 status="modified" >
 </thermalPowerProfile>
 </outConfig>
</configConfMo>
```

Disabling Thermal Profile

Request:

```
<configConfMo cookie='1452550587/731da901-1729-1917-800b-00f8c322e470'
dn='sys/chassis-1/server-2/budget/thermal-prof'>
<inConfig>
 <thermalPowerProfile dn='sys/chassis-1/server-2/budget/thermal-prof'
  profileEnabled='no' />
 </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/server-2/budget/thermal-prof"
cookie="1472719802/6298eaac-6e3b-1b6e-8005-ddf0c322e470" response="yes">
```

```
<outConfig>
 <thermalPowerProfile dn="sys/chassis-1/server-2/budget/thermal-prof"
  profileType="thermal" profileEnabled="no" temperature="40" powerLimit="150"
  status="modified" >
 </thermalPowerProfile>
</outConfig>
</configConfMo>
```

Retrieving the CPU Power Utilization for Server Nodes

Request:

```
<configResolveClass cookie='1453435371/7a33a835-e529-19e5-8002-00f8c322e470'
inHierarchical='true' classId='serverUtilization'/>
```

Response:

```
<configResolveClass cookie="1472719802/6298eaac-6e3b-1b6e-8005-ddf0c322e470"
response="yes" classId="serverUtilization">
<outConfigs>
<serverUtilization dn="sys/chassis-1/server-1/utilization" overallUtilization="0"
 cpuUtilization="0" memoryUtilization="0" ioUtilization="0" >
 </serverUtilization>
 <serverUtilization dn="sys/chassis-1/server-2/utilization" overallUtilization="96"
  cpuUtilization="100" memoryUtilization="4" ioUtilization="0" >
 </serverUtilization></outConfigs>
</configResolveClass>
```

Retrieving Power Monitoring Details of Chassis and Server Nodes

Request:

```
<configResolveClass cookie='1453489912/722ed3e2-f129-19f1-8002-6b47145daae4'
inHierarchical='true' classId='powerMonitor'/>
```

Response:

```
<configResolveClass cookie="1453489912/722ed3e2-f129-19f1-8002-6b47145daae4"
response="yes" classId="powerMonitor">
<outConfigs>
 <powerMonitor domain="Platform" current="220" minimum="90" maximum="342" average="184"
  period="0days 1:58:1" dn="sys/chassis-1/server-1/pwrmonitor-Platform" >
 </powerMonitor>
<powerMonitor domain="CPU" current="158" minimum="46" maximum="259" average="127"
 period="0days 1:58:1" dn="sys/chassis-1/server-1/pwrmonitor-CPU" >
</powerMonitor>
 <powerMonitor domain="Memory" current="18" minimum="8" maximum="33" average="15"
 period="0days 1:58:1" dn="sys/chassis-1/server-1/pwrmonitor-Memory" >
</powerMonitor><powerMonitor domain="Platform" current="132" minimum="96" maximum="366"
 average="118" period="0days 1:58:3" dn="sys/chassis-1/server-2/pwrmonitor-Platform" >
</powerMonitor><powerMonitor domain="CPU" current="74" minimum="42" maximum="251" average="62"

 period="0days 1:58:3" dn="sys/chassis-1/server-2/pwrmonitor-CPU" >
</powerMonitor>
<powerMonitor domain="Memory" current="22" minimum="16" maximum="64" average="20"
period="0days 1:58:3" dn="sys/chassis-1/server-2/pwrmonitor-Memory" >
</powerMonitor>
</outConfigs>
</configResolveClass>
```

Retrieving Chassis Power Budget Details

Request:

```
<configResolveClass cookie='1453489912/722ed3e2-f129-19f1-8002-6b47145daae4'
inHierarchical='true' classId='chassisPowerBudget'/>
```

Response:

```
<configResolveClass cookie="1453489912/722ed3e2-f129-19f1-8002-6b47145daae4"
response="yes" classId="chassisPowerBudget">
```

```
<outConfigs>
<chassisPowerBudget dn="sys/chassis-1/budget" adminState="enabled" minPower="888"
 maxPower="1361" budget="1200" capBudget="740" runPowCharAtBoot="yes"
 powerCharStatus="Completed successfully" adminAction="no-op" >
<autoPowerProfile rn="auto-prof" profileType="auto_balance" profileEnabled="no"
 priority="manual" priorServerId="1" powerLimitServer1="499" powerLimitServer2="415"
 allowThrottle="no" suspendPeriod="" corrAction="none" corrTime="1" >
</autoPowerProfile>
</chassisPowerBudget></outConfigs>
</configResolveClass>
```

Managing Boot Order

The examples in this section show how to use the Cisco IMC XML API to retrieve and configure the boot order. Each example shows the XML API request followed by the response from Cisco IMC.

This section includes the following examples:

Retrieving Precision Boot Order Information

Request:

```
<configResolveClass cookie="1256524707/a7d534e9-cd76-16cd-8014-aaec921b0ff4"
        inHierarchical="true" classId="lsbootDevPrecision"/>
```

Response:

```
<configResolveClass cookie="1256524707/a7d534e9-cd76-16cd-8014-aaec921b0ff4" response="yes"

classId="lsbootDevPrecision">
  <outConfigs>
    <lsbootDevPrecision dn="sys/chassis-1/server-1/boot-precision" name="boot-precision"
        purpose="operational" rebootOnUpdate="no" reapply="no" configuredBootMoe="None"
      lastConfiguredBootOrderSource="CIMC"/>
    <lsbootDevPrecision dn="sys/chassis-1/server-2/boot-precision" name="boot-precision"
     purpose="operational" rebootOnUpdate="no" reapply="no" configuredBootMode="Uefi"
      lastConfiguredBootOrderSource="CIMC">
      <lsbootVMedia name="VMEDIA-KVMFDD" type="VMEDIA" subtype="kvm-mapped-fdd"
      access="read-write-drive" order="1" state="Disabled" rn="vm-VMEDIA-KVMFDD"/>
      <lsbootVMedia name="VMEDIA-1" type="VMEDIA" subtype="kvm-mapped-dvd"
       access="read-only-local" order="8" state="Disabled" rn="vm-VMEDIA-1"/>
        .
        .
        .
     <lsbootPchStorage name="TEST-PCH" type="PCHSTORAGE" lun="3" order="12" state="Disabled"
 rn="pchstorage-TEST-PCH"/>
      <lsbootUefiShell name="EFI0-1" type="UEFISHELL" order="13" state="Disabled"
rn="uefishell-EFI0-1"/>
```

```
    </lsbootDevPrecision>
  </outConfigs>
</configResolveClass>
```

Retrieving Boot Order Table Information (Server 1)

Request:

```
<configResolveDn cookie="1256527817/b46e308c-ce76-16ce-8003-aaec921b0ff4"
dn='sys/chassis-1/server-1/bios/bdgep' inHierarchical="true"/>
```

Response:

```
<configResolveDn cookie="1256527817/b46e308c-ce76-16ce-8003-aaec921b0ff4"
response="yes" dn="sys/chassis-1/server-1/bios/bdgep">
  <outConfig>
    <biosBOT dn="sys/chassis-1/server-1/bios/bdgep">
      <biosBootMode rn="boot-mode" actualBootMode="Legacy"/>
      <biosBootDevPrecision descr="Cisco vKVM-Mapped vDVD1.22"
       order="1" name="CDROM" type="VMEDIA" subtype="kvm-mapped-dvd" rn="bdvp-1"/>
      <biosBootDevPrecision descr="Cisco CIMC-Mapped vDVD1.22"
       order="2" name="CDROM" type="VMEDIA" subtype="cimc-mapped-dvd" rn="bdvp-2"/>
      <biosBootDevPrecision descr="UEFI: Built-in EFI Shell "
       order="3" name="NonPolicyTarget" type="EFI" rn="bdvp-3"/>
      <biosBootDevPrecision descr="(Bus 04 Dev 00)PCI RAID Adapter"
       order="4" name="NonPolicyTarget" type="HDD" slot="M" rn="bdvp-4"/>
      <biosBootDevPrecision descr="Cisco vKVM-Mapped vHDD1.22"
      order="5" name="NonPolicyTarget" type="VMEDIA" subtype="kvm-mapped-hdd" rn="bdvp-5"/>

      <biosBootDevPrecision descr="Cisco vKVM-Mapped vFDD1.22"
      order="6" name="NonPolicyTarget" type="VMEDIA" subtype="kvm-mapped-fdd" rn="bdvp-6"/>

      <biosBootDevPrecision descr="Cisco CIMC-Mapped vHDD1.22"
      order="7" name="NonPolicyTarget" type="VMEDIA" subtype="cimc-mapped-hdd" rn="bdvp-7"/>

    </biosBOT>
  </outConfig>
</configResolveDn>
```

Retrieving Boot Order Table Information (Server 2)

Request:

```
<configResolveDn cookie="1256527817/b46e308c-ce76-16ce-8003-aaec921b0ff4"
dn='sys/chassis-1/server-2/bios/bdgep' inHierarchical="true"/>
```

Response:

```
<configResolveDn cookie="1256527817/b46e308c-ce76-16ce-8003-aaec921b0ff4"
response="yes" dn="sys/chassis-1/server-2/bios/bdgep">
  <outConfig>
    <biosBOT dn="sys/chassis-1/server-2/bios/bdgep">
      <biosBootMode rn="boot-mode" actualBootMode="Uefi"/>
      <biosBootDevPrecision descr="UEFI: Cisco vKVM-Mapped vDVD1.22"
      order="1" name="CDROM" type="EFI" rn="bdvp-1"/>
      <biosBootDevPrecision descr="UEFI: Built-in EFI Shell "
      order="2" name="NonPolicyTarget" type="EFI" rn="bdvp-2"/>
      <biosBootDevPrecision descr="Cisco vKVM-Mapped vDVD1.22"
      order="3" name="NonPolicyTarget" type="VMEDIA"
      subtype="kvm-mapped-dvd" rn="bdvp-3"/>
      <biosBootDevPrecision descr="Cisco vKVM-Mapped vHDD1.22"
      order="4" name="NonPolicyTarget" type="VMEDIA"
      subtype="kvm-mapped-hdd" rn="bdvp-4"/>
      <biosBootDevPrecision descr="Cisco vKVM-Mapped vFDD1.22"
      order="5" name="NonPolicyTarget" type="VMEDIA"
      subtype="kvm-mapped-fdd" rn="bdvp-5"/>
      <biosBootDevPrecision descr="Cisco CIMC-Mapped vDVD1.22"
      order="6" name="NonPolicyTarget" type="VMEDIA"
      subtype="cimc-mapped-dvd" rn="bdvp-6"/>
```

```
      <biosBootDevPrecision descr="Cisco CIMC-Mapped vHDD1.22"
      order="7" name="NonPolicyTarget" type="VMEDIA"
      subtype="cimc-mapped-hdd" rn="bdvp-7"/>
    </biosBOT>
  </outConfig>
</configResolveDn>
```

Retrieving Actual Boot Order Information (Server 1 and Server 2)

Request:

```
<configResolveClass cookie="1256525715/b603f647-ce76-16ce-8015-aaec921b0ff4"
inHierarchical="true" classId="biosBootDevPrecision"/>
```

Response:

```
<configResolveClass cookie="1256525715/b603f647-ce76-16ce-8015-aaec921b0ff4" res
ponse="yes" classId="biosBootDevPrecision">
  <outConfigs>
    <biosBootDevPrecision descr="Cisco vKVM-Mapped vDVD1.22" order="1"
    name="CDROM" type="VMEDIA" subtype="kvm-mapped-dvd"
    dn="sys/chassis-1/server-1/bios/bdgep/bdvp-1"/> <biosBootDevPrecision
    descr="Cisco CIMC-Mapped vDVD1.22" order="2"
    name="CDROM" type="VMEDIA" subtype="cimc-mapped-dvd"
    dn="sys/chassis-1/server-1/bios/bdgep/bdvp-2"/>
    <biosBootDevPrecision descr="UEFI: Built-in EFI Shell " order="3"
    name="NonPolicyTarget" type="EFI" dn="sys/chassis-1/server-1/bios/bdgep/bdvp-3"/>
    <biosBootDevPrecision descr="(Bus 04 Dev 00)PCI RAID Adapter" order="4"
   name="NonPolicyTarget" type="HDD" slot="M" dn="sys/chassis-1/server-1/bios/bdgep/bdvp-4"/>

    <biosBootDevPrecision descr="Cisco vKVM-Mapped vHDD1.22" order="5"
    name="NonPolicyTarget" type="VMEDIA" subtype="kvm-mapped-hdd"
    dn="sys/chassis-1/server-1/bios/bdgep/bdvp-5"/>
    <biosBootDevPrecision descr="Cisco vKVM-Mapped vFDD1.22" order="6"
    name="NonPolicyTarget" type="VMEDIA" subtype="kvm-mapped-fdd"
    dn="sys/chassis-1/server-1/bios/bdgep/bdvp-6"/>
    <biosBootDevPrecision descr="Cisco CIMC-Mapped vHDD1.22" order="7"
    name="NonPolicyTarget" type="VMEDIA" subtype="cimc-mapped-hdd"
    dn="sys/chassis-1/server-1/bios/bdgep/bdvp-7"/>

    <biosBootDevPrecision descr="UEFI: Cisco vKVM-Mapped vDVD1.22" order="1"
    name="CDROM" type="EFI" dn="sys/chassis-1/server-2/bios/bdgep/bdvp-1"/>
    <biosBootDevPrecision descr="UEFI: Built-in EFI Shell " order="2"
    name="NonPolicyTarget" type="EFI" dn="sys/chassis-1/server-2/bios/bdgep/bdvp-2"/>
    <biosBootDevPrecision descr="Cisco vKVM-Mapped vDVD1.22" order="3"
    name="NonPolicyTarget" type="VMEDIA" subtype="kvm-mapped-dvd"
    dn="sys/chassis-1/server-2/bios/bdgep/bdvp-3"/>
    <biosBootDevPrecision descr="Cisco vKVM-Mapped vHDD1.22" order="4"
    name="NonPolicyTarget" type="VMEDIA" subtype="kvm-mapped-hdd"
    dn="sys/chassis-1/server-2/bios/bdgep/bdvp-4"/>
    <biosBootDevPrecision descr="Cisco vKVM-Mapped vFDD1.22" order="5"
    name="NonPolicyTarget" type="VMEDIA" subtype="kvm-mapped-fdd"
    dn="sys/chassis-1/server-2/bios/bdgep/bdvp-5"/>
    <biosBootDevPrecision descr="Cisco CIMC-Mapped vDVD1.22" order="6"
    name="NonPolicyTarget" type="VMEDIA" subtype="cimc-mapped-dvd"
    dn="sys/chassis-1/server-2/bios/bdgep/bdvp-6"/>
    <biosBootDevPrecision descr="Cisco CIMC-Mapped vHDD1.22"
    order="7" name="NonPolicyTarget" type="VMEDIA" subtype="cimc-mapped-hdd"
    dn="sys/chassis-1/server-2/bios/bdgep/bdvp-7"/>
  </outConfigs>
</configResolveClass>
```

Retrieving Actual Boot Order Information using DN

Request:

```
<configResolveDn cookie="1256525715/b603f647-ce76-16ce-8015-aaec921b0ff4"
dn='sys/chassis-1/server-2/bios/bdgep/bdvp-5' inHierarchical="false"/>
```

Response:

```
<configResolveDn cookie="1256525715/b603f647-ce76-16ce-8015-aaec921b0ff4"
response="yes" dn="sys/chassis-1/server-2/bios/bdgep/bdvp-5">
  <outConfig>
    <biosBootDevPrecision descr="Cisco vKVM-Mapped vFDD1.22" order="5"
    name="NonPolicyTarget" type="VMEDIA" subtype="kvm-mapped-fdd"
    dn="sys/chassis-1/server-2/bios/bdgep/bdvp-5"/>
  </outConfig>
</configResolveDn>
```

Retrieving BIOS Boot Mode Information using DN

Request:

```
<configResolveDn cookie="1256526667/3bbac6ee-cd76-16cd-8002-aaec921b0ff4"
dn='sys/chassis-1/server-1/bios/bdgep/boot-mode' inHierarchical="true"/>
```

Response:

```
<configResolveDn cookie="1256526667/3bbac6ee-cd76-16cd-8002-aaec921b0ff4"
  response="yes" dn="sys/chassis-1/server-1/bios/bdgep/boot-mode">
  <outConfig>
    <biosBootMode dn="sys/chassis-1/server-1/bios/bdgep/boot-mode"
     actualBootMode="Legacy"/>
  </outConfig>
</configResolveDn>
```

Configuring Legacy to UEFI Boot Mode

Request:

```
<configConfMo cookie='1256586379/e30f586c-dc76-16dc-8008-aaec921b0ff4'
dn='sys/chassis-1/server-2/boot-precision'><inConfig><lsbootDevPrecision
configuredBootMode='Uefi'/></inConfig></configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/server-2/boot-precision"
cookie="1256586379/e30f586c-dc76-16dc-8008-aaec921b0ff4" response="yes">
  <outConfig>
    <lsbootDevPrecision dn="sys/chassis-1/server-2/boot-precision"
name="boot-precision" purpose="operational" rebootOnUpdate="no"
reapply="no" configuredBootMode="Uefi" lastConfiguredBootOrderSource="CIMC"
status="modified"/>
  </outConfig>
</configConfMo>
```

Retrieving UEFI Secure Boot Information

Request:

```
<configResolveClass cookie="1256580961/3d850c6c-db76-16db-8006-aaec921b0ff4"
inHierarchical="false" classId="lsbootBootSecurity"/>
```

Response:

```
<configResolveClass cookie="1256580961/3d850c6c-db76-16db-8006-aaec921b0ff4" response="yes"

   classId="lsbootBootSecurity">
  <outConfigs>
    <lsbootBootSecurity dn="sys/chassis-1/server-1/boot-policy/boot-security"
secureBoot="disabled"/>
```

```
    <lsbootBootSecurity dn="sys/chassis-1/server-2/boot-policy/boot-security"
secureBoot="disabled"/>
  </outConfigs>
</configResolveClass>
```

Enabling UEFI Secure Boot (Server 2)

Request:

```
<configConfMo cookie='1256583805/6a769571-db76-16db-8007-aaec921b0ff4'
dn='sys/chassis-1/server-2/boot-policy/boot-security'><inConfig><lsbootBootSecurity
secureBoot='enable'/></inConfig></configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/server-2/boot-policy/boot-security"
cookie="1256583805/6a769571-db76-16db-8007-aaec921b0ff4" response="yes">
  <outConfig>
    <lsbootBootSecurity dn="sys/chassis-1/server-2/boot-policy/boot-security"
secureBoot="enabled" status="modified"/>
  </outConfig>
</configConfMo>
```

Managing Boot Devices

The examples in this section show how to use the Cisco IMC XML API to manage a boot device. Each example shows the XML API request followed by the response from Cisco IMC.

This section includes the following examples:

- Creating a Boot Device, on page 67
- Modifying a Boot Device, on page 68
- Deleting a Boot Device, on page 69

Creating a Boot Device

Request:

```
<configConfMo cookie="1418657182/795cfede-420a-1a42-800e-91fabb1b0ff4"
dn="sys/chassis-1/server-1/boot-precision" inHierarchical="true">
  <inConfig>
    <lsbootDevPrecision dn="sys/chassis-1/server-1/boot-precision"
    rebootOnUpdate="no" reapply="yes" >
     <lsbootVMedia name="vmedia-c3260-test" type="VMEDIA"
     order="1" state="Disabled" rn="vm-vmedia-c3260-test" />
     <lsbootPxe name="pxe-c3260-test" type="PXE" slot="1" port="1"
     order="2" state="Disabled" rn="pxe-pxe-c3260-test" />
     <lsbootHdd name="hdd-c3260-test" type="LOCALHDD" order="3"
     state="Disabled" rn="hdd-hdd-c3260-test" />
     <lsbootSan name="san-c3260-test" type="SAN" slot="4" order="4"
     state="Disabled" rn="san-san-c3260-test" />
     <lsbootIscsi name="iscsi-c3260-test" type="ISCSI" slot="7"
     order="5" state="Disabled" rn="iscsi-iscsi-c3260-test" />
    </lsbootDevPrecision>
  </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/server-1/boot-precision"
cookie="1418657182/795cfede-420a-1a42-800e-91fabb1b0ff4" response="yes">
  <outConfig>
```

```
    <lsbootDevPrecision dn="sys/chassis-1/server-1/boot-precision"
     name="boot-precision" purpose="operational" rebootOnUpdate="no"
     reapply="no" configuredBootMode="Uefi"
     lastConfiguredBootOrderSource="CIMC" status="modified">
      <lsbootVMedia name="vmedia-c3260-test" type="VMEDIA" order="1"
      state="Disabled" rn="vm-vmedia-c3260-test" status="modified"/>
      .
      .
      .
      <lsbootPxe name="pxe-c3260-test" type="PXE" slot="1" port="1"
      order="2" state="Disabled" rn="pxe-pxe-c3260-test" status="modified"/>
      .
      .
      .
      <lsbootHdd name="hdd-c3260-test" type="LOCALHDD" order="3"
      state="Disabled" rn="hdd-hdd-c3260-test" status="modified"/>
      .
      .
      .
      <lsbootSan name="san-c3260-test" type="SAN" slot="4" order="4"
      state="Disabled" rn="san-san-c3260-test" status="modified"/>
      <lsbootIscsi name="iscsi-c3260-test" type="ISCSI" slot="7"
      order="5" state="Disabled" rn="iscsi-iscsi-c3260-test" status="modified"/>
      .
      .
      .
    </lsbootDevPrecision>
  </outConfig>
</configConfMo>
```

Modifying a Boot Device

Request:

```
<configConfMo cookie="1418661362/99fd4a5a-440a-1a44-800f-91fabb1b0ff4"
dn="sys/chassis-1/server-1/boot-precision" inHierarchical="true">
  <inConfig>
      <lsbootDevPrecision dn="sys/chassis-1/server-1/boot-precision"
       rebootOnUpdate="yes" status="modified">
      <lsbootVMedia name="vmedia-c3260-test" order="2" />
      <lsbootIscsi name="iscsi-c3260-test" order="4" />
      </lsbootDevPrecision>
  </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/server-1/boot-precision"
cookie="1418657182/795cfede-420a-1a42-800e-91fabb1b0ff4"
response="yes">
  <outConfig>
    <lsbootDevPrecision dn="sys/chassis-1/server-1/boot-precision"
    name="boot-precision" purpose="operational" rebootOnUpdate="no"
    reapply="no" configuredBootMode="Uefi" lastConfiguredBootOrderSource="CIMC"
    status="modified">
      <lsbootVMedia name="vmedia-c3260-test" type="VMEDIA" order="2"
      state="Disabled" rn="vm-vmedia-c3260-test" status="modified"/>
      .
      .
      .
      <lsbootPxe name="pxe-c3260-test" type="PXE" slot="1" port="1"
      order="1" state="Disabled" rn="pxe-pxe-c3260-test" status="modified"/>
      .
      .
      .
      <lsbootHdd name="hdd-c3260-test" type="LOCALHDD" order="3"
      state="Disabled" rn="hdd-hdd-c3260-test" status="modified"/>
      .
      .
      .
      <lsbootSan name="san-c3260-test" type="SAN" slot="4" order="5s"
```

```
      state="Disabled" rn="san-san-c3260-test" status="modified"/>
      <lsbootIscsi name="iscsi-c3260-test" type="ISCSI" slot="7" order="4"
      state="Disabled" rn="iscsi-iscsi-c3260-test" status="modified"/>
      .
      .
      .
    </lsbootDevPrecision>
  </outConfig>
</configConfMo>
```

Deleting a Boot Device

Request:

```
<configConfMo cookie="1418635679/ac17fae8-3d0a-1a3d-800a-91fabb1b0ff4"
dn="sys/chassis-1/server-1/boot-precision" inHierarchical="true">
  <inConfig>
    <lsbootDevPrecision dn="sys/chassis-1/server-1/boot-precision"
    rebootOnUpdate="yes" status="modified">
      <lsbootVMedia name="NIIODCIMCDVD" type="VMEDIA"
     subtype="cimc-mapped-dvd" access="read-only-remote" order="3" status="removed"/>
    </lsbootDevPrecision>
  </inConfig>
</configConfMo>
```

DIMM Blacklisting

The examples in this section show how to use the Cisco IMC XML API to retrieve and enable DIMM Blacklisting. Each example shows the XML API request followed by the response from Cisco IMC.

This section includes the following examples:

- Retrieving DIMM Blacklisting Details, on page 69
- Retrieving DIMM Blacklisting Details using DN, on page 70
- Enabling DIMM Blacklisting, on page 70
- Disabling DIMM Blacklisting, on page 70

Retrieving DIMM Blacklisting Details

Request:

```
<configResolveClass cookie="1256404558/fa002591-b176-16b1-8007-aaec921b0ff4"
inHierarchical="false"  classId="memoryArray"/>
```

Response:

```
<configResolveClass cookie="1256404558/fa002591-b176-16b1-8007-aaec921b0ff4"
response="yes" classId="memoryArray">
  <outConfigs>
    <memoryArray dn="sys/chassis-1/server-1/board/memarray-1" currCapacity="131072" id="1"

    maxDevices="16" populated="16" presence="equipped" overallDIMMStatus="green"
    dimmBlackList="enabled" redundantMemory="0" memoryRASPossible="Independent Mirroring
Lockstep "
    memoryConfiguration="Independent " failedMemory="0"
    ignoredMemory="0" numOfIgnoredDimms="0" numOfFailedDimms="0"/>
    <memoryArray dn="sys/chassis-1/server-2/board/memarray-1"
   currCapacity="262144" id="1" maxDevices="16" populated="16"
   presence="equipped" overallDIMMStatus="red" dimmBlackList="enabled"
   redundantMemory="0" memoryRASPossible="Independent "
   memoryConfiguration="Independent " failedMemory="98304"
   ignoredMemory="0" numOfIgnoredDimms="0" numOfFailedDimms="6"/>
```

```
  </outConfigs>
</configResolveClass>
```

Retrieving DIMM Blacklisting Details using DN

Request:

```
<configResolveDn cookie="1256404558/fa002591-b176-16b1-8007-aaec921b0ff4"
dn='sys/chassis-1/server-1/board/memarray-1' inHierarchical="false"/>
```

Response:

```
<configResolveDn cookie="1256404558/fa002591-b176-16b1-8007-aaec921b0ff4"
response="yes" dn="sys/chassis-1/server-1/board/memarray-1">
  <outConfig>
    <memoryArray dn="sys/chassis-1/server-1/board/memarray-1" currCapacity="131072" id="1"

     maxDevices="16" populated="16" presence="equipped" overallDIMMStatus="green"
     dimmBlackList="enabled" redundantMemory="0"
     memoryRASPossible="Independent Mirroring Lockstep "
     memoryConfiguration="Independent " failedMemory="0"
     ignoredMemory="0" numOfIgnoredDimms="0" numOfFailedDimms="0"/>
  </outConfig>
</configResolveDn>
```

Enabling DIMM Blacklisting

Request:

```
<configConfMo cookie='1256416392/bc0f62a9-b476-16b4-8009-aaec921b0ff4'
dn='sys/chassis-1/server-2/board/memarray-1'><inConfig><memoryArray
dimmBlackList='enable'/></inConfig></configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/server-2/board/memarray-1"
 cookie="1256418045/eb9986c2-b476-16b4-8002-aaec921b0ff4" response="yes">
  <outConfig>
    <memoryArray dn="sys/chassis-1/server-2/board/memarray-1"
    currCapacity="262144" id="1" maxDevices="16" populated="16"
     presence="equipped" overallDIMMStatus="green" dimmBlackList="enabled"
    redundantMemory="0" memoryRASPossible="Independent "
    memoryConfiguration="Independent " failedMemory="98304"
    ignoredMemory="0" numOfIgnoredDimms="0" numOfFailedDimms="6"
    status="modified"/>
  </outConfig>
</configConfMo>
```

Disabling DIMM Blacklisting

Request:

```
<configConfMo cookie='1256416392/bc0f62a9-b476-16b4-8009-aaec921b0ff4'
dn='sys/chassis-1/server-2/board/memarray-1'><inConfig><memoryArray
dimmBlackList='disable'/></inConfig></configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/server-2/board/memarray-1"
 cookie="1256418045/eb9986c2-b476-16b4-8002-aaec921b0ff4" response="yes">
  <outConfig>
    <memoryArray dn="sys/chassis-1/server-2/board/memarray-1"
    currCapacity="262144" id="1" maxDevices="16" populated="16"
     presence="equipped" overallDIMMStatus="green" dimmBlackList="disabled"
    redundantMemory="0" memoryRASPossible="Independent "
    memoryConfiguration="Independent " failedMemory="98304"
    ignoredMemory="0" numOfIgnoredDimms="0" numOfFailedDimms="6"
```

```
    status="modified"/>
  </outConfig>
</configConfMo>
```

BIOS Settings

The examples in this section show how to use the Cisco IMC XML API to manage BIOS settings. Each example shows the XML API request followed by the response from Cisco IMC.

This section includes the following examples:

- Entering BIOS Setup, on page 71
- Retrieving BIOS Token Values, on page 71
- Configuring BIOS Parameters, on page 71

Entering BIOS Setup

Request:

```
<configConfMo cookie='1418468130/1d579ff4-160a-1a16-8004-91fabb1b0ff4'
dn='sys/chassis-1/server-1/bios'><inConfig><biosUnit
adminAction='enter-bios-setup'/></inConfig></configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/server-1/bios"
 cookie="1418468130/1d579ff4-160a-1a16-8004-91fabb1b0ff4" response="yes">
  <outConfig>
    <biosUnit dn="sys/chassis-1/server-1/bios" model="UCSC-C3X60-BASE"
    vendor="Cisco Systems Inc" adminAction="no-op" status="modified"/>
  </outConfig>
</configConfMo>
```

Retrieving BIOS Token Values

Request:

```
<configResolveClass cookie='1418722466/fced6450-520a-1a52-8014-91fabb1b0ff4'
inHierarchical='false' classId='biosVfLegacyUSBSupport' />
```

Response:

```
<configResolveClass cookie="1418722466/fced6450-520a-1a52-8014-91fabb1b0ff4"
 response="yes" classId="biosVfLegacyUSBSupport">
  <outConfigs>
    <biosVfLegacyUSBSupport dn="sys/chassis-1/server-1/bios/bios-settings/LegacyUSB-Support"

    vpLegacyUSBSupport="enabled"/>
    <biosVfLegacyUSBSupport dn="sys/chassis-1/server-1/bios/bios-defaults/LegacyUSB-Support"

    vpLegacyUSBSupport="enabled"/>
  </outConfigs>
</configResolveClass>
```

Configuring BIOS Parameters

Request:

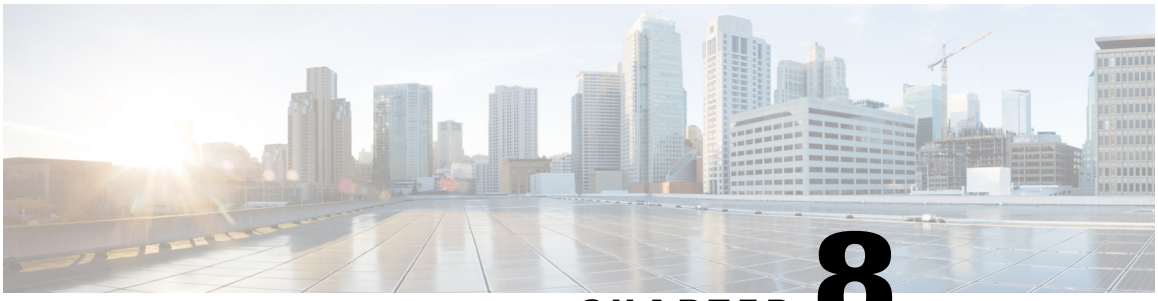
```
<configConfMo cookie="1418723449/52bb08e3-520a-1a52-8015-91fabb1b0ff4"
inHierarchical="true" dn="sys/chassis-1/server-1/bios/bios-settings">
        <inConfig>
```

```
        <biosSettings>
        <biosVfLegacyUSBSupport
dn="sys/chassis-1/server-1/bios/bios-settings/LegacyUSB-Support"
        vpLegacyUSBSupport="disabled" />
        </biosSettings>
        </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/server-1/bios/bios-settings" cookie="1418723449/
52bb08e3-520a-1a52-8015-91fabb1b0ff4" response="yes">
  <outConfig>
    <biosSettings dn="sys/chassis-1/server-1/bios/bios-settings" status="modified">
      <biosVfIntelVTForDirectedIO rn="Intel-VT-for-directed-IO" vpIntelVTDATSSup
      port="enabled" vpIntelVTDCoherencySupport="disabled"
      vpIntelVTForDirectedIO="enabled" status="modified"/>
      .
      .
      .

      <biosVfLegacyUSBSupport rn="LegacyUSB-Support" vpLegacyUSBSupport="disabled"
status="modified"/>
       .
       .
       .
    </biosSettings>
  </outConfig>
</configConfMo>
```



CHAPTER 8

Managing Remote Presence

This chapter includes the following sections:

Managing Cisco IMC-Mapped vMedia Volume

The examples in this section show how to use the Cisco IMC XML API to manage Cisco IMC-Mapped vMedia volumes. Each example shows the XML API request followed by the response from Cisco IMC.

This section includes the following examples:

Configuring a Cisco IMC-Mapped vMedia Volume

Request:

```
<configConfMo cookie='1418471980/706e0794-180a-1a18-8007-91fabb1b0ff4'
dn='sys/chassis-1/server-1/svc-ext/vmedia-svc/vmmap-HUU' inHierarchical='false'>
<inConfig>
<commVMediaMap dn='sys/chassis-1/server-1/svc-ext/vmedia-svc/vmmap-HUU'
  map='www' mountOptions='username=jchambers,password=john123'
  remoteFile='ucs-c3x60-huu-2.0.2d.iso'
  remoteShare='http://171.70.126.106/firmware-containers/colusa1/ep_mr_colusa2/'
  status='created' volumeName='HUU' />
</inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/server-1/svc-ext/vmedia-svc/vmmap-HUU"
 cookie="1418471980/706e0794-180a-1a18-8007-91fabb1b0ff4" response="yes">
<outConfig>
 <commVMediaMap volumeName="HUU" map="www"
  remoteShare="http://171.70.126.106/firmware-containers/colusa1/ep_mr_colusa1/"
  remoteFile="ucs-c3160-huu-2.0.2d.iso"
  mountOptions="noauto,username=jchambers,password=*******"
  mappingStatus="In Progress" password=""
```

```
  dn="sys/chassis-1/server-1/svc-ext/vmedia-svc/vmmap-HUU" status="created"/>
  </outConfig>
</configConfMo>
```

Retrieving and Verifying Cisco IMC-Mapped vMedia Volume Configuration

Request:

```
<configResolveClass cookie="1418471980/706e0794-180a-1a18-8007-91fabb1b0ff4"
inHierarchical="false" classId="commVMediaMap"/>
```

Response:

```
<configResolveClass cookie="1418471980/706e0794-180a-1a18-8007-91fabb1b0ff4"
response="yes" classId="commVMediaMap">
  <outConfigs>
    <commVMediaMap volumeName="HUU" map="www"
    remoteShare="http://171.70.126.106/firmware-containers/colusa1/ep_mr_colusa2/"
    remoteFile="ucs-c3x60-huu-2.0.2d.iso"
    mountOptions="noauto,username=jchambers,password=*******"
    mappingStatus="OK" password=""
    dn="sys/chassis-1/server-1/svc-ext/vmedia-svc/vmmap-HUU"/>
  </outConfigs>
</configResolveClass>
```

Managing KVM Console

The examples in this section show how to use the Cisco IMC XML API to manage the KVM console. Each example shows the XML API request followed by the response from Cisco IMC.

This section includes the following examples:

- Retrieving Authentication Tokens, on page 74
- Launching KVM Console, on page 74

Retrieving Authentication Tokens

Request:

```
<aaaGetComputeAuthTokens cookie="1256423862/2cdc2ed8-b676-16b6-8005-aaec921b0ff4"/>
```

Response:

```
<aaaGetComputeAuthTokens cookie="1256423862/2cdc2ed8-b676-16b6-8005-aaec921b0ff4"
  outTokens="1804289383,846930886" response="yes"> </aaaGetComputeAuthTokens>
```

Launching KVM Console

KVM Server 1

```
javaws
"https://10.104.255.243/kvm.jnlp?cimcAddr=10.104.255.181&cimcName=chuckrobbins&tkn1=1804289383&tkn2=846930886"
```

KVM Server 2

```
javaws
"https://10.104.255.243/kvm.jnlp?cimcAddr=10.104.255.183&cimcName=jchambers&tkn1=1804289383&tkn2=846930886"
```

Managing Serial Over LAN

The examples in this section show how to use the Cisco IMC XML API to manage retrieving and configuring serial over LAN. Each example shows the XML API request followed by the response from Cisco IMC.

This section includes the following examples:

Retrieving Serial Over LAN Details

Request:

```
<configResolveClass cookie="1256429224/24b9387f-b776-16b7-8009-aaec921b0ff4"
inHierarchical="false" classId="solIf"/>
```

Response:

```
<configResolveClass cookie="1256429224/24b9387f-b776-16b7-8009-aaec921b0ff4"
response="yes" classId="solIf">
  <outConfigs>
    <solIf dn="sys/chassis-1/server-1/sol-if" adminState="enable" name="SoLInterface"
     speed="115200" comport="com0"/>
    <solIf dn="sys/chassis-1/server-2/sol-if" adminState="enable" name="SoLInterface"
     speed="115200" comport="com0"/>
  </outConfigs>
</configResolveClass>
```

Retrieving Serial Over LAN Details for Server 1 Using DN

Request:

```
<configResolveDn cookie="1256429224/24b9387f-b776-16b7-8009-aaec921b0ff4"
dn="sys/chassis-1/server-1/sol-if"  inHierarchical="false"/>
```

Response:

```
<configResolveDn cookie="1256429224/24b9387f-b776-16b7-8009-aaec921b0ff4"
response="yes" dn="sys/chassis-1/server-1/sol-if">
  <outConfig>
    <solIf dn="sys/chassis-1/server-1/sol-if" adminState="enable" name="SoLInterface"
    speed="115200" comport="com0"/>
  </outConfig>
</configResolveDn>
```

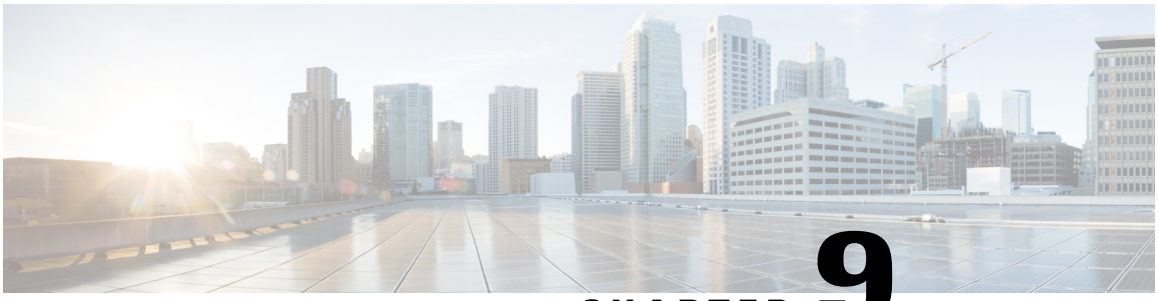
Configuring Serial Over LAN for Server 1

Request:

```
<configConfMo cookie='1256429224/24b9387f-b776-16b7-8009-aaec921b0ff4'
dn='sys/chassis-1/server-2/sol-if'>
<inConfig>
  <solIf dn='sys/chassis-1/server-2/sol-if' adminState='disable'
   speed='38400'  comport='com0'/>
 </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/server-2/sol-if"
cookie="1256429224/24b9387f-b776-16b7-8009-aaec921b0ff4" response="yes">
 <outConfig>
   <solIf dn="sys/chassis-1/server-2/sol-if" adminState="disable"
    name="SoLInterface" speed="38400" comport="com0" status="modified"/>
 </outConfig>
</configConfMo>
```



CHAPTER 9

Configuring Network-Related Settings

This chapter includes the following sections:

Examples of Network Related Tasks

The examples in this section show how to use the Cisco IMC XML API to configure network related tasks. Each example shows the XML API request followed by the response from Cisco IMC.

This section includes the following examples:

Configuring BMC Static IPv6

Request

```
<configConfMo cookie="0956648424/5a81f826-1166-1611-8003-82f92a02b884"
 dn="sys/chassis-1/server-1/mgmt/if-1">
<inConfig>
    <mgmtIf dn="sys/chassis-1/server-1/mgmt/if-1" v6IPAddr="2100:200::125" />
</inConfig>
</configConfMo>
```

Response

```
<configConfMo cookie="0956648424/5a81f826-1166-1611-8003-82f92a02b884"
 response="yes" dn="sys/chassis-1/server-1/mgmt/if-1">
  <outConfig>
    <mgmtIf dn="sys/chassis-1/server-1/mgmt/if-1"
     description="CIMC Interface Network Settings"
     id="1" v4IPAddr="10.104.255.181" mac="F4:0F:1B:FA:48:3C"
```

```
     hostname="C3160-FCH1821JAV4" v6extEnabled="yes"
     v6IPAddr="2002:200::125" v6prefix="64" v6extGw="::"
     v6linkLocal="fe80::f60f:1bff:fefa:483c"
     subject="blade" status="modified"/>
  </outConfig>
</configConfMo>
```

Configuring Static IPv4

Request

```
<configConfMo cookie="0956722178/02141d24-2366-1623-8002-82f92a02b884"
   dn="sys/chassis-1/if-1">
<inConfig>
    <mgmtIf dn="sys/chassis-1/if-1" dhcpEnable="no"
     extMask="255.255.255.128" extGw="10.104.255.129"
     v4IPAddrBmc1="10.104.255.246" v4IPAddrCmc2="10.104.255.247"
     v4IPAddrCmc1="10.104.255.248" extIp="10.104.255.245" />
</inConfig>
</configConfMo>
```

Note There is no response because the IPv4 configuration terminates all the SSH/WebUI/XMLAPI sessions. You need to reconnect to the server with the new IPv4 address configuration.

Configuring DHCPv4

Request

```
<configConfMo cookie="0956737017/af85c151-2666-1626-8002-82f92a02b884"
  dn="sys/chassis-1/if-1">
<inConfig>
    <mgmtIf dn="sys/chassis-1/if-1"
     dhcpEnable="yes" extMask="255.255.255.128" extGw="10.104.255.129" />
</inConfig>
</configConfMo>
```

Note There is no response because the IPv4 configuration terminates all the SSH/WebUI/XMLAPI sessions. You need to reconnect to the server with the new IPv4 address configuration.

Configuring Static IPv6 and IPv4

Request

```
<configConfMoconfigConfMo cookie="0956739710/edb4ef0e-2766-1627-8003-82f92a02b884"
   dn="sys/chassis-1/if-1">
<inConfig>
    <mgmtIf dn="sys/chassis-1/if-1" dhcpEnable="no"
     extMask="255.255.255.128" extGw="10.104.255.129"
     v4IPAddrBmc1="10.104.255.246" v4IPAddrCmc2="10.104.255.247"
     v4IPAddrCmc1="10.104.255.248" extIp="10.104.255.245"
     v6dhcpEnable="no" v6prefix="64" v6extIp="2001:200::125"
     v6extGw="2000:200::125" v6extEnabled="yes" v6IPAddrBmc1="2002:200::125"
     v6IPAddrCmc1="2004:200::125" v6IPAddrCmc2="2005:200::125"/>
</inConfig>
</configConfMo>
```

Note There is no response because the IPv4 configuration terminates all the SSH/WebUI/XMLAPI sessions. You need to reconnect to the server with the new IPv4 address configuration.

Configuring Network Interface Card Mode

Request

```
<configConfMo cookie='0949273658/385bfaaf-5c5f-1f5c-8004-d37c77e2cff4'
 dn='sys/chassis-1/if-1'>
 <inConfig>
   <mgmtIf nicMode="cisco_card" nicRedundancy="active-active"/>
 </inConfig>
</configConfMo>
```

Note There is no response due to a valid change in the card mode. You can verify the NIC mode and NIC redundancy values (4 and 1 respectively) by using the command **nicMode="cisco_card" nicRedundancy="active-active"** from the CMC command line. CMC1, CMC2, Cisco IMC 1 and Cisco IMC 2 must reflect the same values (4 and 1).

Examples of Virtual Interface Tasks

The examples in this section show how to use the Cisco IMC XML API to retrieve and perform virtual interface tasks. Each example shows the XML API request followed by the response from Cisco IMC.

This section includes the following examples:

Retrieving Virtual Interface Configuration

Request

```
<configResolveDn cookie='0956568556/dd519b64-ff65-15ff-8003-82f92a02b884'
 inHierarchical='false' dn='sys/chassis-1/if-1'/>
```

Response

```
<configResolveDn cookie="0956568556/dd519b64-ff65-15ff-8003-82f92a02b884"
 response="yes" dn="sys/chassis-1/if-1">
  <outConfig>
    <mgmtIf dn="sys/chassis-1/if-1"
```

```
    description="Virtual Management Interface Network Settings"
    id="1" extEnabled="yes" extIp="" v4IPAddrCmc1="0.0.0.0"
    v4IPAddrCmc2="0.0.0.0" v4IPAddrBmc1="0.0.0.0" v4IPAddrBmc2="0.0.0.0"
    extMask="255.255.255.128" extGw="10.104.255.129" ifType="virtual"
    mac="84:B8:02:2A:F9:82" vHostname="" dhcpEnable="yes" dnsUsingDhcp="no"
    ddnsEnable="yes" ddnsDomain="" dnsPreferred="171.68.226.120"
    dnsAlternate="0.0.0.0" nicMode="dedicated" vicSlot="1" nicRedundancy="none"
    vlanEnable="no" vlanId="1" vlanPriority="2" portProfile=""
    v6extEnabled="yes" v6extIp="::" v6extGw="::" v6prefix="64"
    v6linkLocal="fe80::86b8:2ff:fe2a:f982" v6SlaacIp="::"
    v6dhcpEnable="yes" v6dnsUsingDhcp="no" v6dnsPreferred="::"
    v6dnsAlternate="::" v6IPAddrCmc1="::" v6IPAddrCmc2="::"
    v6IPAddrBmc1="::" v6IPAddrBmc2="::" subject="virtual"/>
  </outConfig>
</configResolveDn>
```

Enabling IPv6 Using DHCPv6

Request

```
<configConfMo cookie="1435897051/c7da4b34-f119-19f1-8002-838077e2cff4"
 dn="sys/chassis-1/if-1">
<inConfig>
    <mgmtIf dn="sys/chassis-1/if-1" v6extEnabled="yes"  v6dhcpEnable="yes"/>
</inConfig>
</configConfMo>
```

Response

```
<configConfMo cookie="1435897051/c7da4b34-f119-19f1-8002-838077e2cff4"
 response="yes" dn="sys/chassis-1/if-1" >
  <outConfig>
    <mgmtIf dn="sys/chassis-1/if-1"
     description="Virtual Management Interface Network Settings"
     id="1" extEnabled="yes" extIp="10.106.145.166" v4IPAddrCmc1="0.0.0.0"
     v4IPAddrCmc2="0.0.0.0" v4IPAddrBmc1="0.0.0.0" v4IPAddrBmc2="0.0.0.0"
     extMask="255.255.255.0" extGw="10.106.145.1" ifType="virtual"
     mac="F4:CF:E2:77:80:83" vHostname="C3260-FCH18207WF3"
     dhcpEnable="yes" dnsUsingDhcp="yes" ddnsEnable="yes"
     ddnsDomain="" dnsPreferred="171.70.168.183" dnsAlternate="0.0.0.0"
     nicMode="dedicated" vicSlot="2" nicRedundancy="none" vlanEnable="no"
     vlanId="1" vlanPriority="0" portProfile="" v6extEnabled="no"
     v6extIp="::" v6extGw="::" v6prefix="64" v6linkLocal="::" v6SlaacIp="::"
     v6dhcpEnable="no" v6dnsUsingDhcp="no" v6dnsPreferred="::"
     v6dnsAlternate="::" v6IPAddrCmc1="::" v6IPAddrCmc2="::"
     v6IPAddrBmc1="::" v6IPAddrBmc2="::" subject="virtual" status="modified"/>
  </outConfig>
</configConfMo>
```

Note The **v6extEnabled** is a common attribute and it is applied to the network interfaces of all the components. So, in the above response this attribute is shown as **No**. However, if you perform Get operation or use CLI to verify this attribute, the IPv6 is shown as **Enabled**.

Disabling IPv6

Request

```
<configConfMo cookie="1435897793/485b3ad8-f119-19f1-8003-838077e2cff4"
 dn="sys/chassis-1/if-1">
<inConfig>
    <mgmtIf dn="sys/chassis-1/if-1" v6extEnabled="no"/>
</inConfig>
</configConfMo>
```

Response

```
<configConfMocookie="1435897793/485b3ad8-f119-19f1-8003-838077e2cff4"
response="yes"  dn="sys/chassis-1/if-1" >
  <outConfig>
    <mgmtIf dn="sys/chassis-1/if-1"
     description="Virtual Management Interface Network Settings" id="1"
     extEnabled="yes" extIp="10.106.145.166" v4IPAddrCmc1="0.0.0.0"
     v4IPAddrCmc2="0.0.0.0" v4IPAddrBmc1="0.0.0.0" v4IPAddrBmc2="0.0.0.0"
     extMask="255.255.255.0" extGw="10.106.145.1" ifType="virtual"
     mac="F4:CF:E2:77:80:83" vHostname="C3260-FCH18207WF3" dhcpEnable="yes"
     dnsUsingDhcp="yes" ddnsEnable="yes" ddnsDomain=""
     dnsPreferred="171.70.168.183" dnsAlternate="0.0.0.0"
     nicMode="dedicated" vicSlot="2" nicRedundancy="none"
     vlanEnable="no" vlanId="1" vlanPriority="0" portProfile=""
     v6extEnabled="no" v6extIp="2010:201::283"
     v6extGw="fe80::5e50:15ff:fef0:b042" v6prefix="119"
     v6linkLocal="fe80::f6cf:e2ff:fe77:8083"
     v6SlaacIp="2010:201::f6cf:e2ff:fe77:8083" v6dhcpEnable="yes"
     v6dnsUsingDhcp="no" v6dnsPreferred="::" v6dnsAlternate="::"
     v6IPAddrCmc1="::" v6IPAddrCmc2="::" v6IPAddrBmc1="::"
     v6IPAddrBmc2="::" subject="virtual" status="modified"/>
  </outConfig>
</configConfMo>
```

Note The **v6extEnabled** is a common attribute and it is applied to the network interfaces of all the components. So, in the above response this attribute is shown as **Yes**. However, if you perform Get operation or use CLI to verify this attribute, the IPv6 is shown as **Disabled**.

Enabling Static IPv6 on Virtual Interface

Request

```
<configConfMo cookie="1435898185/7cbeb77f-f119-19f1-8004-838077e2cff4"
 dn="sys/chassis-1/if-1">
<inConfig>
    <mgmtIf dn="sys/chassis-1/if-1" v6extEnabled="yes"
     v6prefix="64" v6dnsUsingDhcp="no" v6dhcpEnable="no"
     v6extIp="2010:201::97" v6extGw="2010:201::1"/>
</inConfig>
</configConfMo>
```

Response

```
<configConfMo cookie="1435898185/7cbeb77f-f119-19f1-8004-838077e2cff4"
 response="yes" dn="sys/chassis-1/if-1"
errorCode="2999" invocationResult="unidentified-fail"
errorDescr="Invalid request - component bmc1 - Missing or invalid IPv6 address provided"/>
```

Note When you configure static IPv6, provide static IPv6 addresses for all the CMC-X, BMC-X and virtual interface.

Configuring DNS servers

Request

```
<configConfMo cookie="0956619238/87045c2f-0b66-160b-8003-82f92a02b884"
 dn="sys/chassis-1/if-1">
<inConfig>
```

```
  <mgmtIf dn="sys/chassis-1/if-1" v6extEnabled="yes"
  v6dnsPreferred="2005:420:54ff:74::204:6120" v6dnsAlternate="2005:201::200"/>
</inConfig>
</configConfMo>
```

Response

```
<configConfMo cookie="0956619238/87045c2f-0b66-160b-8003-82f92a02b884"
 response="yes" dn="sys/chassis-1/if-1">
  <outConfig>
    <mgmtIf dn="sys/chassis-1/if-1"
      description="Virtual Management Interface Network Settings"
      id="1" extEnabled="yes" extIp="" v4IPAddrCmc1="0.0.0.0"
      v4IPAddrCmc2="0.0.0.0" v4IPAddrBmc1="0.0.0.0"
      v4IPAddrBmc2="0.0.0.0" extMask="255.255.255.128"
      extGw="10.104.255.129" ifType="virtual" mac="84:B8:02:2A:F9:82"
      vHostname="" dhcpEnable="yes" dnsUsingDhcp="no" ddnsEnable="yes"
      ddnsDomain="" dnsPreferred="171.68.226.120" dnsAlternate="0.0.0.0"
      nicMode="dedicated" vicSlot="1" nicRedundancy="none"
      vlanEnable="no" vlanId="1" vlanPriority="2" portProfile=""
      v6extEnabled="yes" v6extIp="::" v6extGw="::" v6prefix="64"
      v6linkLocal="fe80::86b8:2ff:fe2a:f982" v6SlaacIp="::"
      v6dhcpEnable="yes" v6dnsUsingDhcp="no"
      v6dnsPreferred="2010:420:54ff:74::204:6120" v6dnsAlternate="2010:201::200"
      v6IPAddrCmc1="::" v6IPAddrCmc2="::" v6IPAddrBmc1="::"
      v6IPAddrBmc2="::" subject="virtual" status="modified"/>
  </outConfig>
</configConfMo>
```

Configuring Static IPv6

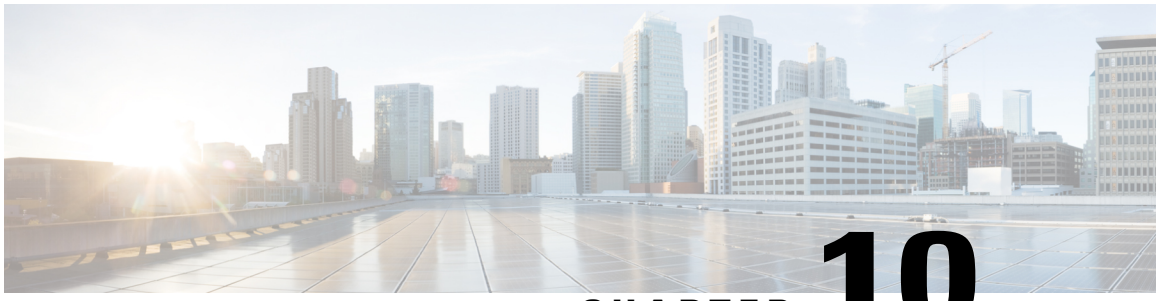
The following example shows to configure static IPv6 on UCS-C3260 M4 server with 2 SIOC and a server node.

Request

```
<configConfMo cookie="0956641159/6f8470a7-0f66-160f-8002-82f92a02b884"
 dn="sys/chassis-1/if-1">
<inConfig>
    <mgmtIf dn="sys/chassis-1/if-1"  v6dhcpEnable="no"
     v6prefix="64" v6extIp="2001:200::125"
     v6extGw="2000:200::125" v6extEnabled="yes"
      v6IPAddrBmc1="2002:200::125" v6IPAddrCmc1="2004:200::125"
     v6IPAddrCmc2="2005:200::125"/>
</inConfig>
</configConfMo
```

Response

```
<configConfMo cookie="0956641159/6f8470a7-0f66-160f-8002-82f92a02b884"
 response="yes" dn="sys/chassis-1/if-1" >
  <outConfig>
    <mgmtIf dn="sys/chassis-1/if-1"
     description="Virtual Management Interface Network Settings"
     id="1" extEnabled="yes" extIp="" v4IPAddrCmc1="0.0.0.0"
     v4IPAddrCmc2="0.0.0.0" v4IPAddrBmc1="0.0.0.0"
     v4IPAddrBmc2="0.0.0.0" extMask="" extGw="" ifType="virtual"
     mac="84:B8:02:2A:F9:82" vHostname="" dhcpEnable="yes" dnsUsingDhcp="no"
     ddnsEnable="yes" ddnsDomain="" dnsPreferred="171.68.226.120"
     dnsAlternate="0.0.0.0" nicMode="dedicated" vicSlot="1"
     nicRedundancy="none" vlanEnable="no" vlanId="1" vlanPriority="2"
     portProfile="" v6extEnabled="yes" v6extIp="::" v6extGw="::"
     v6prefix="64" v6linkLocal="fe80::86b8:2ff:fe2a:f982" v6SlaacIp="::"
     v6dhcpEnable="yes" v6dnsUsingDhcp="no" v6dnsPreferred="2005:420:54ff:74::204:6120"
     v6dnsAlternate="2005:201::200" v6IPAddrCmc1="::"
     v6IPAddrCmc2="::" v6IPAddrBmc1="::" v6IPAddrBmc2="::"
     subject="virtual" status="modified"/>
  </outConfig>
</configConfMo>
```



CHAPTER 10

Managing Storage Adapters

This chapter includes the following sections:

Server RAID Management Tasks

The examples in this section show how to use the Cisco IMC XML API to retrieve and perform server RAID management tasks. Each example shows the XML API request followed by the response from Cisco IMC.

This section includes the following examples:

- Clearing a Virtual Drive as Transport Ready, on page 91
- Setting up a Virtual Drive as Transport Ready with Exclude-All Options, on page 90
- Clearing a Virtual Drive as Transport Ready, on page 91
- Retrieving NVMe Storage Controller Details, on page 92
- Retrieving Firmware Version of a NVMe Controller, on page 93

Retrieving Storage Controller Using DN

Request:

```
<configResolveDn cookie="1422613813/9657c746-dc0d-1ddc-8004-91fabb1b0ff4"
inHierarchical="false" dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ"/>
```

Response:

```
<configResolveDn cookie="1422613813/9657c746-dc0d-1ddc-8004-91fabb1b0ff4" response="yes"
dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ"/>
  <outConfig>
    <storageController id="SLOT-MEZZ"
         model="RAID controller for UCS C3X60 Storage Servers"
     pciSlot="SLOT-MEZZ" presence="equipped" raidSupport="yes"
     serial="" type="SAS"
     vendor="LSI Logic" adminAction="no-op"
         dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ"/>
  </outConfig>
</configResolveDn>
```

Retrieving Storage Controller Properties

Request:

```
<configResolveClass cookie="1422613813/9657c746-dc0d-1ddc-8004-91fabb1b0ff4"
classId="storageControllerProps" inHierarchical="false"/>
```

Response:

```
<configResolveClass cookie="1422613813/9657c746-dc0d-1ddc-8004-91fabb1b0ff4"
     response="yes" classId="storageControllerProps">
 <outConfigs>
    <storageControllerProps
         dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/controller-props"
    pciSlot="SLOT-MEZZ" health="Severe Fault"
        controllerStatus="Optimal" batteryStatus="Optimal"
        bbuPresent="true" raidChipTempCentigrade="82" backendPortCount="8"
    memorySize="4095 MB" cacheMemorySize="3534 MB"
    virtualDriveCount="1" degradedVirtualDriveCount="0"
       offlineVirtualDriveCount="0" physicalDriveCount="60"
       criticalPhysicalDriveCount="0" failedPhysicalDriveCount="5"
    memoryCorrectableErrors="0" memoryUncorrectableErrors="0" bootDrive="0"
        bootDriveIsPhysicalDrive="false" supportsRaid0="true" supportsRaid1="true"
        supportsRaid5="true" supportsRaid6="true" supportsRaid00="true"
        supportsRaid10="true" supportsRaid50="true"
supportsRaid60="true"supportsRaid1e="false"
        supportsRaid1erlq0="true" supportsRaid1e0rlq0="true" supportsRaidsrl03="true"
        dateOfManufacture="N/A" revision="N/A" prebootCliVersion="01.07-05:#%0000"
    webBiosVersion="5.08-0007" nvdataVersion="3.1411.00-0011"
        bootBlockVersion="3.06.00.00-0001" bootVersion="N/A"
        nvramPresent="true" serialDebuggerPresent="true" flashPresent="true"
        sasAddress0="500e004aaaaabb7f" sasAddress1="500e004aaaaaaa7f"
        sasAddress2="0000000000000000" sasAddress3="0000000000000000"
        sasAddress4="0000000000000000" sasAddress5="0000000000000000"
        sasAddress6="0000000000000000" sasAddress7="0000000000000000"
        serial="" firmwarePackageBuild="24.7.0-0030" ttyLogStatus="Not Downloaded"/>
```

```
  </outConfigs>
</configResolveClass>
```

Retrieving Storage Raid Battery Properties

Request:

```
<configResolveDn cookie="1422614651/60391de0-dc0d-1ddc-8005-91fabb1b0ff4"
inHierarchical="false" dn='sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/raid-battery'/>
```

Response:

```
<configResolveDn cookie="1422614651/60391de0-dc0d-1ddc-8005-91fabb1b0ff4" response="yes"
dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/raid-battery">
  <outConfig>
    <storageRaidBattery dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/raid-battery"

      batteryType="TMM-C SuperCap" health="Good" batteryStatus="Optimal"
      batteryPresent="true" chargingState="N/A" retentionTime="N/A"
 temperature="36 degrees C" temperatureHigh="false"
      designVoltage="9.500 V" voltage="9.464 V" current="0.000 A" learnMode="Auto"
      completedChargeCycles="N/A" learnCycleStatus="Successful" learnCycleRequested="false"

      nextLearnCycle="2015-02-09 09:02" designCapacity="288 Joules" fullCapacity="N/A"
      remainingCapacity="N/A" relativeStateOfCharge="N/A" absoluteStateOfCharge="N/A"
      expectedMarginOfError="N/A" manufacturer="LSI" dateOfManufacture="2013-08-15"
      serialNumber="6582" firmwareVersion="26535-02" adminAction="no-op"/>
  </outConfig>
</configResolveDn>
```

Retrieving Storage Virtual Drive Properties

Request:

```
<configResolveClass cookie="1422614651/60391de0-dc0d-1ddc-8005-91fabb1b0ff4"
classId="storageVirtualDrive" inHierarchical="false"/>
```

Response:

```
<configResolveClass cookie="1422614651/60391de0-dc0d-1ddc-8005-91fabb1b0ff4"
 response="yes" classId="storageVirtualDrive">
 <outConfigs>
   <storageVirtualDrive id="0" name="junk_vd" raidLevel="RAID 5" size="399999 MB"
   vdStatus="Optimal" health="Good"  bootDrive="true"
   stripSize="64 KB" drivesPerSpan="4"
   spanDepth="1" accessPolicy="Read-Write" cachePolicy="Direct"
   readAheadPolicy="None"
   requestedWriteCachePolicy="Write Through" currentWriteCachePolicy="Write Through"
   diskCachePolicy="Unchanged" allowBackgroundInit="true" autoSnapshot="false"
   autoDeleteOldest="true" driveState="Optimal" adminAction="no-op" targetId="0"
physicalDrivesList="Physical drives to be used for virtual drive reconstruction.
pd_id1,pd_id2,...pd_idN"
   dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/vd-0"/>
 </outConfigs>
</configResolveClass>
```

Retrieving Storage Virtual Drive Properties using DN

Request:

```
<configResolveDn cookie="1422614651/60391de0-dc0d-1ddc-8005-91fabb1b0ff4"
inHierarchical="false" dn='sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/vd-0'/>
```

Response:

```
<configResolveDn cookie="1422614651/60391de0-dc0d-1ddc-8005-91fabb1b0ff4" response="yes"
dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/vd-0">
```

```
 <outConfig>
  <storageVirtualDrive id="0" name="junk_vd" raidLevel="RAID 5"
   size="399999 MB" vdStatus="Optimal" health="Good" bootDrive="true"
       stripSize="64 KB" drivesPerSpan="4" spanDepth="1" accessPolicy="Read-Write"
      cachePolicy="Direct" readAheadPolicy="None" requestedWriteCachePolicy="Write Through"

       currentWriteCachePolicy="Write Through" diskCachePolicy="Unchanged"
       allowBackgroundInit="true" autoSnapshot="false" autoDeleteOldest="true"
   driveState="Optimal" adminAction="no-op" targetId="0"
    physicalDrivesList="Physical drives to be used for virtual drive reconstruction.
...pd_id1,pd_id2,...pd_idN"
   dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/vd-0"/>
  </outConfig>
</configResolveDn>
```

Retrieving Storage Controller Settings using DN

Request:

```
<configResolveDn
cookie="1422614651/60391de0-dc0d-1ddc-8005-91fabb1b0ff4"inHierarchical="false"
 dn='sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/controller-settings'/>
```

Response:

```
<configResolveDn cookie="1422614651/60391de0-dc0d-1ddc-8005-91fabb1b0ff4" response="yes"
dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/controller-settings">
 <outConfig>
  <storageControllerSettings
dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/controller-settings"
   pciSlot="SLOT-MEZZ" predictiveFailPollInterval="300
   sec" rebuildRate="30 %" patrolReadRate="30 %"
   consistencyCheckRate="30 %" reconstructionRate="30 %"
   cacheFlushInterval="4 sec" spinupDriveCount="4" spinupDelay="6 sec"
   physDriveCoercionMode="1 GB" clusterEnable="false" batteryWarning="true"
   eccBucketLeakRate="1440 min" exposeEnclosureDevices="true" maintainPdFailHistory="false"

   enableCopybackOnSmart="true" enableCopybackToSsdOnSmartError="true"
   ncqStatus="enabled" enableJbod="false" spinDownUnconfigured="true"
enableSsdPatrolRead="false"
   autoEnhancedImport="true"/>
   </outConfig>
</configResolveDn>
```

Retrieving Storage Local Disk Properties

Request:

```
<configResolveClass cookie="1422618600/f24b2a39-dd0d-1ddd-8006-91fabb1b0ff4"
classId="storageLocalDisk" inHierarchical="true"/>
```

Response:

```
<configResolveClass cookie="1422618600/f24b2a39-dd0d-1ddd-8006-91fabb1b0ff4"
     response="yes" classId="storageLocalDisk">
 <outConfigs>
  <storageLocalDisk id="1" pdStatus="Online" health="Good" predictiveFailureCount="0"
  linkSpeed="6.0 Gb/s" interfaceType="SAS" mediaType="HDD" coercedSize="3814697
      MB" vendor="WD" productId="WD4001FYYG-01SL3" driveFirmware="VR07"
      driveSerialNumber="WMC1F1927484" driveState="online" online="true"
      dedicatedHotSpareForVDId="" adminAction="no-op"
  dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/pd-1">
     <storageLocalDiskProps rn="general-props" physicalDrive="1"
      pdStatus="Online" health="Good" enclosureDeviceId="103"
     deviceId="43" sequenceNumber="2"
     mediaErrorCount="0" otherErrorCount="0" predictiveFailureCount="0"
  linkSpeed="6.0 Gb/s" interfaceType="SAS" mediaType="HDD"
      blockSize="512" blockCount="7814037168"
  bootDrive="false" rawSize="3815447 MB"
```

```
      nonCoercedSize="3814935 MB"
      coercedSize="3814697 MB" powerState="active" sasAddress0="50000c0f01e85d26"
      sasAddress1="50000c0f01e85d27"/>
      <storageOperation rn="storage-operation" lropInProgress="false"
        currentLrop="No operation in progress" progressPercent="0" elapsedSeconds="0"/>
      </storageLocalDisk>
...
  <storageLocalDisk id="60" pdStatus="Unconfigured Good"
    health="Good" predictiveFailureCount="0" linkSpeed="6.0 Gb/s"
        interfaceType="SAS" mediaType="HDD" coercedSize="3814697 MB" vendor="WD"
        productId="WD4001FYYG-01SL3" driveFirmware="VR07"
        driveSerialNumber="WMC1F1926558" driveState="unconfigured good" online="true"
   dedicatedHotSpareForVDId="" adminAction="no-op"
       dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/pd-60">
  <storageLocalDiskProps rn="general-props" physicalDrive="60" pdStatus="Unconfigured Good"

       health="Good" enclosureDeviceId="103" deviceId="102" sequenceNumber="1"
       mediaErrorCount="0" otherErrorCount="0" predictiveFailureCount="0"
       linkSpeed="6.0 Gb/s" interfaceType="SAS" mediaType="HDD" blockSize="512"
       blockCount="7814037168" bootDrive="false" rawSize="3815447 MB"
       nonCoercedSize="3814935 MB" coercedSize="3814697 MB" powerState="active"
       sasAddress0="50000c0f012eb1f6" sasAddress1="50000c0f012eb1f7"/>
  <storageOperation rn="storage-operation" lropInProgress="false"
       currentLrop="No operation in progress" progressPercent="0" elapsedSeconds="0"/>
    </storageLocalDisk>
  </outConfigs>
</configResolveClass>
```

Retrieving Storage Local Disk Properties using DN

Request:

```
<configResolveDn cookie="1422619590/3ebc17b5-dd0d-1ddd-8007-91fabb1b0ff4"
inHierarchical="true"
dn='sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/pd-25'/>
```

Response:

```
<configResolveDn cookie="1422619590/3ebc17b5-dd0d-1ddd-8007-91fabb1b0ff4" response="yes"
dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/pd-25">
  <outConfig>
    <storageLocalDisk id="25" pdStatus="Unconfigured Good" health="Good"
     predictiveFailureCount="0" linkSpeed="6.0 Gb/s" interfaceType="SAS" mediaType="HDD"
     coercedSize="3814697 MB" vendor="TOSHIBA" productId="MG03SCA400"
     driveFirmware="5702" driveSerialNumber="14O0A0C8FVU4" driveState="unconfigured good"
online="true"
     dedicatedHotSpareForVDId="" adminAction="no-op"
     dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/pd-25">
      <storageLocalDiskProps rn="general-props" physicalDrive="25"
     pdStatus="Unconfigured Good" health="Good" enclosureDeviceId="103" deviceId="67"
sequenceNumber="1"
     mediaErrorCount="0" otherErrorCount="0" predictiveFailureCount="0"
     linkSpeed="6.0 Gb/s" interfaceType="SAS" mediaType="HDD" blockSize="512"
blockCount="7814037168"
     bootDrive="false" rawSize="3815447 MB" nonCoercedSize="3814935 MB"
     coercedSize="3814697 MB" powerState="active" sasAddress0="5000039548e07caa"
     sasAddress1="5000039548e07cab"/>
     <storageOperation rn="storage-operation" lropInProgress="false"
     currentLrop="No operation in progress" progressPercent="0" elapsedSeconds="0"/>
    </storageLocalDisk>
  </outConfig>
</configResolveDn>
```

Retrieving Unused Local Disk Details

Request:

```
<configResolveClass cookie="1422693427/17670951-ef0d-1def-8002-91fabb1b0ff4"
classId="storageUnusedLocalDisk" inHierarchical="false"> </configResolveClass>
```

Response:

```
<configResolveClass cookie="1422693427/17670951-ef0d-1def-8002-91fabb1b0ff4" response="yes"

classId="storageUnusedLocalDisk">
  <outConfigs>
    <storageUnusedLocalDisk id="5" pdStatus="Unconfigured Good" coercedSize="381
     4697 MB" health="Good" mediaType="HDD" vendor="WD"
     dn="sys/chassis-1/server-1/bord/storage-SAS-SLOT-MEZZ/virtual-drive-create/pd-5"/>
 <storageUnusedLocalDisk
     id="6" pdStatus="Unconfigured Good" coercedSize="3814697 MB"
     health="Good" mediaType="HDD" vendor="WD"
     dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/virtual-drive-create/pd-6"/>
...
 <storageUnusedLocalDisk id="59" pdStatus="Unconfigured Good" coercedSize="3814697 MB"
     health="Good" mediaType="HDD" vendor="WD"
     dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/virtual-drive-create/pd-59"/>
    <storageUnusedLocalDisk id="60" pdStatus="Unconfigured Good"
     coercedSize="3814697 MB" health="Good" mediaType="HDD" vendor="WD"
     dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/virtual-drive-create/pd-60"/>
  </outConfigs>
</configResolveClass>
```

Retrieving Storage Virtual Drive with Drive Group Space

Request:

```
<configResolveClass cookie="1422693427/17670951-ef0d-1def-8002-91fabb1b0ff4"
inHierarchical="false"classId="storageVirtualDriveWithDriveGroupSpace">
</configResolveClass>
```

Response:

```
<configResolveClass cookie="1422693427/17670951-ef0d-1def-8002-91fabb1b0ff4"
   response="yes" classId="storageVirtualDriveWithDriveGroupSpace">
  <outConfigs>
    <storageVirtualDriveWithDriveGroupSpace id="0" name="junk_vd" raidLevel="5"
     usedPhysicalDriveIds="1,2,3,4" maxAvailableSpace="11044089 MB"
     vdStatus="Optimal" health="Good"
     dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/virtual-drive-carve/vd-0"/>
  </outConfigs>
</configResolveClass>
```

Retrieving Storage Firmware Boot Loader Version

Request:

```
<configResolveDn cookie="1422620517/18f4158f-de0d-1dde-8008-91fabb1b0ff4"
inHierarchical="false"
 dn='sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/fw-boot-loader'/>
```

Response:

```
<configResolveDn cookie="1422620517/18f4158f-de0d-1dde-8008-91fabb1b0ff4"
response="yes" dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/fw-boot-loader">
  <outConfig>
    <firmwareRunning dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/fw-boot-loader"

     description="LSI Storage controller currently running boot loader firmware version"
     deployment="boot-loader" type="storage-controller"
version="6.22.03.0_4.16.08.00_0x060B0200"/>
  </outConfig>
</configResolveDn>
```

Retrieving Storage Firmware Running Version

Request:

```
<configResolveDn cookie="1422621263/953322c7-de0d-1dde-8009-91fabb1b0ff4"
inHierarchical="false"dn='sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/fw-system'
/>
```

Response:

```
<configResolveDn cookie="1422621263/953322c7-de0d-1dde-8009-91fabb1b0ff4" response="yes"
dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/fw-system">
  <outConfig>
    <firmwareRunning dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/fw-system"
     description="LSI Storage controller currently running firmware version"
deployment="system"
     type="storage-controller" version="4.270.00-3995"/>
  </outConfig>
</configResolveDn>
```

Retrieving Storage Firmware Boot Definition

Request:

```
<configResolveDn cookie="1422621263/953322c7-de0d-1dde-8009-91fabb1b0ff4"
inHierarchical="true"dn='sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/fw-boot-def'/>
```

Response:

```
<configResolveDn cookie="1422621263/953322c7-de0d-1dde-8009-91fabb1b0ff4" response="yes"
dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/fw-boot-def">
  <outConfig>
    <firmwareBootDefinition
dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/fw-boot-def"
     type="storage-controller"> <firmwareBootUnit rn="bootunit-combined"
     description="LSI Storage controller startup BIOS firmware version"
     adminState="triggered" image="running" resetOnActivate="no"
     type="combined" version="6.22.03.0_4.16.08.00_0x060B0200"/>
    </firmwareBootDefinition>
  </outConfig>
</configResolveDn>
```

Creating a Virtual Drive from Unused Physical Drive

Request:

```
<configConfMo cookie="cookiecutter"
inHierarchical="false"
dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/virtual-drive-create">
```

Response:

```
<configConfMo cookie="1422704637/95968593-f10d-1df1-8002-91fabb1b0ff4" response="yes">
dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/virtual-drive-create"
  <outConfig>
    <storageVirtualDriveCreatorUsingUnusedPhysicalDrive
     dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/virtual-drive-create"
 description="Create a new virtual drive from unused physical drive(s) available in the
rack server.
    Children objects of this object, lists all currently available unused physical drives."

     minRequiredPhysicalDrives="RAID 0 at least 1 physical drive, RAID 1 at least 2
    physical drives, RAID 5 at least 3 physical drives, RAID 6 at least 3 physical drives,

     RAID 10 at least 4 physical drives, RAID 50 or RAID 60 at least 6 physical drives"
 virtualDriveName="" raidLevel="" size=""
     driveGroup="[span1_pd_id1,span1_pd_id2,span1_pd_id3...,span1_pd_idN]
    [span2_pd_id1,span2_pd_id2,span2_pd_id3...,span2_pd_idN]..."
```

```
    writePolicy="" createdVirtualDriveDn="" operStatus=""
    adminState="triggered" status="modified"/>
  </outConfig>
</configConfMo>
```

Deleting a Virtual Drive

Request:

```
<configConfMo cookie="1422880823/ae111d8a-1a0e-1e1a-8002-91fabb1b0ff4"
inHierarchical="false" dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/vd-0">
```

Response:

```
<configConfMo cookie="1422880823/ae111d8a-1a0e-1e1a-8002-91fabb1b0ff4" response="yes">
dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/vd-0"
  <outConfig>
</outConfig>
</configConfMo>
```

Modifying a Virtual Drive

Request:

```
<configConfMo cookie="1424262372/b1fdd9a1-5c0f-1f5c-8002-91fabb1b0ff4"
dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/vd-3">
<inConfig>
 <storageVirtualDrive dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/vd-3"
   id="3" adminAction="set-boot-drive"/>
 </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/vd-3"
 cookie="1424262372/b1fdd9a1-5c0f-1f5c-8002-91fabb1b0ff4" response="yes">
<outConfig>
<storageVirtualDrive id="3" name="jchambers_vd" raidLevel="RAID 5" size="6000 MB"
  vdStatus="Optimal" health="Good" bootDrive="true" stripSize="64 KB" drivesPerSpan="4"
  spanDepth="1" accessPolicy="Read-Write" cachePolicy="Direct" readAheadPolicy="None"
  requestedWriteCachePolicy="Always Write Back" currentWriteCachePolicy="Write Back"
  diskCachePolicy="Unchanged" allowBackgroundInit="true" autoSnapshot="false"
  autoDeleteOldest="true" driveState="Optimal" adminAction="no-op" targetId="3"
  physicalDrivesList="Physical drives to be used for virtual drive reconstruction.
  pd_id1,pd_id2,...pd_idN" dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/vd-3"
  status="modified" ></storageVirtualDrive>
</outConfig>
</configConfMo>
```

Setting up a Virtual Drive as Transport Ready with Exclude-All Options

Request:

```
<configConfMo dn='sys/chassis-1/server-1/board/storage-SAS-SLOT-HBA/vd-0'
   inHierarchical='true' cookie='1390285111/1af97ad2-f075-1075-8003-bcbec261b284'>
   <inConfig>
<storageVirtualDrive dn='sys/chassis-1/server-1/board/storage-SAS-SLOT-HBA/vd-0' id='0'
adminAction='set-transport-ready' hotspareAction='exclude-all'/>
  </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-HBA/vd-0"
   cookie="1390285111/1af97ad2-f075-1075-8003-bcbec261b284" response="yes">
  <outConfig>
    <storageVirtualDrive id="0" name="RAID1_12" raidLevel="RAID 1"
```

```
    size="952720 MB" vdStatus="Optimal" health="Good" bootDrive="false" stripSize="64k"
     drivesPerSpan="2" spanDepth="1" accessPolicy="Transport Ready" cachePolicy="direct-io"

      readPolicy="no-read-ahead" requestedWriteCachePolicy="write-through"
       currentWriteCachePolicy="write-through" diskCachePolicy="unchanged"
allowBackgroundInit="true"
       autoSnapshot="false" autoDeleteOldest="true" driveState="Optimal" fdeCapable="no"
fdeEnabled="no"
      adminAction="no-op" hotspareAction="no-op" targetId="0"
       physicalDrivesList="Physical drives to be used for virtual drive reconstruction.
pd_id1,pd_id2,...pd_idN"
       dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-HBA/vd-0" status="modified">
     <storageLocalDiskUsage startingBlock="0" numberOfBlocks="1951170560" virtualDrive="0"

        physicalDrive="1" state="online" span="0" rn="pd-1" status="modified"/>
     <storageLocalDiskUsage startingBlock="0" numberOfBlocks="1951170560" virtualDrive="0"

       physicalDrive="2" state="online" span="0" rn="pd-2" status="modified"/>
      <storageOperation rn="storage-operation" lropInProgress="false"
      currentLrop="No operation in progress" progressPercent="0" elapsedSeconds="0"
status="modified"/>
    </storageVirtualDrive>
  </outConfig>
</configConfMo>
```

Note You can also set a Virtual Drive as Transport Ready using the **include-all** and **include-dhsp hotspareAction** values.

Clearing a Virtual Drive as Transport Ready

Request:

```
<configConfMo dn='sys/chassis-1/server-1/board/storage-SAS-SLOT-HBA/vd-0'
  inHierarchical='true' cookie='1390285111/1af97ad2-f075-1075-8003-bcbec261b284'>
  <inConfig>
   <storageVirtualDrive dn='sys/chassis-1/server-1/board/storage-SAS-SLOT-HBA/vd-0' id='0'

    adminAction='clear-transport-ready'/>
  </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-HBA/vd-0"
  cookie="1390285111/1af97ad2-f075-1075-8003-bcbec261b284" response="yes">
  <outConfig>
    <storageVirtualDrive id="0" name="RAID1_12" raidLevel="RAID 1" size="952720 MB"
vdStatus="Optimal"
     health="Good" bootDrive="false" stripSize="64k" drivesPerSpan="2"
      spanDepth="1" accessPolicy="read-write" cachePolicy="direct-io"
readPolicy="no-read-ahead"
       requestedWriteCachePolicy="write-through" currentWriteCachePolicy="write-through"
        diskCachePolicy="unchanged" allowBackgroundInit="true" autoSnapshot="false"
         autoDeleteOldest="true" driveState="Optimal" fdeCapable="no" fdeEnabled="no"

          adminAction="no-op" hotspareAction="no-op" targetId="0"
         physicalDrivesList="Physical drives to be used for virtual drive reconstruction.
        pd_id1,pd_id2,...pd_idN" dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-HBA/vd-0"

        status="modified">
      <storageLocalDiskUsage startingBlock="0" numberOfBlocks="1951170560" virtualDrive="0"

       physicalDrive="1" state="online" span="0" rn="pd-1" status="modified"/>
     <storageLocalDiskUsage startingBlock="0" numberOfBlocks="1951170560" virtualDrive="0"

    physicalDrive="2" state="online" span="0" rn="pd-2" status="modified"/>
```

```
     <storageOperation rn="storage-operation" lropInProgress="false"
    currentLrop="No operation in progress" progressPercent="0" elapsedSeconds="0"
status="modified"/>
    </storageVirtualDrive>
  </outConfig>
</configConfMo>
```

Retrieving NVMe Storage Controller Details

Request:

```
<configResolveClass cookie='1472022403/94a78a4a-3acc-1acc-8003-95542a2fa074'
inHierarchical='true' classId='storageControllerNVMe'/>"
```

Response:

```
<configResolveClass cookie="1472022403/94a78a4a-3acc-1acc-8003-95542a2fa074"
response="yes" classId="storageControllerNVMe">
<outConfigs>
 <storageControllerNVMe id="SLOT-5" model="Cisco UCS (SN150) HHHL 3800 GB NVMe based PCIe
SSD"
  vendor="HGST" serial="STM0001A7BD9" health="Good" controllerStatus="Optimal"
  controllerChipTempCelsius="39" driveLifeUsed="0" performanceLevel="100"
  ledFaultStatus="Healthy" percentageTotalPowerOnHour="6"
  dn="sys/chassis-1/server-1/board/storage-NVMe-SLOT-5" >
 <firmwareRunning rn="fw-boot-loader" description="NVMe Storage controller currently
  running boot loader firmware version" deployment="boot-loader"
  type="storage-controller-NVMe" version="N/A" >
 </firmwareRunning>
<firmwareRunning rn="fw-system" description="NVMe Storage controller
 currently running firmware version" deployment="system" type="storage-controller-NVMe"
 version="KMCCP105" >
</firmwareRunning>
</storageControllerNVMe>
<storageControllerNVMe
 id="SLOT-4" model="Cisco UCS (P3700) HHHL 800 GB NVMe based PCIe SSD" vendor="Intel"
 serial="CVFT6024002U800CGN" health="Good" controllerStatus="Optimal"
 controllerChipTempCelsius="24" driveLifeUsed="0" performanceLevel="100"
 ledFaultStatus="Healthy" percentageTotalPowerOnHour="5"
 dn="sys/chassis-1/server-1/board/storage-NVMe-SLOT-4" >
<firmwareRunning
 rn="fw-boot-loader" description="NVMe Storage controller currently running
 boot loader firmware version" deployment="boot-loader" type="storage-controller-NVMe"
 version="N/A" >
 </firmwareRunning>
<firmwareRunning
 rn="fw-system" description="NVMe Storage controller currently running firmware version"
 deployment="system" type="storage-controller-NVMe"
 version="8DV1CP01" >
 </firmwareRunning>
 </storageControllerNVMe>
<storageControllerNVMe
 id="SLOT-2" model="Cisco UCS (P3700) HHHL 1600 GB NVMe based PCIe SSD"
 vendor="Intel" serial="CVFT5433007S1P6DGN" health="Good" controllerStatus="Optimal"
 controllerChipTempCelsius="27" driveLifeUsed="0" performanceLevel="100"
 ledFaultStatus="Healthy" percentageTotalPowerOnHour="3"
 dn="sys/chassis-1/server-1/board/storage-NVMe-SLOT-2" >
<firmwareRunning
 rn="fw-boot-loader" description="NVMe Storage controller currently running boot
 loader firmware version" deployment="boot-loader" type="storage-controller-NVMe"
 version="N/A" >
 </firmwareRunning>
<firmwareRunning
 rn="fw-system" description="NVMe Storage controller currently running firmware version"
 deployment="system" type="storage-controller-NVMe" version="8DV1CP01" >
 </firmwareRunning>
 </storageControllerNVMe>
</outConfigs>
</configResolveClass>
```

Retrieving Firmware Version of a NVMe Controller

Request:

```
<configResolveDn cookie='1405200895/c56a2fcc-fe06-1e06-8062-f4f25b78ac58'
inHierarchical='false' dn='sys/chassis-1/server-1/board/storage-NVMe-FrontPCIe1/fw-system'/>
```

Response:

```
<configResolveDn cookie="1405200895/c56a2fcc-fe06-1e06-8062-f4f25b78ac58"
response="yes" dn="sys/chassis-1/server-1/board/storage-NVMe-FrontPCIe1/fw-system">
<outConfig>
 <firmwareRunning dn="sys/chassis-1/server-1/board/storage-NVMe-FrontPCIe1/fw-system"
  description="NVMe Storage controller currently running firmware version"
  deployment="system" type="storage-controller-NVMe"
 version="KMCCP105" >
</firmwareRunning>
</outConfig>
</configResolveDn>
```

Managing Storage Controllers

The examples in this section show how to use the Cisco IMC XML API to manage storage adapters. Each example shows the XML API request followed by the response from Cisco IMC.

This section includes the following examples:

Modifying Virtual Drive Write Policy

Request:

```
<configConfMo cookie='1424263006/67c0160e-5d0f-1f5d-8003-91fabb1b0ff4'
dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/vd-3">
 <inConfig>
  <storageVirtualDrive dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/vd-3"
   id="3" requestedWriteCachePolicy="Write Through"/>
 </inConfig>
</configConfMo>
```

Response:

```
<configConfMo cookie="1424263006/67c0160e-5d0f-1f5d-8003-91fabb1b0ff4" response="yes">
  dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/vd-3"
 <outConfig>
  <storageVirtualDrive id="3" name="jchambers_vd" raidLevel="RAID 5" size="6000 MB"
  vdStatus="Optimal" health="Good" bootDrive="true" stripSize="64 KB"
  drivesPerSpan="4" spanDepth="1" accessPolicy="Read-Write" cachePolicy="Direct"
  readAheadPolicy="None" requestedWriteCachePolicy=" Write Through"
```

```
  currentWriteCachePolicy="Write Through" diskCachePolicy="Unchanged"
  allowBackgroundInit="true" autoSnapshot="false" autoDeleteOldest="true"
  driveState="Optimal" adminAction="no-op" targetId="3"
  physicalDrivesList="Physical drives to be used for virtual drive reconstruction.
pd_id1,pd_id2,...pd_idN"
  dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/vd-3" status="modified">
  </storageVirtualDrive>
 </outConfig>
</configConfMo>
```

Making a Global Hot Spare

Request:

```
<configConfMo cookie=''1424263731/c78680e7-5c0f-1f5c-8004-91fabb1b0ff4"
 dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/pd-6">
   <inConfig>
     <storageLocalDisk dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/pd-6"
         id="6" adminAction="make-global-hot-spare"/>
   </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/pd-6"
cookie="1424263731/c78680e7-5c0f-1f5c-8004-91fabb1b0ff4" response="yes">
  <outConfig>
    <storageLocalDisk id="6" pdStatus="Global Hot Spare" health="Good"
     predictiveFailureCount="0" linkSpeed="6.0 Gb/s" interfaceType="SAS"
     mediaType="HDD" coercedSize="3814697 MB" vendor="WD" productId="WD4001FYYG-01SL3"
     driveFirmware="VR07" driveSerialNumber="WMC1F1928912" driveState="hot spare"
     online="false" dedicatedHotSpareForVDId="" adminAction="no-op"
     dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/pd-6"
     status="modified">
    </storageLocalDisk>
  </outConfig>
</configConfMo>
```

Removing Physical Disk from Hot Spare

Request:

```
 <configConfMocookie='1424264173/880690ac-5c0f-1f5c-8005-91fabb1b0ff4'
dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/pd-6">
   <inConfig>
    <storageLocalDisk
         dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/pd-6" id="6"
         adminAction="remove-hot-spare"/>
  </inConfig>
</configConfMo>
```

Response:

```
<configConfMo cookie="1424264173/880690ac-5c0f-1f5c-8005-91fabb1b0ff4" response="yes">
     dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/pd-6"
  <outConfig>
    <storageLocalDisk id="6" pdStatus="Unconfigured Good" health="Good"
     predictiveFailureCount="0" linkSpeed="6.0 Gb/s" interfaceType="SAS"
     mediaType="HDD" coercedSize="3814697 MB" vendor="WD" productId="WD4001FYYG-01SL3"
     driveFirmware="VR07" driveSerialNumber="WMC1F1928912" driveState="unconfigured good"
     online="true" dedicatedHotSpareForVDId="" adminAction="no-op"
     dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/pd-6"
     status="modified">
    </storageLocalDisk>
  </outConfig>
</configConfMo>
```

Making a Dedicated Hot Spare

Request:

```
<configConfMo cookie='1424264173/880690ac-5c0f-1f5c-8005-91fabb1b0ff4'
 dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/pd-6">
<inConfig>
```

Response:

```
<configConfMo cookie="1424264173/880690ac-5c0f-1f5c-8005-91fabb1b0ff4" response="yes">
     dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/pd-6"
    <outConfig>
      <storageLocalDisk id="6" pdStatus="Dedicated Hot Spare" health="Good"
       predictiveFailureCount="0" linkSpeed="6.0 Gb/s" interfaceType="SAS"
       mediaType="HDD" coercedSize="3814697 MB" vendor="WD" productId="WD4001FYYG-01SL3"
       driveFirmware="VR07" driveSerialNumber="WMC1F1928912" driveState="hot spare"
       online="false" dedicatedHotSpareForVDId="" adminAction="no-op"
       dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/pd-6"
       status="modified" >
     </storageLocalDisk>
   </outConfig>
</configConfMo>
```

Removing a Drive from Hot Spare Pools

Request:

```
  <configConfMo cookie='1424264173/880690ac-5c0f-1f5c-8005-91fabb1b0ff4'
 dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/pd-6">
    <inConfig>
      <storageLocalDisk dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/pd-6"
       adminAction="prepare-for-removal"/>
    </inConfig>
</configConfMo>
```

Response:

```
<configConfMo cookie="1424264173/880690ac-5c0f-1f5c-8005-91fabb1b0ff4" response="yes">
     dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/pd-6"
 <outConfig>
   <storageLocalDisk id="6" pdStatus="Ready to Remove" health="Moderate Fault"
    predictiveFailureCount="0"predictiveFailureCount="0" linkSpeed="6.0 Gb/s"
 interfaceType="SAS" mediaType="HDD"
    coercedSize="3814697 MB" vendor="WD" productId="WD4001FYYG-01SL3" driveFirmware="VR07"
    driveSerialNumber="WMC1F1928912" driveState="unconfigured good" online="true"
    dedicatedHotSpareForVDId="" adminAction="no-op"
    dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ/pd-6"
    status="modified">
  </storageLocalDisk>
 </outConfig>
</configConfMo>
```

Enabling JBOD

Request:

```
<configConfMo cookie='cookiecutter'
 dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ">
  <inConfig>
    <storageController dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ"
        adminAction="enable-jbod"/>
  </inConfig>
</configConfMo>
```

Response:

```
<configConfMo cookie="1424337949/70faa56b-6e0f-1f6e-8004-91fabb1b0ff4"
     response="yes" errorCode="2003"
 dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ"
     invocationResult="unidentified-fail" errorDescr="Operation failed.
     Changing JBOD mode is not allowed on controller">
</configConfMo>
```

Disabling JBOD

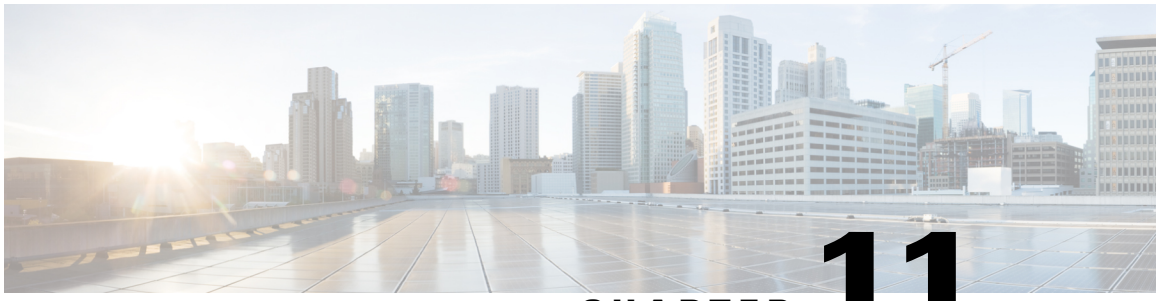
Request:

```
<configConfMo cookie='1424337112/0c6e0e69-6d0f-1f6d-8003-91fabb1b0ff4'
 dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ">
    <inConfig>
     <storageController
        dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ" adminAction="disable-jbod"/>

    </inConfig>
</configConfMo
```

Response:

```
<configConfMo cookie="1424337112/0c6e0e69-6d0f-1f6d-8003-91fabb1b0ff4"
     response="yes" errorCode="2003"
 dn="sys/chassis-1/server-1/board/storage-SAS-SLOT-MEZZ"
     invocationResult="unidentified-fail" errorDescr="Operation failed.
     Changing JBOD mode is not allowed on controller">
</configConfMo>
```



CHAPTER 11

Configuring Communication Services

This chapter includes the following sections:

Configuring Communication Services

The examples in this section show how to use the Cisco IMC XML API to retrieve and configure communication services. Each example shows the XML API request followed by the response from Cisco IMC.

This section includes the following examples:

- Retrieving the Configured SNMP Engine ID, on page 104

Retrieving Server Communication Services Details

Request:

```
<configResolveClass cookie="1418471980/706e0794-180a-1a18-8007-91fabb1b0ff4"
classId="commSvcRack" inHierarchical="true"></configResolveClass>
```

Response:

```
<configResolveClass cookie="1418471980/706e0794-180a-1a18-8007-91fabb1b0ff4"
response="yes" classId="commSvcRack">
  <outConfigs>
    <commSvcRack dn="sys/chassis-1/server-1/svc-ext">
      <commKvm rn="kvm-svc" adminState="enabled" totalSessions="4" activeSessions="1"
       port="2068" encryptionState="enabled" localVideoState="enabled"/>
      <commVMedia rn="vmedia-svc" adminState="enabled" activeSessions="0"
       encryptionState="disabled"/>
      <commIpmiLan rn="ipmi-lan-svc" adminState="enabled" priv="admin"
       key="0000000000000000000000000000000000000000"/>
    </commSvcRack>
  </outConfigs>
</configResolveClass>
```

Retrieving Server Communication Services Details Using DN

Request:

```
<configResolveDn dn='sys/chassis-1/server-1/svc-ext' inHierarchical='true'>
cookie='1418471980/706e0794-180a-1a18-8007-91fabb1b0ff4' </configResolveDn>
```

Response:

```
<configResolveDn cookie="1418471980/706e0794-180a-1a18-8007-91fabb1b0ff4"
  response="yes" dn="sys/chassis-1/server-1/svc-ext">
  <outConfig>
    <commSvcRack dn="sys/chassis-1/server-1/svc-ext">
      <commKvm rn="kvm-svc" adminState="enabled" totalSessions="4" activeSessions="1"
       port="2068" encryptionState="enabled" localVideoState="enabled"/>
      <commVMedia rn="vmedia-svc" adminState="enabled" activeSessions="0"
      encryptionState="disabled"/>
      <commIpmiLan rn="ipmi-lan-svc" adminState="enabled" priv="admin"
       key="0000000000000000000000000000000000000000"/>
    </commSvcRack>
  </outConfig>
</configResolveDn>
```

Retrieving IPMI over LAN Details for CMCs

Request:

```
<configResolveClass cookie="0948771774/afdfe352-e75e-1ee7-8002-e87877e2cff4"
classId="commEpIpmiLan" inHierarchical="true"></configResolveClass>
```

Response:

```
<configResolveClass cookie="0948771774/afdfe352-e75e-1ee7-8002-e87877e2cff4"
 response="yes" classId="commEpIpmiLan">
  <outConfigs>
    <commEpIpmiLan dn="sys/chassis-1/slot-1/cmc-ipmi-lan" adminState="enabled"
     priv="admin" key="1111100000000000000000000000000000000000"/>
    <commEpIpmiLan dn="sys/chassis-1/slot-2/cmc-ipmi-lan" adminState="enabled"
     priv="admin" key="0000000000000000000000000000000000000000"/>
  </outConfigs>
</configResolveClass>
```

Retrieving IPMI over LAN Details for CMC Using DN

Request:

```
<configResolveDn cookie="0948771774/afdfe352-e75e-1ee7-8002-e87877e2cff4"
  response="yes" dn="sys/chassis-1/slot-2/cmc-ipmi-lan">
```

Response:

```
<outConfig>
    <commEpIpmiLan dn="sys/chassis-1/slot-2/cmc-ipmi-lan" adminState="enabled"
     priv="admin"  key="00000000000000000000000000000000000000000000"/>
  </outConfig>
</configResolveDn>
```

Configuring IPMI over LAN for CMCs

Request:

```
<configConfMo cookie='0948774645/804a211e-e85e-1ee8-8002-e87877e2cff4'
dn='sys/chassis-1/slot-2/cmc-ipmi-lan'>
<inConfig>
  <commEpIpmiLan adminState='enabled'
   key="1111100000000000000000000000000000022222"/>
   </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/slot-2/cmc-ipmi-lan"
cookie="0948774645/804a211e-e85e-1ee8-8002-e87877e2cff4" response="yes">
  <outConfig>
    <commEpIpmiLan dn="sys/chassis-1/slot-2/cmc-ipmi-lan" adminState="enabled"
    priv="admin" key="1111100000000000000000000000000000022222"
    status="modified"/>
  </outConfig>
</configConfMo>
```

Retrieving IPMI over LAN Details for BMCs

Request:

```
<configResolveClass cookie="0948774645/804a211e-e85e-1ee8-8002-e87877e2cff4"
classId="commIpmiLan" inHierarchical="true"></configResolveClass>
```

Response:

```
<configResolveClass cookie="0948774645/804a211e-e85e-1ee8-8002-e87877e2cff4"
response="yes" classId="commIpmiLan">
  <outConfigs>
    <commIpmiLan dn="sys/chassis-1/server-1/svc-ext/ipmi-lan-svc"
    adminState="disabled"
     priv="admin" key="00000000000000000000000000000000000000000000"/>
  </outConfigs>
```

Retrieving IPMI over LAN Details Using DN for BMC

Request:

```
<configResolveDn cookie="0948775445/40885b7d-e85e-1ee8-8004-e87877e2cff4"
dn='sys/chassis-1/server-1/svc-ext/ipmi-lan-svc' inHierarchical="true"/>
```

Response:

```
<configResolveDn cookie="0948775445/40885b7d-e85e-1ee8-8004-e87877e2cff4"
   response="yes" dn="sys/chassis-1/server-1/svc-ext/ipmi-lan-svc">
  <outConfig>
    <commIpmiLan dn="sys/chassis-1/server-1/svc-ext/ipmi-lan-svc"
     adminState="disabled" priv="admin" key="00000000000000000000000000000000000000000000"/>
  </outConfig>
</configResolveDn>
```

Configuring IPMI over LAN Details for BMC

Request:

```
<configConfMo cookie='0948775445/40885b7d-e85e-1ee8-8004-e87877e2cff4'
 dn='sys/chassis-1/server-1/svc-ext/ipmi-lan-svc'>
 <inConfig>
   <commIpmiLan adminState='enabled' key="33333000000000000000000000000000044444"/>
 </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/server-1/svc-ext/ipmi-lan-svc"
   cookie="0948775445/40885b7d-e85e-1ee8-8004-e87877e2cff4" response="yes">
  <outConfig>
    <commIpmiLan dn="sys/chassis-1/server-1/svc-ext/ipmi-lan-svc"
    adminState="enabled" priv="admin"
    key="33333000000000000000000000000000044444" status="modified"/>
  </outConfig>
```

Retrieving Chassis Communication Services Details

Request:

```
<configResolveClass cookie="1256517526/630efb57-cc76-16cc-8009-aaec921b0ff4"
response="yes" classId="commSvcEp">
```

Response:

```
<outConfigs>
<commSvcEp dn="sys/svc-ext">
  <commHttp rn="http-svc" adminState="enabled" descr="HTTP Service" name="http"
   port="80" redirectState="enabled" sessionTimeout="1800" maximumSessions="4"
   activeSessions="0" proto="tcp"/>
   <commHttps rn="https-svc" adminState="enabled" descr="Secure HTTP Service"
    name="https" port="443" sessionTimeout="1800" maximumSessions="4" activeSession
    ="0" proto="tcp"/>
   <commSsh rn="ssh-svc" adminState="enabled" descr="Secure Shell Server" name="ssh"
    port="22" sessionTimeout="1800" maximumSessions="4" activeSessions="1"
    proto="tcp"/>
   <commSnmp rn="snmp-svc" descr="SNMP Service" name="snmp" adminState="enabled"
    port="161" proto="udp" community="public" trapCommunity="" com2Sec="full"
    sysContact="who@where" sysLocation="unknown">
   <commSnmpTrap id="1" adminState="disabled" version="v3" notificationType="traps"
    port="162" hostname="0.0.0.0" user="unknown" rn="snmp-trap-1"/>
   <commSnmpTrap id="2" adminState="disabled" version="v3" notificationType="traps"
    port="162" hostname="0.0.0.0" user="unknown" rn="snmp-trap-2"/>
    .
    .
   <commSnmpUser id="1" name="" securityLevel="" auth="" authPwd="" privacy=""
    privacyPwd="" rn="snmpv3-user-1"/>
   <commSnmpUser id="2" name="" securityLevel="" auth="" authPwd="" privacy=""
    privacyPwd="" rn="snmpv3-user-2"/>
    .
    .
   <commSyslog rn="syslog" adminState="enabled" descr="Syslog Service" name="syslog"
    port="514" proto="tcp" localSeverity="debug" remoteSeverity="warning">
    commSyslogClient adminState="disabled" hostname="0.0.0.0" name="primary" port="514"
    rn="client-primary"/>
   <commSyslogClient adminState="disabled" hostname="0.0.0.0" name="secondary" port="514"
    rn="client-secondary"/>
  </commSyslog>
   <commNtpProvider rn="ntp-svc" description="Network Time Protocol" ntpEnable="no"
    ntpServer1=" " ntpServer2=" " ntpServer3=" " ntpServer4=" "/>
    </commSvcEp>
  </outConfigs>
</configResolveClass>
```

Retrieving NTP Using DN

Request:

```
<configResolveDn cookie="1256518508/a00063d2-cc76-16cc-800a-aaec921b0ff4"
dn='sys/svc-ext/ssh-svc' inHierarchical="false"/>
```

Response:

```
<configResolveDn cookie="1256518508/a00063d2-cc76-16cc-800a-aaec921b0ff4"
response="yes" dn="sys/svc-ext/ssh-svc">
  <outConfig>
    <commSsh dn="sys/svc-ext/ssh-svc" adminState="enabled" descr="Secure Shell Server"
     name="ssh" port="22" sessionTimeout="1800" maximumSessions="4"
     activeSessions="1" proto="tcp"/>
  </outConfig>
</configResolveDn>
```

Configuring NTP Servers

Request:

```
<configConfMo cookie="1256520649/5b73df6a-cd76-16cd-800b-aaec921b0ff4"
dn="sys/svc-ext/ntp-svc">
  <inConfig>
   <commNtpProvider dn="sys/svc-ext/ntp-svc" ntpEnable="no" ntpServer1="10.104.255.217"
    ntpServer2="ntpserver2.cisco.com" ntpServer3="ntp.globalcomp.net"
    ntpServer4="ntpserver4.cisco.com"/>
    </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/svc-ext/ntp-svc"
cookie="1256520649/5b73df6a-cd76-16cd-800b-aaec921b0ff4" response="yes">
<outConfig>
 <commNtpProvider dn="sys/svc-ext/ntp-svc" description="Network Time Protocol"
  ntpEnable="no" ntpServer1="10.104.255.217" ntpServer2="ntpserver2.cisco.com"
  ntpServer3="ntp.globalcomp.net" ntpServer4="ntpserver4.cisco.com" status="modified"/>
 </outConfig>
</configConfMo>
```

Retrieving SNMP Details

Request:

```
<configResolveClass cookie="1256522119/b7fff8fc-cc76-16cc-800c-aaec921b0ff4" inH
ierarchical="true" classId="aaaLdap"/>
```

Response:

```
<configResolveClass cookie="1256522119/b7fff8fc-cc76-16cc-800c-aaec921b0ff4"
response="yes" classId="aaaLdap">
 <outConfigs>
  <aaaLdap dn="sys/ldap-ext" adminState="disabled" basedn="" domain=""
   filter="sAMAccountName" attribute="CiscoAvPair" timeout="60" encryption="enabled"
   locateDirectoryUsingDNS="no" dnsDomainSource="extracted-domain" dnsSearchDomain=""
   dnsSearchForest="" ldapServer1="" ldapServerPort1="389" ldapServer2=""
   ldapServerPort2="389" ldapServer3="" ldapServerPort3="389" ldapServer4=""
   ldapServerPort4="3268" ldapServer5="" ldapServerPort5="3268" ldapServer6=""
   ldapServerPort6="3268" bindMethod="login-credentials" bindDn="" password=""
   groupAuth="disabled" groupAttribute="memberOf">
   .
   .
   rn="rolegroup-27"/> <aaaLdapRoleGroup id="28" name="" domain="" role=""
   adminAction="no-op"   rn="rolegroup-28"/>
    </aaaLdap>
  </outConfigs>
</configResolveClass>
```

Retrieving SNMP Details Using DN

Request:

```
<configResolveDn cookie="1256523667/8f16501a-ce76-16ce-8012-aaec921b0ff4"
  inHierarchical="true" dn='sys/svc-ext/snmp-svc'/>
```

Response:

```
<configResolveDn cookie="1256523667/8f16501a-ce76-16ce-8012-aaec921b0ff4"
 response="yes" dn="sys/svc-ext/snmp-svc">
  <outConfig>
    <commSnmp dn="sys/svc-ext/snmp-svc" descr="SNMP Service" name="snmp"
       adminState="enabled" port="161" proto="udp"
      community="public" trapCommunity=""
      com2Sec="full" sysContact="who@where" sysLocation="unknown">
      <commSnmpTrap id="1" adminState="disabled" version="v3"
       notificationType="traps" port="162" hostname="0.0.0.0"
       user="unknown" rn="snmp-trap-1"/>
      <commSnmpTrap id="2" adminState="disabled" version="v3"
       notificationType="traps" port="162" hostname="0.0.0.0"
       user="unknown" rn="snmp-trap-2"/>
      <commSnmpTrap id="3" adminState="disabled" version="v3"
       notificationType="traps" port="162" hostname="0.0.0.0"
       user="unknown" rn="snmp-trap-3"/>
      <commSnmpTrap id="4" adminState="disabled"
       version="v3" notificationType="traps"
       port="162" hostname="0.0.0.0" user="unknown" rn="snmp-trap-4"/>
      <commSnmpTrap id="5" adminState="disabled" version="v3"
       notificationType="traps" port="162"
       hostname="0.0.0.0" user="unknown" rn="snmp-trap-5"/>
      <commSnmpTrap id="6" adminState="disabled"
       version="v3" notificationType="traps" port="162"
       hostname="0.0.0.0" user="unknown" rn="snmp-trap-6"/>
      <commSnmpTrap id="7" adminState="disabled" version="v3"
       notificationType="traps" port="162" hostname="0.0.0.0"
       user="unknown" rn="snmp-trap-7"/>
      <commSnmpTrap id="8" adminState="disabled"
       version="v3" notificationType="traps"
       port="162" hostname="0.0.0.0" user="unknown" rn="snmp-trap-8"/>
      <commSnmpTrap id="9" adminState="disabled"
       version="v3" notificationType="traps" port="162"
       hostname="0.0.0.0" user="unknown" rn="snmp-trap-9"/>
      <commSnmpTrap id="10" adminState="disabled" version="v3"
       notificationType="traps" port="162" hostname="0.0.0.0"
       user="unknown" rn="snmp-trap-10"/>
      <commSnmpTrap id="11" adminState="disabled" version="v3"
       notificationType="traps" port="162" hostname="0.0.0.0"
        user="unknown" rn="snmp-trap-11"/>
      <commSnmpTrap id="12" adminState="disabled" version="v3"
       notificationType="traps" port="162" hostname="0.0.0.0"
      user="unknown" rn="snmp-trap-12"/>
      <commSnmpTrap id="13" adminState="disabled"
       version="v3" notificationType="traps" port="162"
        hostname="0.0.0.0" user="unknown" rn="snmp-trap-13"/>
...
...
...
    </commSnmp>
  </outConfig>
</configResolveDn>
```

Enabling SNMP and Configuring Parameters

Request:

```
<configConfMo cookie="0961471706/5f867d7e-746a-1a74-8004-77812a02b884"
inHierarchical="false" dn="sys/svc-ext/snmp-svc">
```

```
<inConfig>
 <commSnmp dn="sys/svc-ext/snmp-svc" adminState="enabled" community="Top$ecrt"
  trapCommunity="Dar$Good" com2Sec="full" sysContact="snmpadmin@cisco" sysLocation="San
Jose"/>
</inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/svc-ext/snmp-svc"
 cookie="0961471706/5f867d7e-746a-1a74-8004-77812a02b884" response="yes">
<outConfig>
 <commSnmp dn="sys/svc-ext/snmp-svc" descr="SNMP Service" name="snmp"
  adminState="enabled" port="161" proto="udp" community="Top$ecrt"
  trapCommunity="Dar$Good" com2Sec="full" sysContact="snmpadmin@cisco"
  sysLocation="San Jose" engineId="80 00 1F 88 80 64 DA CE 16 39 4E E5 0F"
  status="modified" >
 </commSnmp>
</outConfig>
</configConfMo>
```

Configuring SNMP Users

Request:

```
<configConfMo cookie="0961472433/0c3bd92d-756a-1a75-8005-77812a02b884"
inHierarchical="false" dn="sys/svc-ext/snmp-svc/snmpv3-user-1">
<inConfig>
  <commSnmpUser dn="sys/svc-ext/snmp-svc/snmpv3-user-1" id="1" name="admin"
   auth="MD5" authPwd="cisco123" privacy="DES" securityLevel="authpriv"
   privacyPwd="ciscoucs"/>
  </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/svc-ext/snmp-svc/snmpv3-user-1"
cookie="0961472433/0c3bd92d-756a-1a75-8005-77812a02b884" response="yes">
<outConfig>
 <commSnmpUser id="1" name="admin" securityLevel="authpriv" auth="MD5"
  authPwd="" privacy="DES" privacyPwd="" adminAction="no-op"
  dn="sys/svc-ext/snmp-svc/snmpv3-user-1" status="modified" >
 </commSnmpUser>
 </outConfig>
</configConfMo>
```

Configuring SNMP Traps

Request:

```
<configConfMo cookie="0961472980/a220c9eb-746a-1a74-8006-77812a02b884"
inHierarchical="false" dn="sys/svc-ext/snmp-svc/snmp-trap-1">
 <inConfig>
  <commSnmpTrap dn="sys/svc-ext/snmp-svc/snmp-trap-1" id="1" adminState="enabled"
   hostname="10.106.146.33" notificationType="traps" version="v2c"/>
 </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/svc-ext/snmp-svc/snmp-trap-1"
cookie="0961472980/a220c9eb-746a-1a74-8006-77812a02b884" response="yes">
<outConfig>
 <commSnmpTrap id="1" adminState="enabled" version="v2c" notificationType="traps"
  port="162" hostname="10.106.146.33" user="unknown" adminAction="no-op"
  dn="sys/svc-ext/snmp-svc/snmp-trap-1" status="modified" >
 </commSnmpTrap>
</outConfig>
</configConfMo>
```

Setting Up an SNMP Engine ID

Request:

```
<configConfMo cookie="1468979398/1fd17a6c-3808-1808-8002-ac988322e470"
  inHierarchical="false" dn="sys/svc-ext/snmp-svc">
   <inConfig>
       <commSnmp dn="sys/svc-ext/snmp-svc" adminState="enabled"
    engineIdKey="Test_Engine_Key">
       </commSnmp>
   </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/svc-ext/snmp-svc"
cookie="1468979398/1fd17a6c-3808-1808-8002-ac988322e470"
  response="yes">
   <outConfig>
        <commSnmp dn="sys/svc-ext/snmp-svc" descr="SNMP Service"
         name="snmp" adminState="enabled" port="161" proto="udp" community=""
         trapCommunity="" com2Sec="disabled" sysContact="who@where" sysLocation="unknown"
         engineId="" engineIdKey="Test_Engine_Key" status="modified" >
       </commSnmp>
   </outConfig>
</configConfMo>
```

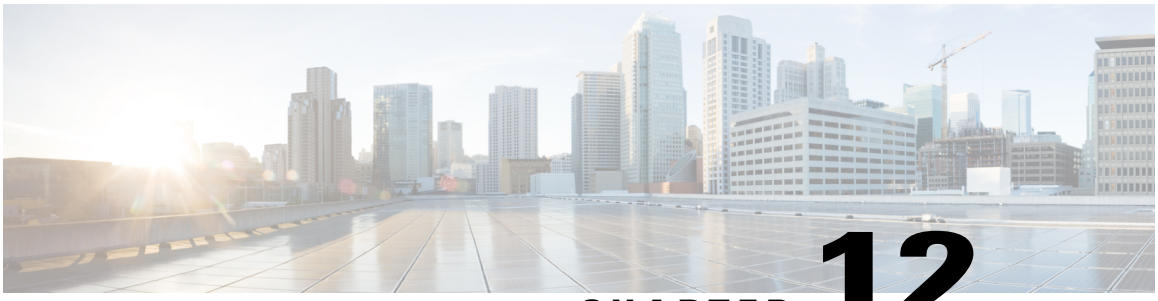
Retrieving the Configured SNMP Engine ID

Request:

```
<configResolveClass cookie="1468979398/1fd17a6c-3808-1808-8002-ac988322e470"
   inHierarchical="false" classId="commSnmp"/>
```

Response:

```
<configResolveClass cookie="1468979398/1fd17a6c-3808-1808-8002-ac988322e470"
   response="yes" classId="commSnmp">
    <outConfigs>
        <commSnmp dn="sys/svc-ext/snmp-svc" descr="SNMP Service"
         name="snmp" adminState="enabled" port="161" proto="udp" community=""
         trapCommunity="" com2Sec="disabled" sysContact="who@where" sysLocation="unknown"
         engineId="80 00 00 09 04 54 65 73 74 5F 45 6E 67 69 6E 65 5F 4B 65 79"
          engineIdKey="Test_Engine_Key">
        </commSnmp>
    </outConfigs>
</configResolveClass>
```



CHAPTER 12

Configuring Platform Event Filters

his chapter includes the following sections:

Configuring Platform Event Filters

The examples in this section show how to use the Cisco IMC XML API to retrieve and perform configuring platform event filters tasks. Each example shows the XML API request followed by the response from Cisco IMC.

This section includes the following examples:

Enabling Platform Event Filters

Request:

```
<configConfMo cookie="1472923819/9b885334-9e3b-1b9e-8003-be18652a6ca4"
dn="sys/chassis-1/event-management">
<inConfig>
    <eventManagement dn="sys/chassis-1/event-management" adminState="enabled"/>
  </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/event-management"
  cookie="1472923819/9b885334-9e3b-1b9e-8003-be18652a6ca4" response="yes">
  <outConfig>
    <eventManagement dn="sys/chassis-1/event-management" adminState="enabled"
     adminAction="no-op" status="modified"/>
  </outConfig>
</configConfMo>
```

Disabling Platform Event Filters

Request:

```
<configConfMo cookie="1472923819/9b885334-9e3b-1b9e-8003-be18652a6ca4"
dn="sys/chassis-1/event-management">
<inConfig>
    <eventManagement dn="sys/chassis-1/event-management" adminState="disabled"/>
  </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/event-management"
  cookie="1472923819/9b885334-9e3b-1b9e-8003-be18652a6ca4" response="yes">
  <outConfig>
    <eventManagement dn="sys/chassis-1/event-management" adminState="disabled"
     adminAction="no-op" status="modified"/>
  </outConfig>
</configConfMo>
```

Resetting Platform Event Filters

Request:

```
<configConfMo cookie="1472923819/9b885334-9e3b-1b9e-8003-be18652a6ca4"
dn="sys/chassis-1/event-management">
  <inConfig>
    <eventManagement dn="sys/chassis-1/event-management"
     adminAction="reset-event-filters"/>
  </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/event-management"
 cookie="1472923819/9b885334-9e3b-1b9e-8003-be18652a6ca4" response="yes">
  <outConfig>
    <eventManagement dn="sys/chassis-1/event-management" adminState="enabled"
     adminAction="no-op" status="modified"/>
  </outConfig>
</configConfMo>
```

Configuring Platform Event Filters

Request:

```
<configConfMo cookie="1400975065/29082126-fa2e-1a2e-8005-bcbec261b284"
dn="sys/rack-unit-1/event-management" inHierarchical="true">
  <inConfig>
    <eventManagement dn="sys/rack-unit-1/event-management">
      <platformEventFilters id="1" action="power-off" rn="pef-1"/>
      <platformEventFilters id="2" action="power-off" rn="pef-2"/>
      <platformEventFilters id="3" action="power-cycle" rn="pef-3"/>
      <platformEventFilters id="4" action="power-cycle" rn="pef-4"/>
    </eventManagement>
  </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/rack-unit-1/event-management"
cookie="1400975065/29082126-fa2e-1a2e-8005-bcbec261b284" response="yes">
  <outConfig>
    <eventManagement dn="sys/rack-unit-1/event-management" adminState="enabled"
      adminAction="no-op" status="modified">
      <platformEventFilters id="1" event="Temperature Critical Assert Filter"
      action="power-off" rn="pef-1" status="modified"/>
      <platformEventFilters id="2" event="Voltage Critical Assert Filter"
       action="power-off" rn="pef-2" status="modified"/>
      <platformEventFilters id="3" event="Current Critical Assert Filter"
       action="power-cycle" rn="pef-3" status="modified"/>
      <platformEventFilters id="4" event="Fan Critical Assert Filter"
       action="power-cycle" rn="pef-4" status="modified"/>
      <platformEventFilters id="5" event="Processor Assert Filter" action="none"
       rn="pef-5" status="modified"/>
      <platformEventFilters id="6" event="Power Supply Critical Assert Filter"
       action="none" rn="pef-6" status="modified"/>
      <platformEventFilters id="7" event="Memory Critical Assert Filter" action="none"
       rn="pef-7" status="modified"/>
    </eventManagement>
  </outConfig>
</configConfMo>
```

Configuring Platform Event Filters

Request:

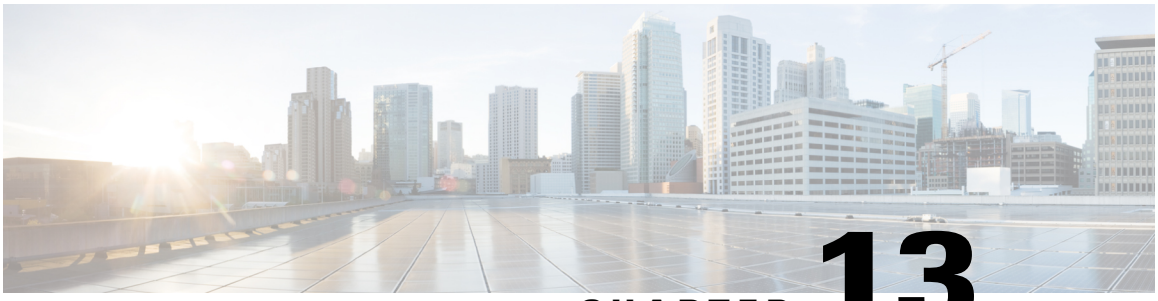
```
<configConfMo cookie="1472923819/9b885334-9e3b-1b9e-8003-be18652a6ca4"
dn="sys/chassis-1/event-management/pef-2">
  <inConfig>
    <platformEventFilters id="2" action="power-cycle" rn="pef-2"/>
  </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/event-management/pef-2"
cookie="1472923819/9b885334-9e3b-1b9e-8003-be18652a6ca4" response="yes">
  <outConfig>
    <platformEventFilters id="2" event="Voltage Warning Assert Filter" action="power-cycle"

     dn="sys/chassis-1/event-management/pef-2" status="modified"/>
  </outConfig>
</configConfMo>
```

Configuring Platform Event Filters



CHAPTER 13

Managing Certificates

This chapter includes the following sections:

Managing Server Certificates

The examples in this section show how to use the Cisco IMC XML API to manage server certificates. Each example shows the XML API request followed by the response from Cisco IMC.

This section includes the following examples:

Retrieving Certificate Details

Request:

```
<configResolveClass cookie="1448761796/eb8a8234-25a4-15a4-8002-9a6ae7925a88"
classId="currentCertificate" inHierarchical="false"></configResolveClass>
```

Response:

```
<configResolveClass cookie="1448761796/eb8a8234-25a4-15a4-8002-9a6ae7925a88"
response="yes" classId="currentCertificate">
 <outConfigs>
  <currentCertificate dn="sys/cert-mgmt/curr-cert" serialNumber="C764DC592E154539"
   countryCode="US" state="California" locality="San Jose" organization="cisco"
   organizationalUnit="cisco" commonName="cisco" issuerCountryCode="US"
   issuerState="California" issuerLocality="San Jose" issuerOrganization="cisco"
   issuerOrganizationalUnit="cisco" issuerCommonName="cisco"
   validFrom="Nov 20 05:11:22 2015 GMT" validTo="Nov 17 05:11:22 2025 GMT"/>
  </outConfigs>
</configResolveClass>
```

Generating Certificate Signing Request

Request:

```
<configConfMo cookie='1448761796/eb8a8234-25a4-15a4-8002-9a6ae7925a88'
dn="sys/cert-mgmt/gen-csr-req" inHierarchical="false">
 <inConfig>
  <generateCertificateSigningRequest commonName="cisco" organization="cisco"
   organizationalUnit="cisco" locality="San Jose" state="California" countryCode="United
States"
   protocol="ftp" remoteServer="10.10.10.10" user="user" pwd="cisco123"
   remoteFile="/tmp/host.csr" dn="sys/cert-mgmt/gen-csr-req"/>
 </inConfig>
</configConfMo>
```

Response:

```
<configResolveClass cookie="1448761796/eb8a8234-25a4-15a4-8002-9a6ae7925a88"
response="yes" classId="currentCertificate">
 <outConfigs>
  <currentCertificate dn="sys/cert-mgmt/curr-cert" serialNumber="C764DC592E154539"
   countryCode="US" state="California" locality="San Jose" organization="cisco"
   organizationalUnit="cisco" commonName="cisco" issuerCountryCode="US"
   issuerState="California" issuerLocality="San Jose" issuerOrganization="cisco"
   issuerOrganizationalUnit="cisco" issuerCommonName="cisco"
   validFrom="Nov 20 05:11:22 2015 GMT" validTo="Nov 17 05:11:22 2025 GMT"/>
 </outConfigs>
</configResolveClass>
```

Retrieving the Status of a Certificate Signing Request

Request:

```
<configResolveClass cookie="1448761796/eb8a8234-25a4-15a4-8002-9a6ae7925a88"
 classId="generateCertificateSigningRequest" inHierarchical="false">
</configResolveClass>
```

Response:

```
<configResolveClass cookie="1448761796/eb8a8234-25a4-15a4-8002-9a6ae7925a88"
response="yes" classId="generateCertificateSigningRequest">
 <outConfigs>
  <generateCertificateSigningRequest dn="sys/cert-mgmt/gen-csr-req"
   commonName="Common Name" organization="Organization" organizationalUnit="Organizational

   Unit" locality="Locality" state="State" countryCode="Country Code" email="Email Address"

   selfSigned="no" protocol="none" remoteServer="" remoteFile="" user="" pwd=""
   csrStatus="Completed CSR"/>
 </outConfigs>
</configResolveClass>
```

Generating Self-Signed Certificate

Request:

```
<configConfMo cookie='1448761796/eb8a8234-25a4-15a4-8002-9a6ae7925a88'
dn="sys/cert-mgmt/gen-csr-req" inHierarchical="false">
 <inConfig>
 <generateCertificateSigningRequest commonName="cisco" organization="cisco"
  organizationalUnit="cisco" locality="Banglore" state="KARNATAKA"
  countryCode="India" dn="sys/cert-mgmt/gen-csr-req" selfSigned="yes"/>
 </inConfig>
</configConfMo>
```

Response:

```
<configConfMocookie="1448761796/eb8a8234-25a4-15a4-8002-9a6ae7925a88" response="yes"
dn="sys/cert-mgmt/gen-csr-req">
 <outConfig>
  <generateCertificateSigningRequest dn="sys/cert-mgmt/gen-csr-req" commonName="Common Name"

   organization="Organization" organizationalUnit="Organizational Unit" locality="Locality"

   state="State" countryCode="Country Code" email="Email Address" selfSigned="no"
   protocol="none" remoteServer="" remoteFile="" user="" pwd=""
   csrStatus="Completed CSR" status="modified"/>
 </outConfig>
</configConfMo>
```

Uploading a Signed Certificate

Request:

```
<configConfMo cookie='1448762867/b32d6bdd-25a4-15a4-8002-9a6ae7925a88'
dn="sys/cert-mgmt/upload-cert" inHierarchical="false">
<inConfig>
 <uploadCertificate adminAction="remote-cert-upload" protocol="sftp" user="user"
  remoteServer="10.10.10.10" remoteFile="/tmp/xmlTest.crt" pwd="cisco123"
  dn="sys/cert-mgmt/upload-cert"/>
 </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/cert-mgmt/upload-cert"
cookie="1448762867/b32d6bdd-25a4-15a4-8002-9a6ae7925a88"
response="yes">
 <outConfig>
  <uploadCertificate dn="sys/cert-mgmt/upload-cert" adminAction="no-op" protocol="none"
   remoteServer="" remoteFile="" user="" pwd="" certificateContent="Certificate Content"
   status="modified"/>
  </outConfig>
</configConfMo>
```

Managing LDAP Certificates

The examples in this section show how to use the Cisco IMC XML API to retrieve and perform LDAP certificate management tasks. Each example shows the XML API request followed by the response from Cisco IMC.

This section includes the following examples:

Enabling Binding of an LDAP CA Certificate

Request:

```
<configConfMo cookie='1457742601/2dd5f334-2dcf-1dcf-8005-515545067ff0'
dn='sys/ldap-ext/ldap-ca-cert-mgmt'>
<inConfig>
 <ldapCACertificateManagement dn='sys/ldap-ext/ldap-ca-cert-mgmt'
  bindingCertificate='enabled'/>
 </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/ldap-ext/ldap-ca-cert-mgmt"
cookie="1470032930/13a3ed5e-38fd-18fd-800f-ad7c7d74a254" response="yes">
 <outConfig>
  <ldapCACertificateManagement dn="sys/ldap-ext/ldap-ca-cert-mgmt"
  description="LDAP CA Certificate Management"
  bindingCertificate="enabled" status="modified"  >
  </ldapCACertificateManagement>
 </outConfig>
</configConfMo>
```

Disabling Binding of CA Certificate

Request:

```
<configConfMo cookie='1457742601/2dd5f334-2dcf-1dcf-8005-515545067ff0'
dn='sys/ldap-ext/ldap-ca-cert-mgmt'>
 <inConfig>
  <ldapCACertificateManagement
  dn='sys/ldap-ext/ldap-ca-cert-mgmt' bindingCertificate='disabled'/>
 </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/ldap-ext/ldap-ca-cert-mgmt"
cookie="1470032930/13a3ed5e-38fd-18fd-800f-ad7c7d74a254" response="yes">
 <outConfig>
  <ldapCACertificateManagement dn="sys/ldap-ext/ldap-ca-cert-mgmt"
  description="LDAP CA Certificate Management"
  bindingCertificate="disabled" status="modified" >
  </ldapCACertificateManagement>
 </outConfig>
</configConfMo>
```

Downloading LDAP CA Certificate using TFTP Protocol

Request:

```
<configConfMo cookie='1470032930/13a3ed5e-38fd-18fd-800f-ad7c7d74a254'
dn='sys/ldap-ext/ldap-ca-cert-mgmt/ldap-ca-cert-download' inHierarchical='false'>
 <inConfig>
  <downloadLdapCACertificate protocol='tftp' remoteServer='10.10.10.10'
   remoteFile='new_com_chain.cer' dn='sys/ldap-ext/ldap-ca-cert-mgmt/ldap-ca-cert-download'/>

 </inConfig>
</configConfMo>
```

TFTP used in the preceding example is the default protocol. You can also download the LDAP CA certificate using the other available protocols such as the FTP, SFTP, SCP and HTTP.

Response:

```
<configConfMo dn="sys/ldap-ext/ldap-ca-cert-mgmt/ldap-ca-cert-download"
cookie="1470032930/13a3ed5e-38fd-18fd-800f-ad7c7d74a254" response="yes">
 <outConfig>
  <downloadLdapCACertificate dn="sys/ldap-ext/ldap-ca-cert-mgmt/ldap-ca-cert-download"
  protocol="none" remoteServer="" remoteFile="" user="" pwd=""
 downloadStatus="COMPLETED"  downloadProgress="100%" status="modified" >
  </downloadLdapCACertificate>
 </outConfig>
</configConfMo>
```

Exporting LDAP CA Certificate

Request:

```
<configConfMo cookie='1463635956/27a0d4af-332c-132c-8004-9206a0395bfc'
dn='sys/ldap-ext/ldap-ca-cert-mgmt/ldap-ca-cert-export' inHierarchical='false'>
 <inConfig>
  <exportLdapCACertificate protocol='tftp' remoteServer='10.10.10.10'
  remoteFile='fasfsaf.csr'dn='sys/ldap-ext/ldap-ca-cert-mgmt/ldap-ca-cert-export'/>
 </inConfig>
</configConfMo>"
```

Response:

```
<configConfMo dn="sys/ldap-ext/ldap-ca-cert-mgmt/ldap-ca-cert-export"
cookie="1470032930/13a3ed5e-38fd-18fd-800f-ad7c7d74a254" response="yes">
 <outConfig>
  <exportLdapCACertificate dn="sys/ldap-ext/ldap-ca-cert-mgmt/ldap-ca-cert-export"
   protocol="none" remoteServer="" remoteFile="" user="" pwd=""
   exportStatus="COMPLETED" exportProgress="100%" status="modified" >
  </exportLdapCACertificate>
 </outConfig>
</configConfMo>
```

TFTP used in the preceding example is the default protocol. You can also download the LDAP CA certificate using the other available protocols such as the FTP, SFTP, SCP and HTTP.

Testing LDAP Binding

Request:

```
<configConfMo cookie='1470032930/13a3ed5e-38fd-18fd-800f-ad7c7d74a254'
dn='sys/ldap-ext/ldap-ca-cert-mgmt/ldap-ca-cert' inHierarchical='false'>
 <inConfig>
  <ldapCACertificate adminAction='test-ldap-binding' user='user' pwd='Test123'
  dn='sys/ldap-ext/ldap-ca-cert-mgmt/ldap-ca-cert'/>
 </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/ldap-ext/ldap-ca-cert-mgmt/ldap-ca-cert"
cookie="1470032930/13a3ed5e-38fd-18fd-800f-ad7c7d74a254" response="yes">
 <outConfig>
  <ldapCACertificate dn="sys/ldap-ext/ldap-ca-cert-mgmt/ldap-ca-cert"
  adminAction="" user="" pwd="" status="modified" >
  </ldapCACertificate>
 </outConfig>
</configConfMo>
```

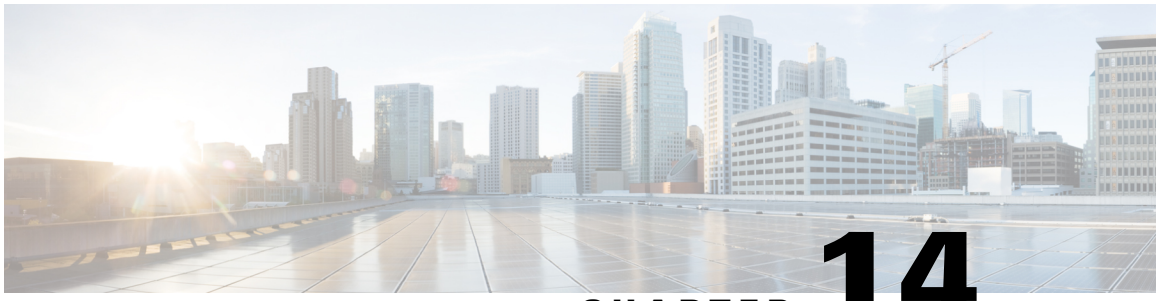
Deleting LDAP CA Certificate

Request:

```
<configConfMo cookie='1457746251/9ec8b64d-2dd0-1dd0-8008-515545067ff0'
dn='sys/ldap-ext/ldap-ca-cert-mgmt/ldap-ca-cert' inHierarchical='false'>
 <inConfig>
  <ldapCACertificate adminAction='delete-ca-certificate'
  dn='sys/ldap-ext/ldap-ca-cert-mgmt/ldap-ca-cert'/>
 </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/ldap-ext/ldap-ca-cert-mgmt/ldap-ca-cert"
cookie="1470032930/13a3ed5e-38fd-18fd-800f-ad7c7d74a254" response="yes">
 <outConfig>
  <ldapCACertificate dn="sys/ldap-ext/ldap-ca-cert-mgmt/ldap-ca-cert"
   adminAction="" user="" pwd="" status="modified" >
  </ldapCACertificate>
 </outConfig>
</configConfMo>
```



CHAPTER 14

Managing Firmware

This chapter includes the following sections:

Managing BIOS Firmware

The examples in this section show how to use the Cisco IMC XML API to retrieve and update BIOS firmware. Each example shows the XML API request followed by the response from Cisco IMC.

This section includes the following examples:

Retrieving BIOS Running Firmware Version

Request:

```
<configResolveDn cookie="1418810141/bed473ba-660a-1a66-800e-91fabb1b0ff4"
dn='sys/chassis-1/server-1/bios/fw-boot-loader' inHierarchical="false"/>
```

Response:

```
<configResolveDn cookie="1418810141/bed473ba-660a-1a66-800e-91fabb1b0ff4"
response="yes" dn="sys/chassis-1/server-1/bios/fw-boot-loader">
  <outConfig>
```

```
    <firmwareRunning dn="sys/chassis-1/server-1/bios/fw-boot-loader"
    description="BIOS currently running firmware version"
    deployment="boot-loader" type="blade-bios" version="C3260M4.2.0.6.17.081720142325"/>
  </outConfig>
</configResolveDn>
```

Retrieving BIOS Backup Firmware Version

Request:

```
<configResolveDn cookie="1418810141/bed473ba-660a-1a66-800e-91fabb1b0ff4"
dn='sys/chassis-1/server-1/bios/fw-updatable' inHierarchical="false"/>
```

Response:

```
<configResolveDn cookie="1418810141/bed473ba-660a-1a66-800e-91fabb1b0ff4"
response="yes" dn="sys/chassis-1/server-1/bios/fw-updatable">
  <outConfig>
    <firmwareUpdatable dn="sys/chassis-1/server-1/bios/fw-updatable"
     description="BIOS firmware version" adminState="triggered"
     deployment="primary" operState="success"
     version="C3260M4.2.0.6.17.081720142325" protocol="none"
     remoteServer="" remotePath="" user="" pwd="" progress="Done,
     OK " type="blade-bios"/>
  </outConfig>
</configResolveDn>
```

Retrieving BIOS Startup Firmware Version

Request:

```
<configResolveDn cookie="1418810756/8f046609-670a-1a67-800f-91fabb1b0ff4"
dn='sys/chassis-1/server-1/bios/fw-boot-def' inHierarchical="true"/>
```

Response:

```
<configResolveDn cookie="1418810756/8f046609-670a-1a67-800f-91fabb1b0ff4"
response="yes" dn="sys/chassis-1/server-1/bios/fw-boot-def">
  <outConfig>
    <firmwareBootDefinition dn="sys/chassis-1/server-1/bios/fw-boot-def"
     type="blade-bios">
    <firmwareBootUnit rn="bootunit-combined" description="BIOS startup
     firmware version" adminState="triggered" image="running"
     resetOnActivate="no" type="combined" version="C3260M4.2.0.6.17.081720142325"/>
    </firmwareBootDefinition>
  </outConfig>
</configResolveDn>
```

Upgrading BIOS Firmware Using TFTP

The following example shows how to upgrade BIOS firmware using TFTP protocol. You can also upgrade the firmware using the following other protocols:

- FTP
- SFTP
- SCP
- HTTP

Request:

```
<configConfMo cookie='1418813107/6f2f499a-670a-1a67-8002-91fabb1b0ff4'
dn='sys/chassis-1/server-1/bios/fw-updatable'>
<inConfig>
  <firmwareUpdatable dn='sys/chassis-1/server-1/bios/fw-updatable'
   adminState='trigger' protocol='tftp' type='blade-bios'
   remoteServer='10.106.27.149' remotePath='sriparim/C3260M4-BIOS-2-0-5-1.cap'/>
```

```
</inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/server-1/bios/fw-updatable"
cookie="1418813107/6f2f499a-670a-1a67-8002-91fabb1b0ff4" response="yes">
  <outConfig>
    <firmwareUpdatable dn="sys/chassis-1/server-1/bios/fw-updatable"
     description="BIOS firmware version" adminState="triggered"
    deployment="primary" operState="failed" version="C3260M4.2.0.1.17.081720142325"
    protocol="none" remoteServer="" remotePath="" user="" pwd=""
    progress=" Update In Progress " type="blade-bios" status="modified"/>
  </outConfig>
</configConfMo>
```

Verifying the Progress of the Upgrade

Request:

```
<configResolveDn cookie="1418813107/6f2f499a-670a-1a67-8002-91fabb1b0ff4"
response="yes" dn="sys/chassis-1/server-1/bios/fw-updatable">
```

Response:

```
  <outConfig>
 <firmwareUpdatable dn="sys/chassis-1/server-1/bios/fw-updatable"
   description="BIOS firmware version" adminState="triggered"
   deployment="primary" operState="success"
   version="C3160M3.2.0.1.17.081720142325" protocol="none"
   remoteServer="" remotePath="" user="" pwd=""
   progress="Done, OK " type="blade-bios"/>
  </outConfig>
</configResolveDn>
```

Activating the Installed BIOS Firmware

Important Host must be powered off while activating BIOS.

Request:

```
<configConfMo dn="sys/chassis-1/server-1/bios/fw-boot-def/bootunit-combined"
cookie="1418814643/d6338300-680a-1a68-8003-91fabb1b0ff4" response="yes">
<inConfig>
 <firmwareBootUnit dn='sys/chassis-1/server-1/bios/fw-boot-def/bootunit-combined'
  adminState='trigger' image='backup' resetOnActivate='yes' />
 </inConfig>
```

Response:

```
<configConfMo dn="sys/chassis-1/server-1/bios/fw-boot-def/bootunit-combined"
cookie="1418814643/d6338300-680a-1a68-8003-91fabb1b0ff4" response="yes">
 <outConfig>
  <firmwareBootUnit dn="sys/chassis-1/server-1/bios/fw-boot-def/bootunit-combined"
   description="BIOS startup firmware version" adminState="triggered"
   image="running" resetOnActivate="no" type="combined"
   version="C3260M4.2.0.6.17.081720142325" status="modified"/>
  </outConfig>
</configConfMo>
```

Managing the BMC Firmware

The examples in this section show how to use the Cisco IMC XML API to retrieve and update BMC firmware. Each example shows the XML API request followed by the response from Cisco IMC.

This section includes the following examples:

Retrieving the BMC Firmware Running Version

Request:

```
<configResolveDn cookie="1421228528/664ed95d-9a0c-1c9a-8002-91fabb1b0ff4"
dn='sys/chassis-1/server-1/mgmt/fw-system' inHierarchical="false"/>
```

Response:

```
 <configResolveDn cookie="1421228528/664ed95d-9a0c-1c9a-8002-91fabb1b0ff4"
response="yes" dn="sys/chassis-1/server-1/mgmt/fw-system">
  <outConfig>
    <firmwareRunning dn="sys/chassis-1/server-1/mgmt/fw-system"
     description="Cisco IMC currently running firmware version"
     deployment="system" type="blade-controller" version="2.0(6.4)"/>
  </outConfig>
</configResolveDn>
```

Updating BMC Firmware

The following example shows how to upgrade BMC firmware using the TFTP protocol. You can also upgrade the firmware using the following other protocols:

- FTP
- SFTP
- SCP
- HTTP

Request:

```
<configConfMo cookie='1421233678/d85dfcf5-9a0c-1c9a-8002-91fabb1b0ff4'
dn='sys/chassis-1/server-1/mgmt/fw-updatable'>
<inConfig>
  <firmwareUpdatable dn='sys/chassis-1/server-1/mgmt/fw-updatable'
   adminState='trigger' protocol='tftp' type='blade-controller'
   remoteServer='10.106.27.149' remotePath='hlkABSG/upd-pkg-c3260-cimc.full.2.0.6.15.bin'/>
</inConfig>
</configConfMo>
```

Response:

```
<outConfig>
 <firmwareUpdatable dn="sys/chassis-1/slot-1/mgmt/fw-updatable"
  description="System IO Controller backup firmware version"
  adminState="triggered" deployment="backup" operState="updating"
  version="0.0(4.r202878)" protocol="none" remoteServer="" remotePath=""
  user="" pwd="" progress="5" type="sioc"/>
  </outConfig>
</configResolveDn>
```

Activating the Firmware

Request:

```
<configConfMo cookie='1421240125/e86a101e-9d0c-1c9d-8002-91fabb1b0ff4'
dn='sys/chassis-1/server-1/mgmt/fw-boot-def/bootunit-combined'>
```

```
 <inConfig>
  <firmwareBootUnit dn='sys/chassis-1/server-1/mgmt/fw-boot-def/bootunit-combined'
   adminState='trigger' image='backup' resetOnActivate='yes' />
 </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/server-1/mgmt/fw-boot-def/bootunit-combined"
cookie="1421240125/e86a101e-9d0c-1c9d-8002-91fabb1b0ff4" response="yes">
<outConfig>
 <firmwareBootUnit dn="sys/chassis-1/server-1/mgmt/fw-boot-def/bootunit-combined"
  description="Cisco IMC startup firmware version" adminState="triggered"
  image="running" resetOnActivate="no" type="combined" version="2.0(6.4)"
  status="modified"/>
  </outConfig>
</configConfMo>
```

Managing the CMC Firmware

The examples in this section show how to use the Cisco IMC XML API to retrieve and update CMC firmware. Each example shows the XML API request followed by the response from Cisco IMC.

This section includes the following examples:

- Updating CMC Firmware, on page 119
- Verifying the Progress of the Upgrade, on page 120
- Activating the Firmware, on page 120

Updating CMC Firmware

The following example shows how to upgrade CMC firmware using TFTP protocol. You can also upgrade the firmware using the following other protocols:

- FTP
- SFTP
- SCP
- HTTP

Request:

```
<configConfMo cookie='1420794510/983be121-350c-1c35-8004-91fabb1b0ff4'
dn='sys/chassis-1/slot-1/mgmt/fw-updatable'>
<inConfig>
 <firmwareUpdatable dn='sys/chassis-1/slot-1/mgmt/fw-updatable'
  adminState='trigger' protocol='tftp' type='sioc' remoteServer='10.106.27.149'
  remotePath='sriparim/chassis.img'/>
</inConfig>
</configConfMo>
```

Response:

```
<outConfig>
 <firmwareUpdatable dn="sys/chassis-1/slot-1/mgmt/fw-updatable"
  description="System IO Controller backup firmware version"
  adminState="triggered" deployment="backup" operState="updating"
  version="0.0(4.r202878)" protocol="none" remoteServer="" remotePath=""
  user="" pwd="" progress="5" type="sioc"/>
  </outConfig>
</configResolveDn>
```

Verifying the Progress of the Upgrade

Request:

```
<configResolveDn cookie="1420807158/bf2e8355-380c-1c38-8002-91fabb1b0ff4"
dn='sys/chassis-1/slot-1/mgmt/fw-updatable' inHierarchical="false"/>
```

Response:

```
  <configResolveDn cookie="1420807158/bf2e8355-380c-1c38-8002-91fabb1b0ff4"
response="yes" dn="sys/chassis-1/slot-1/mgmt/fw-updatable">
 <outConfig>
  <firmwareUpdatable dn="sys/chassis-1/slot-1/mgmt/fw-updatable"
   description="System IO Controller backup firmware version" adminState="triggered"
   deployment="backup" operState="ready" version="0.0(4.r205775)" protocol="none"
   remoteServer="" remotePath="" user="" pwd="" progress="Success" type="sioc"/>
  </outConfig>
</configResolveDn>
```

Activating the Firmware

Request:

```
<configConfMo cookie='1420810495/57090030-390c-1c39-8002-91fabb1b0ff4'
 dn='sys/chassis-1/slot-1/mgmt/fw-boot-def/bootunit-combined'>
  <inConfig>
   <firmwareBootUnit dn='sys/chassis-1/slot-1/mgmt/fw-boot-def/bootunit-combined'
    adminState='trigger' image='backup' resetOnActivate='yes' />
   </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/slot-1/mgmt/fw-boot-def/bootunit-combined"
cookie="1420810495/57090030-390c-1c39-8002-91fabb1b0ff4" response="yes">
 <outConfig>
  <firmwareBootUnit dn="sys/chassis-1/slot-1/mgmt/fw-boot-def/bootunit-combined"
   description="System Input Out Controller startup firmware version"
   adminState="triggered" image="running" resetOnActivate="no" type="combined"
   version="0.0(4.r204950)" status="modified"/>
  </outConfig>
</configConfMo>
```

Managing SAS Expander Firmware

The examples in this section show how to use the Cisco IMC XML API to retrieve and update SAS expander firmware. Each example shows the XML API request followed by the response from Cisco IMC.

This section includes the following examples:

Retrieving SAS Expander Firmware Details

Request:

```
<configResolveClass cookie="1428566069/191135cc-4613-1346-8003-e87877e2cff4"
inHierarchical="true" classId="storageExpander"/>
```

Response:

```
<configResolveClass cookie="1428566069/191135cc-4613-1346-8003-e87877e2cff4"
response="yes" classId="storageExpander">
 <outConfigs>
    <storageExpander id="1" description="Storage controller - SAS Expander"
     dn="sys/chassis-1/expander-sas-1">
      <firmwareRunning rn="fw-system"
      description="SAS Expander currently running firmware version"
      deployment="system" type="expander-sas" version="04.08.01_B052"/>
     <firmwareUpdatable rn="fw-updatable" description="SAS Expander backup firmware version"

       adminState="triggered" deployment="backup" operState="ready"
       version="04.08.01_B052" protocol="none" remoteServer="" remotePath=""
       user="" pwd="" progress="Success" type="expander-sas"/>
      <firmwareBootDefinition rn="fw-boot-def" type="expander-sas">
      <firmwareBootUnit rn="bootunit-combined"
       description="SAS Expander startup firmware version" adminState="triggered"
       image="running" resetOnActivate="no" type="combined" version="04.08.01_B052"/>
      </firmwareBootDefinition>
    </storageExpander>
    <storageExpander id="2" description="Storage controller - SAS Expander"
      dn="sys/chassis-1/expander-sas-2">
     <firmwareRunning rn="fw-system"
      description="SAS Expander currently running firmware
      version" deployment="system" type="expander-sas" version="04.08.01_B052"/>
     <firmwareUpdatable rn="fw-updatable"
      description="SAS Expander backup firmware version" adminState="triggered"
      deployment="backup" operState="ready" version="04.08.01_B052" protocol="none"
     remoteServer="" remotePath="" user="" pwd="" progress="Success" type="expander-sas"/>

     <firmwareBootDefinition rn="fw-boot-def" type="expander-sas">
     <firmwareBootUnit rn="bootunit-combined"
      description="SAS Expander startup firmware version" adminState="triggered"
      image="running" resetOnActivate="no" type="combined" version="04.08.01_B052"/>
      </firmwareBootDefinition>
    </storageExpander>
  </outConfigs>
</configResolveClass>
```

Retrieving Running SAS Expander Firmware Version

Request:

```
<configResolveDn cookie="1428566846/738bfd6c-4613-1346-8004-e87877e2cff4"
dn='sys/chassis-1/expander-sas-2/fw-system' inHierarchical="true"/>
```

Response:

```
<configResolveDn cookie="1428566846/738bfd6c-4613-1346-8004-e87877e2cff4"
response="yes" dn="sys/chassis-1/expander-sas-2/fw-system">
<outConfig>
 <firmwareRunning dn="sys/chassis-1/expander-sas-2/fw-system"
  description="SAS Expander currently running firmware version"
  deployment="system" type="expander-sas" version="04.08.01_B052"/>
 </outConfig>
</configResolveDn>
```

Retrieving SAS Expander Firmware Backup Version

Request:

```
<configResolveDn cookie="1428566846/738bfd6c-4613-1346-8004-e87877e2cff4"
dn='sys/chassis-1/expander-sas-1/fw-updatable' inHierarchical="true"/>
```

Response:

```
<configResolveDn cookie="1428566846/738bfd6c-4613-1346-8004-e87877e2cff4"
response="yes" dn="sys/chassis-1/expander-sas-1/fw-updatable">
  <outConfig>
   <firmwareUpdatable dn="sys/chassis-1/expander-sas-1/fw-updatable"
    description="SAS Expander backup firmware version" adminState="triggered"
    deployment="backup" operState="ready" version="04.08.01_B052" protocol="none"
    remoteServer="" remotePath="" user="" pwd="" progress="Success"
    type="expander-sas"/>
  </outConfig>
</configResolveDn>
```

Retrieving SAS Expander Boot Definition Firmware Version

Request:

```
<configResolveDn cookie="1428566846/738bfd6c-4613-1346-8004-e87877e2cff4"
dn='sys/chassis-1/expander-sas-2/fw-boot-def' inHierarchical="true"/>
```

Response:

```
<configResolveDn cookie="1428566846/738bfd6c-4613-1346-8004-e87877e2cff4"
response="yes" dn="sys/chassis-1/expander-sas-2/fw-boot-def">
  <outConfig>
    <firmwareBootDefinition dn="sys/chassis-1/expander-sas-2/fw-boot-def"
     type="expander-sas">
    <firmwareBootUnit rn="bootunit-combined" description="SAS Expander
     startup firmware version" adminState="triggered" image="running"
     resetOnActivate="no" type="combined" version="04.08.01_B052"/>
    </firmwareBootDefinition>
  </outConfig>
</configResolveDn>
```

Upgrading SAS Expander Firmware on SAS Expander 1

The following example shows how to upgrade SAS expander firmware using TFTP protocol. You can also upgrade the firmware using the following other protocols:

- FTP
- SFTP
- SCP
- HTTP

Request:

```
<configConfMo cookie='1428567858/e6b2ac69-4613-1346-8005-e87877e2cff4'
dn='sys/chassis-1/expander-sas-1/fw-updatable'>
<inConfig>
  <firmwareUpdatable dn='sys/chassis-1/expander-sas-1/fw-updatable'
   adminState='trigger' protocol='tftp' type='expander-sas'
   remoteServer='10.197.125.101' remotePath='xxxxxx/fw_image-B052.bin'/>
 </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/expander-sas-1/fw-updatable"
cookie="1428567858/e6b2ac69-4613-1346-8005-e87877e2cff4" response="yes">
 <outConfig>
   <firmwareUpdatable dn="sys/chassis-1/expander-sas-1/fw-updatable"
    description="SAS Expander backup firmware version" adminState="triggered"
    deployment="backup" operState="updating" version="04.08.01_B052"
    protocol="none" remoteServer="" remotePath="" user="" pwd="" progress="0"
    type="expander-sas" status="modified"/>
  </outConfig>
</configConfMo>
```

Verifying the Status of SAS Expander Firmware Upgrade

Request:

```
<configResolveDn cookie="1428566846/738bfd6c-4613-1346-8004-e87877e2cff4"
dn='sys/chassis-1/expander-sas-1/fw-updatable' inHierarchical="true"/>
```

Response:

```
<configResolveDn cookie="1428567858/e6b2ac69-4613-1346-8005-e87877e2cff4"
response="yes" dn="sys/chassis-1/expander-sas-1/fw-updatable">
 <outConfig>
  <firmwareUpdatable dn="sys/chassis-1/expander-sas-1/fw-updatable"
   description="SAS Expander backup firmware version" adminState="triggered"
   deployment="backup" operState="ready" version="04.08.01_B052" protocol="none"
   remoteServer="" remotePath="" user="" pwd="" progress="Success"
   type="expander-sas"/>
  </outConfig>
</configResolveDn>
```

Activating the Firmware

Request:

```
<configConfMo cookie='1418823162/649dcd2f-6a0a-1a6a-8008-91fabb1b0ff4'
dn='sys/chassis-1/sasexp-1/fw-boot-def/bootunit-combined'>
  <inConfig>
    <firmwareBootUnit dn='sys/chassis-1/sasexp-1/fw-boot-def/bootunit-combined'
     adminState='trigger' image='backup' resetOnActivate='yes' />
    </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/expander-sas-1/fw-boot-def/bootunit-combined"
cookie="1428568472/4e033c69-4713-1347-8006-e87877e2cff4" response="yes">
 <outConfig>
  <firmwareBootUnit dn="sys/chassis-1/expander-sas-1/fw-boot-def/bootunit-combined"
   description="SAS Expander startup firmware version" adminState="triggered"
   image="running" resetOnActivate="no" type="combined" version="04.08.01_B052"
   status="modified"/>
  </outConfig>
</configConfMo>
```

Managing the Network Adapter Firmware

The examples in this section show how to use the Cisco IMC XML API to retrieve and update the network adapter firmware. Each example shows the XML API request followed by the response from Cisco IMC.

This section includes the following examples:

Retrieving the Adapter Firmware Running Version

Request:

```
<configResolveDn cookie='1431644026/649b1615-1316-1613-8003-d27f77e2cff4'
inHierarchical='false' dn='sys/chassis-1/slot-1/shared-io-module/mgmt/fw-system' />
```

Response:

```
<configResolveDn cookie="1431644026/649b1615-1316-1613-8003-d27f77e2cff4"
 response="yes" dn="sys/chassis-1/slot-1/shared-io-module/mgmt/fw-system">
<outConfig>
 <firmwareRunning dn="sys/chassis-1/slot-1/shared-io-module/mgmt/fw-system"
  description="Cisco VIC adapter currently running firmware version" deployment="system"
  type="adaptor" version="4.0(300.22)"/>
 </outConfig>
</configResolveDn>
```

Retrieving the Adapter Firmware Backup Version

Request:

```
<configResolveDn cookie='1431644026/649b1615-1316-1613-8003-d27f77e2cff4'
inHierarchical='false' dn='sys/chassis-1/slot-2/shared-io-module/mgmt/fw-updatable' />
```

Response:

```
<configResolveDn cookie="1431644026/649b1615-1316-1613-8003-d27f77e2cff4"
response="yes" dn="sys/chassis-1/slot-2/shared-io-module/mgmt/fw-updatable">
<outConfig>
<firmwareUpdatable dn="sys/chassis-1/slot-2/shared-io-module/mgmt/fw-updatable"
  description="Cisco VIC adapter backup firmware version" adminState="triggered"
  deployment="backup" version="4.0(300.20)" protocol="none" remoteServer=""
  remotePath="" user="" pwd="" progress="Stage: No  operation (0%), Status: Idle,
  Error: No error" type="adaptor"/>
</outConfig>
</configResolveDn>
```

Retrieving the Adapter Firmware Boot Loader Version

Request:

```
<configResolveDn cookie='1431644026/649b1615-1316-1613-8003-d27f77e2cff4'
inHierarchical='false' dn='sys/chassis-1/slot-1/shared-io-module/mgmt/fw-boot-loader'/>
```

Response:

```
<configResolveDn cookie="1431644026/649b1615-1316-1613-8003-d27f77e2cff4"
response="yes" dn="sys/chassis-1/slot-1/shared-io-module/mgmt/fw-boot-loader">
<outConfig>
 <firmwareRunning dn="sys/chassis-1/slot-1/shared-io-module/mgmt/fw-boot-loader"
  description="Cisco VIC adapter currently running boot loader firmware version"
  deployment="boot-loader" type="adaptor" version="4.0(300.22)"/>
 </outConfig>
</configResolveDn>
```

Updating the Adapter Firmware

The following example shows how to upgrade the adapter firmware using the SCP protocol. You can also upgrade the firmware using the following other protocols:

- FTP
- SFTP
- TFTP
- HTTP

Request:

```
<configConfMo cookie="0955187490/0b79d656-bd64-14bd-8002-82f92a02b884"
dn="sys/chassis-1/slot-1/shared-io-module/mgmt/fw-updatable">
<inConfig>
 <firmwareUpdatable dn="sys/chassis-1/slot-1/shared-io-module/mgmt/fw-updatable"
  adminState="trigger" protocol="scp" user="sriparim" pwd="password"
  remoteServer="10.197.125.101" remotePath="/scp/kjhadhsaim/cruz-latest.bin"
  type="adaptor"/>
 </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/slot-1/shared-io-module/mgmt/fw-updatable"
cookie="0955187490/0b79d656-bd64-14bd-8002-82f92a02b884" response="yes">
 <outConfig>
  <firmwareUpdatable dn="sys/chassis-1/slot-1/shared-io-module/mgmt/fw-updatable"
   description="Cisco VIC adapter backup firmware version" adminState="triggered"
   deployment="backup" version="4.0(300.27)" protocol="none" remoteServer="" remotePath=""

   user="" pwd="" progress="Stage: Transferring (15%), Status: Update in progress,
   Error: No error" type="adaptor" status="modified"/>
 </outConfig>
</configConfMo>
```

Viewing the Progress of the Update

Request:

```
<configResolveDn cookie='0955187490/0b79d656-bd64-14bd-8002-82f92a02b884'
inHierarchical='false' dn='sys/chassis-1/slot-1/shared-io-module/mgmt/fw-updatable' />
```

Response:

```
<configResolveDn cookie="0955187490/0b79d656-bd64-14bd-8002-82f92a02b884"
response="yes" dn="sys/chassis-1/slot-1/shared-io-module/mgmt/fw-updatable">
<outConfig>
 <firmwareUpdatable dn="sys/chassis-1/slot-1/shared-io-module/mgmt/fw-updatable"
  description="Cisco VIC adapter backup firmware version" adminState="triggered"
  deployment="backup" protocol="none" remoteServer="" remotePath="" user="" pwd=""
  progress="Stage: Writing (71%), Status: Update in progress, Error: No error"
  type="adaptor"/>
 </outConfig>
</configResolveDn>
```

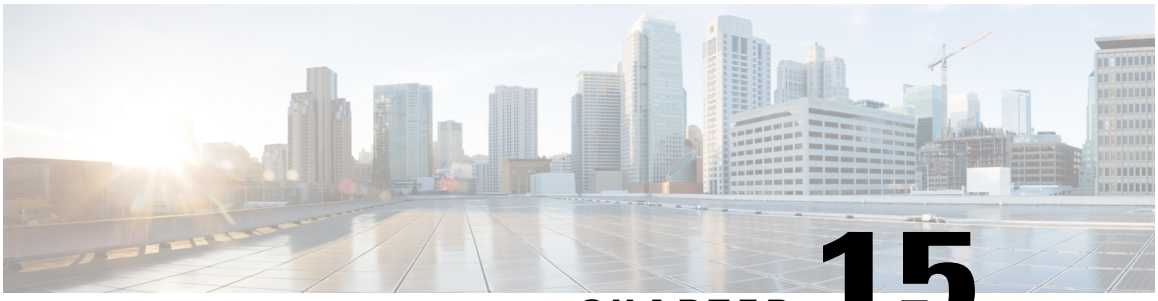
Activating the Firmware

Request:

```
<configConfMo cookie='0955187490/0b79d656-bd64-14bd-8002-82f92a02b884'
dn='sys/chassis-1/slot-1/shared-io-module/mgmt/fw-boot-def/bootunit-combined'>
<inConfig>
 <firmwareBootUnit
  dn='sys/chassis-1/slot-1/shared-io-module/mgmt/fw-boot-def/bootunit-combined'
  adminState='trigger' image='backup' />
 </inConfig>
</configConfMo>
/>
```

Response:

```
<configConfMo dn="sys/chassis-1/slot-1/shared-io-module/mgmt/fw-boot-def/bootunit-
combined" cookie="0955187490/0b79d656-bd64-14bd-8002-82f92a02b884" response="yes">
<outConfig>
 <firmwareBootUnit
  dn="sys/chassis-1/slot-1/shared-io-module/mgmt/fw-boot-def/bootunit-combined"
  description="Cisco VIC adapter startup firmware version" adminState="triggered"
  image="running" resetOnActivate="no" type="combined"
  version="4.0(300.42)" status="modified"/>
 </outConfig>
</configConfMo>
```



CHAPTER 15

Server Utilities

This chapter includes the following sections:

Importing and Exporting Tech Support Logs and Configurations

The examples in this section show how to use the Cisco IMC XML API to import and export technical support logs and configurations. Each example shows the XML API request followed by the response from Cisco IMC.

This section includes the following examples:

Retrieving Tech Support Log

Request:

```
<configResolveClass cookie="1420620154/3f9d68ff-0b0c-1c0b-8002-91fabb1b0ff4"
inHierarchical="false" classId="sysdebugTechSupportExport"/>
```

Response:

```
<configResolveClass cookie="1420620154/3f9d68ff-0b0c-1c0b-8002-91fabb1b0ff4"
response="yes" classId="sysdebugTechSupportExport">
<outConfigs>
   <sysdebugTechSupportExport dn="sys/chassis-1/tech-support" adminState="disabled"
    hostname="10.106.27.149" protocol="TFTP" remoteFile="sriparim-20150104-185635.tar.gz"
    user="" component="all" pwd="" fsmStageDescr="error" fsmProgr="0"
    fsmStatus="tftp-upload-error"/>
```

```
  </outConfigs>
</configResolveClass>
```

Retrieving Tech Support Log Using DN

Request:

```
<configResolveDn cookie="1420620154/3f9d68ff-0b0c-1c0b-8002-91fabb1b0ff4"
dn='sys/chassis-1/tech-support' inHierarchical="false"/>
```

Response:

```
<configResolveDn cookie="1420620154/3f9d68ff-0b0c-1c0b-8002-91fabb1b0ff4"
response="yes" dn="sys/chassis-1/tech-support">
 <outConfig>
  <sysdebugTechSupportExport dn="sys/chassis-1/tech-support" adminState="disabled"
   hostname="10.106.27.149" protocol="TFTP" remoteFile="sriparim-20150104-185635.tar.gz"
   user="" component="all" pwd="" fsmStageDescr="error" fsmProgr="0"
   fsmStatus="tftp-upload-error"/>
  </outConfig>
</configResolveDn>
```

Generating and Exporting Tech Support Data for CMC 1

You also can generate the export the technical support data for the following components:

- only all—For all components.
- cmc2—For CMC 2.
- cimc1—For Cisco IMC server 1.
- cimc2—For Cisco IMC server 2.

The following example shows how to export the technical support data using SCP protocol. You can also upgrade the firmware using the following other protocols:

- FTP
- SFTP
- TFTP
- HTTP

Request:

```
<configConfMo cookie="1420636265/1e591b55-100c-1c10-8005-91fabb1b0ff4"
inHierarchical="false" dn="sys/chassis-1/tech-support">
<inConfig>
  <sysdebugTechSupportExport dn="sys/chassis-1/tech-support" adminState="enabled"
   remoteFile="/home/sriparim/c3260_techsupport.tgz" user="sriparim" pwd="password"
   protocol="scp" component="cmc1" hostname="10.106.27.149"/>
  </inConfig>
```

Response:

```
<configConfMo dn="sys/chassis-1/tech-support"
<outConfig>
 <sysdebugTechSupportExport dn="sys/chassis-1/tech-support"
  adminState="disabled" hostname="10.106.27.149" protocol="scp"
  remoteFile="/home/sriparim/c3260_techsupport.tgz" user="sriparim"
  component="cmc1" pwd="" fsmStageDescr="completed" fsmProgr="100"
  fsmStatus="success" status="modified"/>
 </outConfig>
</configConfMo>
```

Verifying the Status of the Tech Support Data Export

Request:

```
<configResolveDn cookie="1420620154/3f9d68ff-0b0c-1c0b-8002-91fabb1b0ff4"
dn='sys/chassis-1/tech-support' inHierarchical="false"/>
```

Response:

```
<configResolveDn cookie="1420636265/1e591b55-100c-1c10-8005-91fabb1b0ff4"
response="yes" dn="sys/chassis-1/tech-support">
<outConfig>
  <sysdebugTechSupportExport dn="sys/chassis-1/tech-support" adminState="enabled"
   hostname="10.106.27.149" protocol="scp" remoteFile="/home/sriparim/c3260_techsupport.tgz"

   user="sriparim" component="cmc1" pwd="" fsmStageDescr="collecting" fsmProgr="20"
   fsmStatus="exporting"/>
  </outConfig>
</configResolveDn>
```

Exporting CMC Configuration

The following example shows how to export the CMC configuration using the SCP protocol. You can also upgrade the firmware using the following other protocols:

- FTP
- SFTP
- TFTP
- HTTP

Request:

```
<configConfMo cookie="1420636265/1e591b55-100c-1c10-8005-91fabb1b0ff4"
 inHierarchical="false" dn="sys/chassis-1/export-config">
<inConfig>
  mgmtBackup dn="sys/chassis-1/export-config" adminState="enabled"
  entity="CMC" proto="scp" user="uldeshmu" pwd="cisco123" passphrase="abcdefgh"
  hostname="10.104.255.217" remoteFile="/home/uldeshmu/c250_config_export.cfg"/>
</inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/export-config" response="yes">
<outConfig>
  <mgmtBackup dn="sys/chassis-1/export-config"  entity=""  adminState="enabled"
   fsmStageDescr="Collecting configuration data"  fsmRmtInvErrCode=""
   fsmRmtInvErrDescr="NONE"   fsmDescr="export-config" proto="none"   hostname=""
   remoteFile=""   user=""   pwd="" passphrase=""  status="modified" >
  </mgmtBackup>
 </outConfig>
</configConfMo>
```

Exporting BMC Configuration

The following example shows how to export the BMC configuration using the SCP protocol. You can also upgrade the firmware using the following other protocols:

- FTP
- SFTP
- TFTP
- HTTP

Request:

```
<configConfMo cookie='cookiecutter' inHierarchical="false"
dn="sys/chassis-1/server-1/exporter-config">
<inConfig>
  <mgmtBackupServer dn="sys/chassis-1/server-1/exporter-config"
  adminState="enabled" entity="CIMC1" proto="scp" user="uldeshmu"
  pwd="cisco123" hostname="10.104.255.217"
  remoteFile="/home/uldeshmu/c250_config_export_cimc.cfg"/>
</inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/server-1/exporter-config"
cookie="1423754785/ec33d7e1-e50e-1ee5-8002-d27f77e2cff4" response="yes">
<outConfig>
  <mgmtBackupServer dn="sys/chassis-1/server-1/exporter-config" entity=""
   adminState="disabled"  fsmStageDescr="COMPLETED"  fsmRmtInvErrCode="100"
   fsmRmtInvErrDescr="NONE"  fsmDescr="export-config"  proto="none" hostname=""
   remoteFile="" user="" pwd="" status="modified" >
  </mgmtBackupServer>
 </outConfig>
</configConfMo>
```

Resetting to Defaults and Reboot Tasks

The examples in this section show how to use the Cisco IMC XML API to reset to factory defaults and reboot the CMC. Each example shows the XML API request followed by the response from Cisco IMC.

This section includes the following examples:

- Resetting BMC to Defaults, on page 130
- Resetting CMC to Defaults, on page 131
- Rebooting CMC, on page 131

Resetting BMC to Defaults

Request:

```
<configConfMo cookie='1421736489/9cda9dd4-0f0d-1d0f-8002-91fabb1b0ff4'
dn='sys/chassis-1/server-1'>
<inConfig>
  <computeServerNode dn='sys/chassis-1/server-1' adminPower='bmc-reset-default'/>
</inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/server-1"
cookie="1421736489/9cda9dd4-0f0d-1d0f-8002-91fabb1b0ff4" response="yes">
<outConfig>
  <computeServerNode serverId="1" adminPower="policy" availableMemory="262144"
   model="UCSC-C3X60-SVRNB" memorySpeed="1866" name="UCS C3260"
   numOfCores="24" numOfCoresEnabled="24" numOfCpus="2" numOfThreads="48"
   operPower="on" originalUuid="1C0C4600-671D-4B53-A06E-590CD0FEBC85"
   presence="equipped" serial="FCH1821JAV4" totalMemory="262144"
   usrLbl="" uuid="1C0C4600-671D-4B53-A06E-590CD0FEBC85"
   vendor="Cisco Systems Inc" dn="sys/chassis-1/server-1" status="modified"/>
  </outConfig>
</configConfMo>
```

Resetting CMC to Defaults

Request:

```
<configConfMo cookie='1421750857/333d9d59-130d-1d13-8002-91fabb1b0ff4'
dn='sys/chassis-1/slot-1'>
<inConfig>
  equipmentIOCard dn='sys/chassis-1/slot-1' adminPower='cmc-reset-default'/>
</inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/slot-1"
cookie="1421750857/333d9d59-130d-1d13-8002-91fabb1b0ff4" response="yes">
  <outConfig>
    <equipmentIOCard id="1" description="SIOC - System Input Output Controller"
     adminPower="policy" dn="sys/chassis-1/slot-1" status="modified"/>
  </outConfig>
</configConfMo>
```

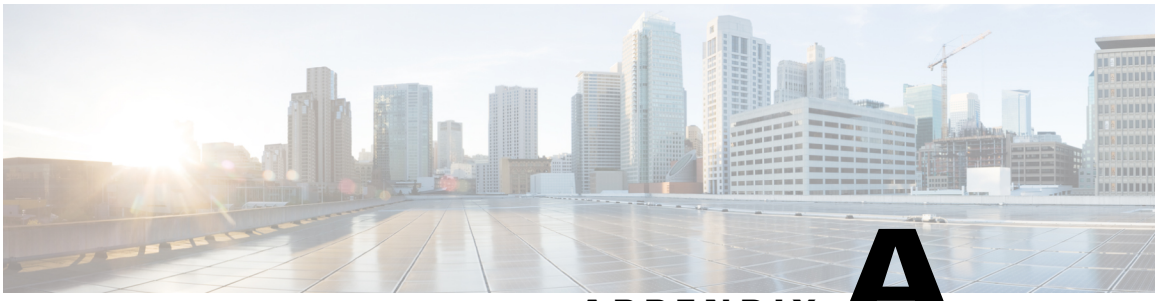
Rebooting CMC

Request:

```
<configConfMo cookie='1421832041/8808b674-260d-1d26-8002-91fabb1b0ff4'
dn='sys/chassis-1/slot-1'>
 <inConfig>
   equipmentIOCard dn='sys/chassis-1/slot-1' adminPower='cmc-reboot'/>
  /inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1/slot-1"
cookie="1421832041/8808b674-260d-1d26-8002-91fabb1b0ff4" response="yes">
<outConfig>
  <equipmentIOCard id="1" description="SIOC - System Input Output Controller"
   adminPower="policy" dn="sys/chassis-1/slot-1" status="modified"/>
  </outConfig>
</configConfMo>
```



APPENDIX A

Notes on Using the configConfMo Method

This appendix includes the following topics:

Defining a Distinguished Name using the configConfMo Method

The **configConfMo** method is used to configure one or more properties in a Managed Object (MO). The MO to be configured is uniquely identified by a Distinguished Name (DN). This chapter shows two ways to provide a DN using the **configConfMo** method.

At the Managed Object Level

You can provide a DN at the Managed Object level. In the following example, the DN " sys/chassis-1/server-1/locator-led" is defined within the "equipmentLocatorLed" MO.

```
<configConfMo
    cookie="<real_cookie>">
    <inConfig>
        <equipmentLocatorLed
            adminState='on'
            dn='sys/chassis-1/server-1/locator-led'>                    <== MO level
        </equipmentLocatorLed>
    </inConfig>
</configConfMo>
```

At the Method and Managed Object Level

You can provide a DN at the Method and Managed Object level. In the following example, the DN "sys/chassis-1/server-1/locator-led" is defined at the **configConfMo** method level and within the "equipmentLocatorLed" MO.

```
<configConfMo
    cookie="<real_cookie>"
    dn='sys/chassis-1/server-1/locator-led'>                    <== Method
level
```

```
    <inConfig>
        <equipmentLocatorLed
            adminState='on'
            dn='sys/chassis-1/server-1/locator-led'>                    <== MO Level
        </equipmentLocatorLed>
    </inConfig>
</configConfMo>
```

Note Specifying a DN at the Method level is optional, and is supported in the Cisco IMC XML API implementation to be consistent with the Cisco UCS Manager XML API implementation.

Using the Optional inHierarchical Attribute

When a **configConfMo** request is sent to Cisco IMC, the response contains only the immediate properties of the MO being configured.

When the optional inHierarchical attribute is included in the **configConfMo** request, the response will be similar to that of the **configResolveDn** request with the inHierarchical attribute set to true. The response contains the properties for the MO being configured along with the properties of any children MOs.

Request:

```
<configConfMo
    cookie="<real_cookie>"
    inHierarchical="true"
    dn='sys/ chassis-1/server-1/locator-led'>
    <inConfig>
        <equipmentLocatorLed
            adminState='on'
            dn='sys/ chassis-1/server-1/locator-led'>
        </equipmentLocatorLed>
    </inConfig>
</configConfMo>
```

Response:

```
<configConfMo
    dn="sys/ chassis-1/server-1/locator-led"
    cookie="<real_cookie>"
    response="yes">
    <outConfig>
        <equipmentLocatorLed
            dn="sys/ chassis-1/server-1/locator-led"
            adminState="inactive"
            color="unknown"
            id="1"
            name=""
            operState="on">
        </equipmentLocatorLed>
    </outConfig>
</configConfMo>
```

Configuring a Single Managed Object

The Cisco IMC XML API implementation accepts only **configConfMo** methods that operate on a single Managed Object (MO). It is invalid to specify a **configConfMo** method that contains multiple MOs even if they are defined in a containment relationship in the Cisco IMC management information model.

The following example shows a valid **configConfMo** method to configure a single MO, "lsbootLan." In this example, the host is configured to use PXE Boot as the first boot option:

```
<configConfMo
    cookie="<real_cookie>">
    <inConfig>
        <lsbootLan                                              <== Single MO
            order="1"
            status="modified"
            dn="sys/chassis-1/server-1/boot-policy/lan" >
        </lsbootLan>
    </inConfig>
</configConfMo>
```

The **configConfMo** method in the following example is invalid because a Parent and Child MOs are specified at the same time. The "equipmentLocatorLed" and "solIf" MOs are child objects of the " computeServerNode" MO in the management information tree. The Cisco IMC XML API implementation does not allow a **configConfMo** method to perform subtree configurations.

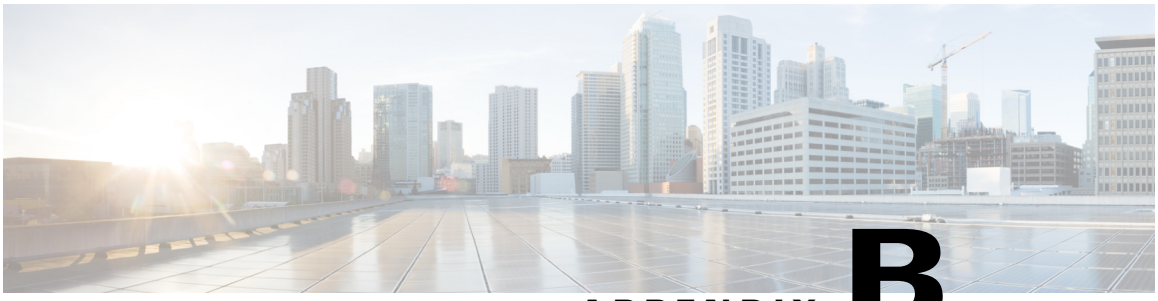
Request:

```
<configConfMo
    cookie="1313084260/40ea8058-aa3e-1a3e-8004-5e61c2e14388"
    dn="sys/chassis-1/server-1" inHierarchical="false">
     <inConfig>
         <computeServerNode                                        <== Parent MO
             adminPower="cycle-immediate"
             usrLbl="Cisco C210 Server"
             dn="sys/chassis-1/server-1">
                  <equipmentLocatorLed                          <== Child MO
                      adminState="on"
                      dn="sys/chassis-1/server-1/locator-led"/>
                  <solIf                                        <== Child MO
                      dn="sys/chassis-1/server-1/solif"
                      adminState="enable"
                      speed="9600"/>
         </ computeServerNode>
      </inConfig>
</configConfMo>
```

Response:

```
XML PARSING ERROR: Element 'equipmentLocatorLed': This element is not expected.
```

Note This method is valid in the Cisco UCS Manager XML API implementation but is not supported in the Cisco IMC XML API implementation.



APPENDIX B

The Cisco IMC Visore Utility

Visore is a utility built into Cisco IMC that allows a user to easily browse Managed Objects (MOs) using an HTML browser. The Visore utility uses the Cisco IMC XML API query methods to browse the MOs active in Cisco IMC. The Visore utility cannot be used to perform configuration operations.

Accessing Visore

To access Visore, open a browser and enter one of the following URLs:

- http://<Cisco IMC IP Address>/visore.html
- https://<Cisco IMC IP Address>/visore.html

When prompted, log in using the same credentials you would use to log in to the Cisco IMC CLI or GUI user interfaces.

Using Visore to Query a Class

To query for a particular class, enter the class name in the **Class or DN** field and click **Run Query**. Visore sends a **configResolveClass** method to Cisco IMC and the requested MO is displayed in a tabular format..

Use the **<** and **>** buttons to retrieve the Parent and Child class of the displayed MO. For example, clicking **>** sends a **configResolveChildren** method to Cisco IMC to query for the child of the MO. Clicking **<** sends a **configResolveParent** method to Cisco IMC to query for the parent of the MO.

Using Visore to Query a Distinguished Name(DN)

To query for a particular DN, enter the DN in the **Class or DN** field and click **Run Query**. Visore sends a **configResolveDn** method to Cisco IMC.

INDEX

U

V

X