



Using Cisco Voice CORBA Gateway Operations

Cisco Voice CORBA Gateway supports the following types of operations:

- [Cisco Voice CORBA Gateway GET and SET, page 4-2](#)
- [Cisco Voice CORBA Gateway Bulk Operations, page 4-5](#)
- [Cisco Voice CORBA Gateway Query, page 4-10](#)
- [Cisco Voice CORBA Gateway Upload, page 4-11](#)

Cisco Voice CORBA Gateway GET and SET

The GET and SET operations support configurations of one Cisco Gateway at a time.

GET Operation

This operation gets the specified attribute values of an EMS object.

```
<GETREQUEST>
  <OBJECTATTRIBUTE name="attributeName1"></OBJECTATTRIBUTE>
  <OBJECTATTRIBUTE name="attributeName2"></OBJECTATTRIBUTE>
  ...
</GETREQUEST>
```

Syntax Description

The GET syntax is a list of attribute names in an XML string. For more information on attributes, see the attribute summary appendices.

Usage Guidelines

To use the GET operation, build an XML string with a list of attributes whose values to be retrieved and invoke the appropriate IDL method on the VCG server (invoke() method).

The XML string with the required attributes looks as follows:

```
<GETREQUEST>
  <OBJECTATTRIBUTE name="attributeName1"></OBJECTATTRIBUTE>
  <OBJECTATTRIBUTE name="attributeName2"></OBJECTATTRIBUTE>
  ...
</GETREQUEST>
```

The result of the operation is an XML string and looks as follows:

```
<OPERATIONRESULTS>
  <RESULTUNIT>
    <GETRESULTS>
      <OBJECTATTRIBUTE name="attributeName1">attributeValue1</OBJECTATTRIBUTE>
      <OBJECTATTRIBUTE name="attributeName2">attributeValue2</OBJECTATTRIBUTE>
      ...
    </GETRESULTS>
  </RESULTUNIT>
</OPERATIONRESULTS>
```

Examples

The following input XML string gets two attribute values:

```
<GETREQUEST>
  <OBJECTATTRIBUTE name="SNMP:SYSTEM-SCALARS-MIB.systemNodeName"></OBJECTATTRIBUTE>
  <OBJECTATTRIBUTE name="SNMP:SYSTEM-SCALARS-MIB.systemNodeType"></OBJECTATTRIBUTE>
</GETREQUEST>
```

The following lines show the result operation that gets the above two attributes:

```
<OPERATIONRESULTS>
  <RESULTUNIT>
    <GETRESULTS>
      <OBJECTATTRIBUTE
name="SNMP:SYSTEM-SCALARS-MIB.systemNodeName">bimini</OBJECTATTRIBUTE>
      <OBJECTATTRIBUTE
name="SNMP:SYSTEM-SCALARS-MIB.systemNodeType">0</OBJECTATTRIBUTE>
    </GETRESULTS>
  </RESULTUNIT>
</OPERATIONRESULTS>
```

SET Operation

This operation set the given object attributes with the corresponding given attribute values of an EMS object.

```
<SETREQUEST>
  <OBJECTATTRIBUTE name="attributeName1">attributeValue1</OBJECTATTRIBUTE>
  <OBJECTATTRIBUTE name="attributeName2">attributeValue1</OBJECTATTRIBUTE>
  ...
</SETREQUEST>
```

Syntax Description

The SET syntax is a list of attribute names and values in an XML string. For more information on attributes, see the attribute summary appendices. Set those attributes that have read/write permission.

Usage Guidelines

To use the SET operation, build an XML string with a list of attributes and values to be set and invoke the appropriate IDL method on the VCG server (invoke() method).

The XML string with the required attributes looks as follows:

```
<SETREQUEST>
  <OBJECTATTRIBUTE name="attributeName1">attributeValue1</OBJECTATTRIBUTE>
  <OBJECTATTRIBUTE name="attributeName2">attributeValue2</OBJECTATTRIBUTE>
  ...
</SETREQUEST>
```

The result of the operation is an XML string and looks as follows:

```
<OPERATIONRESULTS>
  <RESULTUNIT>
    <SETRESULTS>
      <OBJECTATTRIBUTE name="attributeName1">attributeValue1</OBJECTATTRIBUTE>
      <OBJECTATTRIBUTE name="attributeName2">attributeValue2</OBJECTATTRIBUTE>
      ...
    </SETRESULTS>
  </RESULTUNIT>
</OPERATIONRESULTS>
```

Examples

The following input XML string shows how to set a system attribute:

```
<SETREQUEST>
  <OBJECTATTRIBUTE
    name="SNMP:SYSTEM-SCALARS-MIB.systemNodeName">bimini</OBJECTATTRIBUTE>
</SETREQUEST>
```

The following lines show the result operation that sets the above mentioned attribute:

```
<OPERATIONRESULTS>
  <RESULTUNIT>
    <SETRESULTS>
      <OBJECTATTRIBUTE
        name="SNMP:SYSTEM-SCALARS-MIB.systemNodeName">bimini</OBJECTATTRIBUTE>
      </SETRESULTS>
    </RESULTUNIT>
  </OPERATIONRESULTS>
```

Cisco Voice CORBA Gateway Bulk Operations

Bulk operations support multichassis configurations. Cisco Voice CORBA Gateway supports the following bulk operations:

- BULKACTION (software download only)
- BULKGET
- BULKSET

Bulk Action Operation

This operation performs the specified action on a multiple EMS objects.

```
<BULKACTION><ACTIONNAME>SoftwareDownloadAction</ACTIONNAME>
  <MO identity="CMGMMGX8260">
    <OA name= "PATH">cmgm:CMGM_Site_default/10.1.1.1</OA>
    <OA name= "PATH">cmgm:CMGM_Site_default/10.1.1.2</OA>
    . . .
    <OA name= "cmgmCtrlr:cmgmCtrlr-MIB.SoftwareDownloadSourceFile">attributeValue</OA>
    <OA name= "cmgmCtrlr:cmgmCtrlr-MIB.SoftwareDownloadDestFile">attributeValue</OA>
  </MO>
</BULKACTION>
```

Syntax Description

The bulk action syntax includes an action name to be carried out on objects of the class identified by attribute "identify" of the <MO> tag. Furthermore, it specifies the action to be carried out on specific instances of the class specified by the special attribute name "PATH", in the list of attributes specified. The parameters of the action are those attributes which are not the name "PATH" in the list of attributes specified. The above can be repeated for more than one class; the <MO> tag can be repeated multiple times.

Table 4-1 Bulk Action Attributes

attributeName	attributeValue	Type
PATH	The containment path for the EM object. This attribute is required.	char
EMCtrlr:EMCtrlr-MIB.SoftwareDownloadSourceFile	The fully qualified name and path to the software image source file. This attribute is required.	char
EMCtrlr:EMCtrlr-MIB.SoftwareDownloadDestFile	The destination file name. This attribute is required. The name of the software image. This name indicates the card type, major release, minor release, and version, followed by the .fw extension. For example, SCC_R01.02.03.FW indicates SCC software release 1.2.3. The release and version information (R01.02.03) must be 9 characters in length.	char

Usage Guidelines

To use the BULK ACTION operation, build an XML string with list of EMS objects, parameters (attribute names and values, if any). This is an example of the result of a BULKACTION request:

The XML string with the required attributes looks as follows:

```
<BULKACTION> <ACTIONNAME>actionName</ACTIONNAME>
  <MO identity="classIdentifier">
    <OA name="PATH">FullyDistinguishedName1</OA>
    <OA name="PATH">FullyDistinguishedName2</OA>
    . . . .
    <OA name="attributeName1">attributeValue1</OTA>
    <OA name="attributeName2">attributeValue2</OA>
    . . .
  </MO>
</BULKACTION>
```

The result XML string may look like this:

```
<OPERATIONRESULTS>
  <RESULTUNIT>
    <BULKACTIONRESULTS>
      <PATH name="FullyDistinguishedName1">
        <OBJECTATTRIBUTE attributeName> attributeValue</OBJECTATTRIBUTE >
        . . .
      </PATH>
      <PATH name="FullyDistinguishedName2">
        <OBJECTATTRIBUTE attributeName> attributeValue</OBJECTATTRIBUTE >
      </PATH>
      . . .
    </BULKACTIONRESULTS>
  </RESULTUNIT>
</OPERATIONRESULTS>
```

Examples

The following example downloads SCC software to two destinations.

```
<BULKACTION><ACTIONNAME>SoftwareDownloadAction</ACTIONNAME>
  <MO identity="CMGMMGX8260">
    <OA name="PATH">cmgm:CMGM_Site_default/172.16.38.201</OA>
    <OA name="PATH">cmgm:CMGM_Site_default/172.16.38.111</OA>
    <OA name="cmgmCtrlr:cmgmCtrlr-MIB.SoftwareDownloadSourceFile">
      /mgx/scc/sw/SCC_R01.02.03</OA>
    <OA name="cmgmCtrlr:cmgmCtrlr-MIB.SoftwareDownloadDestFile">/SCC_R01.02.03.fw</OA>
  </MO>
</BULKACTION>

<BULKACTION><ACTIONNAME>SoftwareDownloadAction</ACTIONNAME>
  <MO identity="CMGMMGX8260">
    <OA name="PATH">cmgm:CMGM_Site_default/172.16.38.201</OA>
    <OA name="PATH">cmgm:CMGM_Site_default/172.16.38.111</OA>
    <OA name="cmgmCtrlr:cmgmCtrlr-MIB.SoftwareDownloadSourceFile">
      /mgx/scc/sw/SCC_R01.02.03</OA>
    <OA name="cmgmCtrlr:cmgmCtrlr-MIB.SoftwareDownloadDestFile">/SCC_R01.02.03.fw</OA>
  </MO>
</BULKACTION>
```

BULKGET Request Operation

This operation gets attributes from multiple Cisco MGX 8260 Media Gateway objects.

```
<BULKGETREQUEST>
  <MO identity="cmgm:CMGM_Site_default/172.16.38.201">
    <OA name=attributeName1></OA>
    <OA name=attributeName2></OA>
    ...
  </MO>
  <MO identity="cmgm:CMGM_Site_default/172.16.38.111">
    <OA name=attributeName3></OA>
    ...
  </MO>
  ...
</BULKGETREQUEST>
```

Syntax Description

The BULKGETREQUEST operation can read attributes from multiple Cisco MGX 8260 Media Gateways. The request XML string has sections for each chassis, each of which contain a list of attribute names for which attribute values are required. The syntax shows how to use the <MO> and <OA> tags for this type of request.

Usage Guidelines

To use the BULKGETREQUEST operation, build an XML file with a list of chassis, attributes and values to retrieve. The following example shows the results of a BULDGETREQUEST argument:

```
<OPERATIONRESULTS>
  <RESULTUNIT>
    <BULKGETRESULTS>
      <MO identity="cmgm:CMGM_Site_default/172.16.38.201">
        <OA name="SNMP:SYSTEM-SCALARS-MIB.systemNodeName">bimini</OA>
        <OA name="SNMP:SYSTEM-SCALARS-MIB.systemNodeType">0</OA>
      </MO>
      <MO identity="cmgm:CMGM_Site_default/172.16.38.111">
        <OA name="SNMP:SYSTEM-SCALARS-MIB.systemNodeName">oahu</OA>
      </MO>
    </BULKGETRESULTS>
  </RESULTUNIT>
</OPERATIONRESULTS>
```

Examples

The following example shows how to get system scalars from two identities.

```
<BULKGETREQUEST>
  <MO identity="cmgm:/172.16.38.111">
    <OA name="SNMP:SYSTEM-SCALARS-MIB.systemNodeName"></OA>
    <OA name="SNMP:SYSTEM-SCALARS-MIB.systemNodeType"></OA>
  </MO>
  <MO identity="cmgm:/172.17.38.201">
    <OA name="SNMP:SYSTEM-SCALARS-MIB.systemNodeName"></OA>
  </MO>
</BULKGETREQUEST>
```

The following example shows how to get DS1 attributes from two identities.

```
<BULKGETREQUEST>
  <MO identity="cmgm:/172.16.38.111/NSC_Slot_2/DS1_Line_3">
    <OA name="SNMP:Cmgmmodule.rFC1406-MIB_dsx1LineType1"></OA>
    <OA name="SNMP:Cmgmmodule.rFC1406-MIB_dsx1IfIndex1"></OA>
    <OA name="SNMP:Cmgmmodule.rFC1406-MIB_dsx1BertDs0s1"></OA>
    <OA name="SNMP:Cmgmmodule.rFC1406-MIB_dsx1ValidIntervals1"></OA><OA
  </MO>
  <MO identity="cmgm:/172.17.38.111/NSC_Slot_2/DS1_Line_4">
    <OA name="SNMP:Cmgmmodule.rFC1406-MIB_dsx1IfIndex1"></OA>
  </MO>
</BULKGETREQUEST>
```

BULKSET Request Operation

This operation sets attributes in multiple Cisco MGX 8260 Media Gateway objects.

```
<BULKSETREQUEST>
  <MO identity="cmgm:CMGM_Site_default/172.16.38.201">
    <OA name=attributeName1>attributeValue1</OA>
    <OA name=attributeName2>attributeValue2</OA>
    ...
  </MO>
  <MO identity="cmgm:CMGM_Site_default/172.16.38.111">
    <OA name=attributeName3>attributeValue3</OA>
    ...
  </MO>
  ...
</BULKSETREQUEST>
```

Syntax Description

The BULKSETREQUEST operation can read attributes from multiple Cisco MGX 8260 Media Gateways. The request XML string has sections for each chassis, each of which contain a list of attribute names for which attribute values are required. The syntax shows how to use the <MO> and <OA> tags for this type of request.

Usage Guidelines

To use the BULKSETREQUEST operation, build an XML file with a list of chassis, attributes, and values to retrieve. The following example shows the results of a BULDSETREQUEST operation:

```
<OPERATIONRESULTS>
  <RESULTUNIT>
    <BULKSETRESULTS>
      <MO identity="cmgm:CMGM_Site_default/172.16.38.201">
        <OA name="SNMP:SYSTEM-SCALARS-MIB.systemNodeName">bimini</OA>
        <OA name="SNMP:SYSTEM-SCALARS-MIB.systemNodeType">0</OA>
      </MO>
      <MO identity="cmgm:CMGM_Site_default/172.16.38.111">
        <OA name="SNMP:SYSTEM-SCALARS-MIB.systemNodeName">oahu</OA>
      </MO>
    </BULKSETRESULTS>
  </RESULTUNIT>
</OPERATIONRESULTS>
```

Examples

The following example sets the system node name for two identities.

```
<BULKSETREQUEST>
  <MO identity="cmgm:/172.17.38.201">
    <OA name="SNMP:SYSTEM-SCALARS-MIB.systemNodeName">bimini</OA>
  </MO>
  ><MO identity="cmgm:/172.17.38.111">
    <OA name="SNMP:SYSTEM-SCALARS-MIB.systemNodeName">oahu</OA>
  </MO>
</BULKSETREQUEST>
```

Cisco Voice CORBA Gateway Query

The QUERY operation discovers objects in the root path that match the query criteria. The root path is the containment path to search; the query criteria are keywords that identify subchassis objects. This operation facilitates discovering the containment path for all objects of a specific type.

QUERY Operation

This operation searches a root path for child classes.

```
<QUERY>
  <ROOTPATH> cmgm: /</ROOTPATH>
  <CHILDCLASS>identity1</CHILDCLASS>
  <CHILDCLASS>identity2</CHILDCLASS>
  . . .
</QUERY>
```

Syntax Description

The XML input specifies the ROOTPATH in the EMS object tree whose children are to be searched for the matching CLASSNAME.

Usage Guidelines

To query for managed objects, construct an XML string with a list of the identities to discover. Query all sites by specifying EM: as the target. Cisco Voice CORBA Gateway discovers and returns the full path to the discovered objects. The root path of EM: searches for all objects that match the specified identity.

The results of a query return the containment path for each child class found. The following example shows the results of a QUERY for BSC cards:

```
<OPERATIONRESULTS>
  <RESULTUNIT>
    <QUERYRESULTS>
      <PATH>cmgm:/CMGM_Site_Saratoga/172.16.38.111/BSC_Slot_15</PATH>
      <PATH>cmgm:/CMGM_Site_Saratoga/172.16.38.111/BSC_Slot_11</PATH>
      <PATH>cmgm:/CMGM_Site_Saratoga/172.16.38.111/BSC_Slot_16</PATH>
      <PATH>cmgm:/CMGM_Site_Saratoga/172.16.38.111/BSC_Slot_12</PATH>
      <PATH>cmgm:/CMGM_Site_Saratoga/172.16.38.111/BSC_Slot_13</PATH>
      <PATH>cmgm:/CMGM_Site_Saratoga/172.16.38.111/BSC_Slot_14</PATH>
    </QUERYRESULTS>
  </RESULTUNIT>
</OPERATIONRESULTS>
```

Examples

The following example queries for BSC cards.

```
<QUERY><ROOTPATH>cmgm:</ROOTPATH>
  <CHILDCLASS>CMGMBS</CHILDCLASS>
</QUERY>
```

Cisco Voice CORBA Gateway Upload

The UPLOAD discovers object information according to filter criteria and transfers the results to a FTP host.

UPLOAD Operation

This operation discovers information about objects and FTP them to a host computer.

```
<UPLOADREQUEST>
  <USERNAME>hostUsername</USERNAME>
  <PASSWORD>hostPassword</PASSWORD>
  <FTPPFILENAME>destinationFile</FTPPFILENAME>
  <FTPHOSTNAME>hostName</FTPHOSTNAME>
  <CLASSFILTER>
    <MO identity="CMGMMGX8260">
      <OA name= attributeName1></OA>
      <OA name= attributeName2></OA>
      <TB name=tablename1>
        <ROW>
          <TA name=tablecolumnName1></TA>
          <TA name=tablecolumnName2></TA>
        </ROW>
      </TB>
    ...
  </MO>
  <MO identity="CMGMBSC">
    <OA name= attributeName3 ></OA>
    <OA name= attributeName4></OA>
    ...
  </MO>
  ...
</CLASSFILTER>
</UPLOADREQUEST>
```

Syntax Description

Query all sites by specifying EM: as the target, or narrow the upload to specific chassis objects.

(1) Example I of the UPLOAD results.

```
<INVENTORYUPLOAD>
  <MO identity="MGX8260">
    <PATH>cmgm:/</PATH>
    <OID> OpaqueID</OID>
    <OA name=" systemNodeName"> SJ-15</OA>
    <OA name="sysNodeNum">4 </OA>
    <OA name="systemBkPlaneSerialNum"> 3456789 </OA>
  </MO>
  <MO identity="CMGMBSC">
    <PATH>cmgm:/</PATH>
    <OID> OpaqueID </OID>
    <OA name=" logicalCardStatus"> 5</OA>
    <OA name=" logicalCardFrontCardType"> 3</OA>
    <OA name=" logicalCardServices"> 2 </OA>
  </MO>
</INVENTORYUPLOAD>
```

(1) Example II of the UPLOAD results with table attributes for MGX8850.

```
<InventoryUpload>
  <MO identity="CMGMVMGX8850PXM45">
    <PATH>
      Physical:/CMGM_Site_SJ20_1/192.168.118.129
    </PATH>
    <OID>
      257:4114
    </OID>
    <OA name = "contObjPath">Physical:/CMGM_Site_SJ20_1/192.168.118.129</OA>
    <OA name = "contObjName">192.168.118.129</OA>
    <OA name = "SNMPv2-MIB.sysUpTime">225153759</OA>
    <OA name = "SNMPv2-MIB.sysObjectID">1.3.6.1.4.1.9.1.228</OA>
    <OA name = "SNMPv2-MIB.sysDescr">Cisco MGX8850 Wide Area Switch</OA>
    <TB name = "Cmgmvmodule.cMGMVPop2ChassisSmRedMapTable">
    <ROW>
    <TA name = "BASIS-SHELF-MIB.redPrimarySlotNum">7</TA>
    <TA name = "BASIS-SHELF-MIB.redRowStatus">1</TA>
    <TA name = "BASIS-SHELF-MIB.redPrimaryType">No_Value</TA>
    <TA name = "BASIS-SHELF-MIB.redPrimaryState">3</TA>
    <TA name = "BASIS-SHELF-MIB.redSecondarySlotNum">8</TA>
    <TA name = "BASIS-SHELF-MIB.redSecondaryType">No_Value</TA>
    <TA name = "BASIS-SHELF-MIB.redSecondaryState">1</TA>
    <TA name = "BASIS-SHELF-MIB.redType">1</TA>
    <TA name = "BASIS-SHELF-MIB.redCoveringSlot">0</TA>
    <TA name = "BASIS-SHELF-MIB.redFeature">No_Value</TA>
    <TA name = "BASIS-SHELF-MIB.redLineModuleType">No_Value</TA>
    </ROW>
    </TB>
    <OA name = "AMAF-MGMT-MIB.ipaddress">192.168.118.129</OA>
  </MO>
  ...
</InventoryUpload>
```

Examples

The following example uploads information according to a class filter.

```

<UPLOADREQUEST>
  <USERNAME>root</USERNAME>
  <PASSWORD>lab</PASSWORD>
  <FTPFILENAME>/tmp/destination</FTPFILENAME>
  <FTPHOSTNAME>sp-u60-1</FTPHOSTNAME>
  <CLASSFILTER>
    <MO identity="CMGMMGX8260">
      <OA name="SNMP:SYSTEM-SCALARS-MIB.systemDate"></OA>
      <OA name="SNMP:SYSTEM-SCALARS-MIB.systemDs1Type"></OA>
      <OA name="SNMP:SYSTEM-SCALARS-MIB.systemNodeName"></OA>
      <OA name="SNMP:SYSTEM-SCALARS-MIB.systemNodeNum"></OA>
      <OA name="SNMP:SYSTEM-SCALARS-MIB.systemNodeType"></OA>
    </MO>
    <MO identity="CMGMBSC">
      <OA name="SNMP:Cmgmmodule.cCOMMON-SLOT-MIB_logicalCardFwRev1"></OA>
      <OA name="SNMP:Cmgmmodule.cCOMMON-SLOT-MIB_logicalCardHwRev1"></OA>
      <OA name="SNMP:Cmgmmodule.cCOMMON-SLOT-MIB_logicalCardStatus1"></OA>
      <OA name="SNMP:Cmgmmodule.cCOMMON-SLOT-MIB_logicalCardPhysicalSlotNum1"></OA>
    </MO><
    <MO identity="CMGMSCC">
      <OA name="SNMP:Cmgmmodule.cCOMMON-SLOT-MIB_logicalCardFwRev1"></OA>
      <OA name="SNMP:Cmgmmodule.cCOMMON-SLOT-MIB_logicalCardHwRev1"></OA>
      <OA name="SNMP:Cmgmmodule.cCOMMON-SLOT-MIB_logicalCardStatus1"></OA>
      <OA name="SNMP:Cmgmmodule.cCOMMON-SLOT-MIB_logicalCardPhysicalSlotNum1"></OA>
    </MO>
    <MO identity="CMGMDS1">
      <OA name="LocalDB:RFC1406-MIB.dsx1LineIndex"></OA>
      <OA name="SNMP:Cmgmmodule.rFC1406-MIB_dsx1IfIndex1"></OA>
      <OA name="SNMP:Cmgmmodule.rFC1406-MIB_dsx1LineType1"></OA>
      <OA name="SNMP:Cmgmmodule.rFC1406-MIB_dsx1LineCoding1"></OA>
      <OA name="SNMP:Cmgmmodule.rFC1406-MIB_dsx1CircuitIdentifier1"></OA>
    </MO>
    <MO identity="CMGMDS3">
      <OA name="LocalDB:RFC1406-MIB.dsx3LineIndex"></OA>
      <OA name="SNMP:Cmgmmodule.rFC1406-MIB_dsx3IfIndex1"></OA>
      <OA name="SNMP:Cmgmmodule.rFC1406-MIB_dsx3LineType1"></OA>
    </MO>
  </CLASSFILTER>
</UPLOADREQUEST>

```

