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/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/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/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 all components 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 All Components to Factory Defaults, on page 4
- Resetting Storage, VIC, BMC1, BMC2, and CMC Components to Factory Defaults, on page 5
- Rebooting CMC, on page 5

Resetting All Components to Factory Defaults

Request:

```
<configConfMo dn='sys/chassis-1'
cookie="1484920480/dd879b5e-8746-1687-8003-be18652a6ca4" inHierarchical="true">
 <inConfig>
   <equipmentChassis dn='sys/chassis-1' resetComponents='all'>
   </equipmentChassis>
 </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn='sys/chassis-1/server-1'
cookie="1484920480/dd879b5e-8746-1687-8003-be18652a6ca4" response="yes">
  <outConfig>
  <equipmentChassis dn="sys/chassis-1" name="UCS S3260" serial="FOX2037GHYF"
  model="UCSS-S3260-BASE" usrLbl="" assetTag="Unknown"
  serverSIOCConnectivity="Disabled" resetComponents="components"
  rebootHost="no" storageResetStatus="Inprogress" siocResetStatus="NA"
  bmc1ResetStatus="Inprogress" bmc2ResetStatus="Inprogress"
  cmcResetStatus="Inprogress" status="modified">
  </equipmentChassis>
 </outConfig>
</configConfMo>
```

Resetting Storage, VIC, BMC1, BMC2, and CMC Components to Factory Defaults

Request:

```
<configConfMo dn="sys/chassis-1"
cookie="1484920480/dd879b5e-8746-1687-8003-be18652a6ca4" inHierarchical="true">
 <inConfig>
  <equipmentChassis dn="sys/chassis-1"
  resetComponents="bmc1,bmc2,cmc,vic,storage">
  </equipmentChassis>
 </inConfig>
</configConfMo>
```

Response:

```
<configConfMo dn="sys/chassis-1"
cookie="1484920480/dd879b5e-8746-1687-8003-be18652a6ca4" response="yes">
 <outConfig>
  <equipmentChassis dn="sys/chassis-1" name="UCS S3260" serial="FOX2037GHYF"
  model="UCSS-S3260-BASE" usrLbl="" assetTag="Unknown"
  serverSIOCConnectivity="Disabled" resetComponents="components"
  rebootHost="no" storageResetStatus="Inprogress" siocResetStatus="NA"
  bmc1ResetStatus="Inprogress" bmc2ResetStatus="Inprogress"
  cmcResetStatus="Inprogress" status="modified">
  </equipmentChassis>
 </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>
```

Generating Hardware Inventory and Saving it on a Remote Share

The following example shows how to generate and save the hardware inventory. The example uses the SCP protocol, you can also use the HTTP, FTP, SFTP, and TFTP protocols while performing this task.

Generating Hardware Inventory for the Server

Request:

```
<configConfMo cookie="1453259331/c15fc890-29bb-19bb-8003-a2fa28e6a274"
inHierarchical="true" dn="sys/inventory">
<inConfig>
 <mgmtInventory pwd="cisco123" user="user" adminState="trigger"
   hostname="10.104.236.99" remoteFile="/home/user/pynj/FP_XMLAPI_SCP.txt"
   proto="scp" dn="sys/inventory" >
  </mgmtInventory>
 </inConfig>
</configConfMo>
```

Response:

```
<configConfMo cookie="1453259331/c15fc890-29bb-19bb-8003-a2fa28e6a274"
dn="sys/inventory" response="yes" >
 <outConfig>
  <mgmtInventory dn="sys/inventory" adminState="triggered" proto="none"
   hostname="" remoteFile="" user="" pwd="" fsmStatus="IN-PROGRESS"
   progress="5%" status="modified" >
 </mgmtInventory>
 </outConfig>
</configConfMo>
```