



XML Provisioning Guide for Cisco CME/SRST

Revised 08/08/2003

This document describes the XML interface to the Cisco CallManager Express (CME) and the Cisco Survivable Remote Site Telephony (SRST).

This document is divided into the following sections:

- [Target Audience](#)
- [XML Implementation](#)
- [Examples](#)
- [Related Documentation](#)
- [Glossary](#)
- [Cisco Connection Online](#)
- [Documentation CD-ROM](#)

Target Audience

This guide assumes the developer has knowledge of a high-level programming language, such as C++, Java, or an equivalent language. The developer must also have knowledge or experience in the following areas:

- TCP/IP Protocol
- Hypertext Transport Protocol
- Socket programming
- XML

In addition, users of this programming guide must have a firm grasp of XML Schema, which was used to define the AXL requests, responses, and errors. For more information on XML Schema, please refer to <http://www.w3.org/TR/xmlschema-0/>.

XML Implementation

The AVVID XML Layer (AXL) Application Programming Interface (API) provides a mechanism for inserting, retrieving, updating, and removing data from the database using an eXtensible Markup Language (XML) Simple Object Access Protocol (SOAP) interface. This allows a programmer to access data using XML and receive the data in XML form, instead of using a binary library or DLL.

The AXL API methods, known as *requests*, are performed using a combination of HTTP and SOAP. A SOAP is an XML remote procedure call protocol. Users perform requests by sending XML data to the server. The server then returns the AXL *response*; which is also a SOAP message.

Requests	Responses
complexType ISgetGlobal	complexType ISgetGlobalResponse
complexType ISgetDevice	complexType ISgetDeviceResponse
complexType ISgetExtension	complexType ISgetExtensionResponse
complexType ISgetEvtCounts	complexType ISgetEvtCountsResponse
complexType ISgetDevEvts	complexType ISgetDevEvtsResponse
complexType ISgetExtEvts	complexType ISgetExtEvtsResponse
complexType ISsetKeyPhones	complexType ISsetKeyPhonesResponse
complexType ISexecCLI	complexType ISexecCLIResponse

Request Messages

Request methods are XML structures that are passed to the AXL API server. The server receives the XML structures and executes the request. If the request completes successfully, then the appropriate AXL response is returned.

HTTP Format

Requests can be sent through the HTTP POST method. The URL directory after POST should be /ISApi/AXL/V1/soapisapi.is. Two additional headers should also be present in the request:

```
POST/ISApi/AXL/V1/soapisapi.is
Content-type: text/xml
Authorization: Basic <some Basic 64 encoded string>
```

SOAP Format

SOAP messages consist of SOAP-Envelope and SOAP-Body. The format of a SOAP/XML request is:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Body>
    <axl xsi:type="request"
      xmlns="http://www.cisco.com/AXL/1.0"
      xsi:schemaLocation="http://www.cisco.com/AXL/1.0
        http://gkar.cisco.com/shema/axlsoap.xsd">
      // request content
    </axl>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The following XML request methods are supported:

complexType ISgetGlobal

namespace	http://www.cisco.com/AXL/1.0
annotation	Get Status for CME/SRST Global request.
type	extension of axlapi:APIRequest
source	<pre><xsd:complexType name="ISgetGlobal"> <xsd:annotation> <xsd:documentation>Get Status for CME/SRST Global request.</xsd:documentation> </xsd:annotation> <xsd:complexContent> <xsd:extension base="axlapi:APIRequest"/> </xsd:complexContent> </xsd:complexType></pre>

complexType ISgetDevice

namespace	http://www.cisco.com/AXL/1.0
children	element ISgetDevice/ISDevID , element ISgetDevice/ISDevName
annotation	Get Device Status Request.
type	extension of axlapi:APIRequest
source	<pre><xsd:complexType name="ISgetDevice"> <xsd:annotation> <xsd:documentation>Get Device Status Request.</xsd:document> </xsd:annotation> <xsd:complexContent> <xsd:extension base="axlapi:APIRequest"> <xsd:choice> <xsd:annotation> <xsd:documentation>Get Device Status by Device ID or Device Name </xsd:documentation> </xsd:annotation> <xsd:element name="ISDevID" type="xsd:nonNegativeInteger" nullable="false"/> <xsd:element name="ISDevName" type="xsd:string" nullable="false"> <xsd:annotation> <xsd:documentation> Special keyword: "All" for all Devices, "AllRegistered" for all Registered Devices, "AllUnRegistered" for all UnRegistered Devices "AllKeyPhones" for all Keyphones </xsd:documentation> </xsd:annotation> </xsd:choice> </xsd:extension> </xsd:complexContent> </xsd:complexType></pre>

element ISgetDevice/ISDevID

namespace	http://www.cisco.com/AXL/1.0
annotation	Get device status by device ID.
type	xsd:nonNegativeInteger
source	<xsd:element name="ISDevID" type="xsd:nonNegativeInteger" nullable="false" />

element ISgetDevice/ISDevName

namespace	http://www.cisco.com/AXL/1.0
annotation	Get device status by device name.
type	xsd:string
source	<xsd:element name="ISDevName" type="xsd:string" nullable="false">

complexType ISgetExtension

namespace	http://www.cisco.com/AXL/1.0
children	element ISgetExtension/ISextID , element ISgetExtension/ISextNumber
annotation	Get Extension Status Request.
type	extension of axlapi:APIRequest
source	<pre> <xsd:complexType name="ISgetExtension"> <xsd:annotation> <xsd:documentation>Get Extension Status Request.</xsd:documentation> </xsd:annotation> <xsd:complexContent> <xsd:extension base="APIRequest"> <xsd:sequence> <xsd:choice> <xsd:annotation> <xsd:documentation>Get Extension Status by Extension ID, or Extension Name or all. </xsd:documentation> </xsd:annotation> <xsd:element name="ISextID" type="xsd:nonNegativeInteger" nullable="false" /> <xsd:element name="ISextNumber" type="axl:String255"> <xsd:annotation> <xsd:documentation>Special case for ISextNumber: "All" for all Extensions, "AllUp" for all extensions that are "Up" "AllDown" for all extensions that are "Down" </xsd:documentation> </xsd:annotation> </xsd:choice> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> </pre>

element ISgetExtension/ISextID

namespace	http://www.cisco.com/AXL/1.0
annotation	Get extension status by extension ID.
type	xsd:nonNegativeInteger
source	<xsd:element name="ISextID" type="xsd:nonNegativeInteger" nullable="false" />

element ISgetExtension/ISextNumber

namespace	http://www.cisco.com/AXL/1.0
annotation	Get extension status by extension number.
type	axl:String255
source	<xsd:element name="ISextNumber" type="axl:String255">

complexType ISgetEvtCounts

namespace	http://www.cisco.com/AXL/1.0
children	None
annotation	Get counts of device/extension events request.
type	extension of axlapi:APIRequest
source	<pre><xsd:complexType name="ISgetEvtCounts"> <xsd:annotation> <xsd:documentation>Get Counts of Device/Extension Events Request.</xsd:documentation> </xsd:annotation> <xsd:complexContent> <xsd:extension base="axlapi:APIRequest" /> </xsd:complexContent> </xsd:complexType></pre>

complexType ISgetDevEvts

namespace	http://www.cisco.com/AXL/1.0
children	element ISgetDevEvts/ISDevEvtID, element ISgetDevEvts/ISDevID, element ISgetDevEvts/ISDevName
annotation	Get device event request.
type	extension of axlapi:APIRequest
source	<pre><xsd:complexType name="ISgetDevEvts" final="#all"> <xsd:annotation> <xsd:documentation>Get Device Event Request.</xsd:documentation> </xsd:annotation> <xsd:complexContent> <xsd:extension base="axlapi:APIRequest"> <xsd:sequence> <xsd:annotation> <xsd:documentation>Get Device Event by Event ID, or Device ID or Device Name. </xsd:documentation> </xsd:annotation> <choice> <xsd:element name="ISDevEvtID" type="axl:nonNegativeInteger"/> <xsd:element name="ISDevID" type="axl:nonNegativeInteger" /> <xsd:element name="ISDevName" type="xsd:string"> <xsd:annotation> <xsd:documentation>Get Device Event by name. Special case: "All" for all device events. </xsd:documentation> </xsd:annotation> </xsd:element> </choice> </xsd:sequence> </xsd:complexContent> </xsd:complexType></pre>

element ISgetDevEvts/ISDevEvtID

namespace	http://www.cisco.com/AXL/1.0
annotation	Get device event by event ID.
type	axl:nonNegativeInteger
source	<xsd:element name="ISDevEvtID" type="axl:nonNegativeInteger" />

element ISgetDevEvts/ISDevID

namespace	http://www.cisco.com/AXL/1.0
annotation	Get extension status by device ID.
type	axl:nonNegativeInteger
source	<xsd:element name="ISDevID" type="axl:nonNegativeInteger" />

element ISgetDevEvts/ISDevName

namespace	http://www.cisco.com/AXL/1.0
annotation	Get extension status by device name.
type	xsd:string
source	<xsd:element name="ISDevName" type="xsd:string">

complexType ISgetExtEvts

namespace	http://www.cisco.com/AXL/1.0
children	ISExtEvtID , ISExtID , ISExtNumber
annotation	Get extension event request.
type	extension of axlapi:APIRequest
source	<pre> <xsd:complexType name="ISgetExtEvts" final="#all"> <xsd:annotation> <xsd:documentation>Get Extension Event Request.</xsd:documentation> </xsd:annotation> <xsd:complexContent> <xsd:extension base="axlapi:APIRequest"> <xsd:sequence> <xsd:annotation> <xsd:documentation>Get Extension Event by Event ID, or Extension ID or Extension Number. </xsd:documentation> </xsd:annotation> <xsd:choice> <xsd:element name="ISExtEvtID" type="axl:nonNegativeInteger"/> <xsd:element name="ISExtID" type="axl:nonNegativeInteger" /> <xsd:element name="ISExtNumber" type="axl:String255"> <xsd:annotation> <xsd:documentation> Get extension events by extension number. Special case: all for all extension events. </xsd:documentation> </xsd:annotation> </xsd:choice> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> </pre>

element ISgetExtEvts/ISExtEvtID

namespace	http://www.cisco.com/AXL/1.0
annotation	Get extension event by event ID.
type	axl:nonNegativeInteger
source	<xsd:element name="ISExtEvtID" type="axl:nonNegativeInteger" />

element ISgetExtEvts/ISExtID

namespace	http://www.cisco.com/AXL/1.0
annotation	Get extension event by extension ID.
type	axl:nonNegativeInteger
source	<xsd:element name="ISExtID" type="axl:nonNegativeInteger" />

element ISgetExtEvts/ISExtNumber

namespace	http://www.cisco.com/AXL/1.0
annotation	Get extension event by extension number.
type	axl:String255
source	<xsd:element name="ISExtNumber" type="axl:String255">

complexType ISsetKeyPhones

namespace	http://www.cisco.com/AXL/1.0
children	ISPhoneName
annotation	Set a list of phones as keyphone.
type	extension of axlapi:APIRequest
source	<pre> <xsd:complexType name="ISsetKeyPhones"> <xsd:annotation> <xsd:documentation>Set a list of phones as keyphone</xsd:documentation> </xsd:annotation> <xsd:complexContent> <xsd:extension base="axlapi:APIRequest" /> <xsd:sequence> <xsd:annotation> <xsd:documentation>A list of Device Events.</xsd:documentation> </xsd:annotation> <xsd:element name="ISPhoneName" type="axl:string" minOccurs="0" maxOccurs="unbound"> <xsd:annotation> <xsd:documentation>A list of phone names ("SEP" + MacAddress") </xsd:documentation> </xsd:annotation> </xsd:element> </xsd:sequence> </xsd:extension> </xsd:complexContent> </pre>

element ISgetKeyPhones/ISPhoneName

namespace	http://www.cisco.com/AXL/1.0
annotation	Get list of phone names.
type	axl:string
source	<xsd:element name="ISPhoneName" type="axl:string" minOccurs="0" maxOccurs="unbound">

complexType IExecCLI

namespace	http://www.cisco.com/AXL/1.0
children	CLI
annotation	A list of CLI to be executed.
type	extension of axlapi:APIRequest
source	<pre><xsd:complexType name="IExecCLI"> <xsd:annotation> <xsd:documentation>A list of CLI to be executed.</xsd:documentation> </xsd:annotation> <xsd:complexContent> <xsd:extension base="axlapi:APIRequest" /> <xsd:sequence> <xsd:annotation> <xsd:documentation>A list of CLI.</xsd:documentation> </xsd:annotation> <xsd:element name="CLI" type="axl:String64" /> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType></pre>

element IExecCLI/CLI

namespace	http://www.cisco.com/AXL/1.0
annotation	Get list of CLIs.
type	axl:String64
facets	maxLength 64
source	<pre><xsd:element name="CLI" type="axl:String64" /></pre>

Response Messages

All responses are named identically to the associated requests, except that the word “Response” has been appended. For example, the XML response returned from an *ISetGlobal* request is called *ISetGlobalResponse*.

If an error occurs, then an XML error structure is returned wrapped inside of a SOAP Fault structure.

HTTP Format

A Cisco CME or Cisco SRST sends responses using a standard HTTP response format. If a request is successfully processed, a status code 200 is sent.

SOAP Format

A SOAP /AXL response format is similar to a SOAP/AXL request, except the body is as follows:

```
<axl xsi:type="response"
  xmlns="http://www.cisco.com/AXL/1.0"
  xsi:schemaLocation="http://www.cisco.com/AXL/1.0
    http://gkar.cisco.com/xml/xmlschema/axlsoap.xsd">
  // response content
</axl>
```

complexType ISgetGlobalResponse

namespace	http://www.cisco.com/AXL/1.0
children	ISGlobal
annotation	Response for CME/SRST Global Status.
type	extension of axlapi:APIResponse
source	<pre><xsd:complexType name="ISgetGlobalResponse"> <xsd:annotation> <xsd:documentation>Response for CME/SRST Global Status. </xsd:documentation> </xsd:annotation> <xsd:complexContent> <xsd:extension base="axlapi:APIResponse"> <xsd:sequence> <xsd:element name="ISGlobal" type="ISStatusItem"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType></pre>

element ISgetGlobalResponse/ISGlobal

namespace	http://www.cisco.com/AXL/1.0
annotation	Response for CME/SRST Global Status.
type	ISStatusItem
source	<pre><xsd:element name="ISGlobal" type="ISStatusItem" /></pre>

complexType ISgetDeviceResponse

namespace	http://www.cisco.com/AXL/1.0
children	ISDevice
annotation	Response for get device status.
type	extension of axlapi:APIResponse
source	<pre><xsd:complexType name="ISgetDeviceResponse"> <xsd:annotation> <xsd:documentation>Response for Get Device Status.</xsd:documentation> </xsd:annotation> <xsd:complexContent> <xsd:extension base="axlapi:APIResponse"> <xsd:sequence> <xsd:annotation> <xsd:documentation> Response with a list of Device Status. </xsd:documentation> <xsd:annotation> <xsd:element name="ISDevice" type="axl:ISDevStatusItem" minOccurs="0" maxOccurs="unbound" /> </xsd:annotation> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType></pre>

element ISgetDeviceResponse/ISDevice

namespace	http://www.cisco.com/AXL/1.0
annotation	Response for CME/SRST Global Status.
type	axl:ISDevStatusItem
source	<code><xsd:element name="ISDevice" type="axl:ISDevStatusItem" minOccurs="0" maxOccurs="unbound" /></code>

complexType ISgetExtensionResponse

namespace	http://www.cisco.com/AXL/1.0
children	ISExtension
annotation	Response for get extension status.
type	extension of axlapi:APIResponse
source	<pre><xsd:complexType name="ISgetExtensionResponse"> <xsd:annotation> <xsd:documentation>Response for Get Extension Status. </xsd:documentation> </xsd:annotation> <xsd:complexContent> <xsd:extension base="axlapi:APIResponse"> <xsd:sequence> <xsd:annotation> <xsd:documentation>Response with a list of Extension Status. </xsd:documentation> </xsd:annotation> <xsd:element name="ISExtension" type="axl:ISExtStatusItem" minOccurs="0" maxOccurs="unbound" /> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType></pre>

element ISgetExtensionResponse/ISExtension

namespace	http://www.cisco.com/AXL/1.0
annotation	Extension Status for CME /SRST.
type	axl:ISExtStatusItem
source	<code><xsd:element name="ISExtension" type="axl:ISExtStatusItem" minOccurs="0" maxOccurs="unbound" /></code>

complexType ISgetEvtCountsResponse

namespace	http://www.cisco.com/AXL/1.0
children	ISDevEvtCount, ISExtEvtCount
annotation	Response for get extension status.
type	extension of axlapi:APIResponse
source	<pre><xsd:complexType name="ISgetEvtCountsResponse"> <xsd:annotation> <xsd:documentation>Response for Get Device and Extension Events Counts.</xsd:documentation> </xsd:annotation> <xsd:complexContent> <xsd:extension base="axlapi:APIResponse"> <xsd:sequence> <xsd:annotation> <xsd:documentation>Counts of Device and Extension Events in Events tables. </xsd:documentation> </xsd:annotation> <xsd:element name="ISDevEvtCount" type="axl:nonNegativeInteger"/> <xsd:element name="ISExtEvtCount" type="axl:nonNegativeInteger"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType></pre>

element ISgetEvtCountsResponse/ISDevEvtCount

namespace	http://www.cisco.com/AXL/1.0
annotation	Device event count.
type	axl:nonNegativeInteger
source	<xsd:element name="ISDevEvtCount" type="axl:nonNegativeInteger" />

element ISgetEvtCountsResponse/ISExtEvtCount

namespace	http://www.cisco.com/AXL/1.0
annotation	Extension event count.
type	axl:nonNegativeInteger
source	<xsd:element name="ISExtEvtCount" type="axl:nonNegativeInteger" />

complexType ISgetDevEvtsResponse

namespace	http://www.cisco.com/AXL/1.0
children	ISDevEvent
annotation	Device event response.
type	extension of axlapi:APIResponse
source	<pre><xsd:complexType name="ISgetDevEvtsResponse"> <xsd:annotation> <xsd:documentation>Device Event Response.</xsd:documentation> </xsd:annotation> <xsd:complexContent> <xsd:extension base="axlapi:APIResponse"> <xsd:sequence> <xsd:annotation> <xsd:documentation>A list of Device Events.</xsd:documentation> </xsd:annotation> <xsd:element name="ISDevEvent" type="axl:ISDevEventItem" minOccurs="0" maxOccurs="unbound" /> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType></pre>

element ISgetDevEvtsResponse/ISDevEvent

namespace	http://www.cisco.com/AXL/1.0
annotation	List of device events.
type	axl:ISDevEventItem
source	<pre><xsd:element name="ISDevEvent" type="axl:ISDevEventItem" minOccurs="0" maxOccurs="unbound" /></pre>

complexType ISgetExtEvtsResponse

namespace	http://www.cisco.com/AXL/1.0
children	IExtEvent
annotation	Extension event response.
type	Extension of axlapi:APIResponse
source	<pre><xsd:complexType name="ISgetExtEvtsResponse"> <xsd:annotation> <xsd:documentation>Extension Event Response.</xsd:documentation> </xsd:annotation> <xsd:complexContent> <xsd:extension base="axlapi:APIResponse"> <xsd:sequence> <xsd:annotation> <xsd:documentation>A list of Extension Events. </xsd:documentation> </xsd:annotation> <xsd:element name="IExtEvent" type="axl:IExtEventItem" minOccurs="0" maxOccurs="unbound" /> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType></pre>

element ISgetExtEvtsResponse/IExtEvent

namespace	http://www.cisco.com/AXL/1.0
annotation	List of extension events.
type	axl:IExtEventItem
source	<pre><xsd:element name="IExtEvent" type="axl:IExtEventItem" minOccurs="0" maxOccurs="unbound" /></pre>

complexType ISsetKeyPhonesResponse

namespace	http://www.cisco.com/AXL/1.0
children	ISsetResult
annotation	Response for set keyphone.
type	Extension of axlapi:APIResponse
source	<pre><xsd:complexType name="ISsetKeyPhonesResponse"> <xsd:annotation> <xsd:documentation>Response for set keyphone.</xsd:documentation> </xsd:annotation> <xsd:simpleType> <xsd:extension base="axlapi:APIResponse"> <xsd:sequence> <xsd:element name="ISsetResult" type="axl:boolean"/> </xsd:sequence> </xsd:extension> </xsd:simpleType> </xsd:complexType></pre>

element ISsetKeyPhonesResponse/ISsetResult

namespace	http://www.cisco.com/AXL/1.0
annotation	Boolean to indicate if the set if successful or not.
type	axl:boolean
source	<xsd:element name="ISsetResult" type="axl:boolean" />

complexType ISexecCLIResponse

namespace	http://www.cisco.com/AXL/1.0
children	ISexecResult
annotation	Response for executing batch CLIs.
type	Extension of axlapi:APIResponse
source	<pre><xsd:complexType name="ISexecCLIResponse"> <xsd:annotation> <xsd:documentation>Response for executing batch CLIs. </xsd:documentation> </xsd:annotation> <xsd:simpleType> <xsd:extension base="axlapi:APIResponse"> <xsd:sequence> <xsd:annotation> <xsd:documentation>Result of execution CLI. 0: successful >0: line number of CLI caused problem </xsd:documentation> <xsd:element name="ISexecResult" type="xsd:nonNegativeInteger" /> </xsd:sequence> </xsd:extension> </xsd:simpleType> </xsd:complexType></pre>

element ISexecCLIResponse/ISexecResult

namespace	http://www.cisco.com/AXL/1.0
annotation	Boolean to indicate if the set if successful or not.
type	xsd:nonNegativeInteger
source	<xsd:element name="ISexecResult" type="xsd:nonNegativeInteger" />

Examples

Transaction to get Device Status

Request (Server CME/SRST)

```
POST /ISApi/AXL/V1/soapisapi.is
Host: server-w2k:80
Accept: text/xml
Authorization: Basic Q2lzY287VHJpdmVauQ==
Content-type: text/xml
Content-length: 0

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2000/101999/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Body>
    <axl xsi:type="request"
xmlns="http://www.cisco.com/AXL/1.0"
xsi:schemaLocation="http://www.cisco.com/AXL/1.0
http://gkar.cisco.com/xml/xmlschema/axlsoap.xsd">
      <request xsi:type="ISgetDevice">
        <ISDevName>SEP003094C2F4D8</ISDevName>
      </request>
    </axl>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response (CME/SRST Server)

Report a 7960 phone with two lines configured; both lines are idle. It is registered to the CME/SRST 2 minutes after system restart. TAPI client register to CME/SRST 0.5 minute later.

```
HTTP 1.1/ 200 OK
Mime-version: 1.0
Pragma: no-cache
Content-type: text/xml

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2000/101999/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Body>
    <axl xsi:type="response"
xmlns="http://www.cisco.com/AXL/1.0"
xsi:schemaLocation="http://www.cisco.com/AXL/1.0
http://gkar.cisco.com/schema/xml/xml/axlsoap.xsd">
      <response xsi:type="ISgetDeviceResponse">
        <ISDevice>
          <ISDevID>1</ISDevID>
          <ISDevName> SEP003094C2F4D8</ISDevPhysicalAdd>
          <ISDevVlanId>125</ISDevVlanId>
          <ISDevType>7960</ISDevType>
          <ISDevDesc>IP Phone in Cube I2-11, building 22, 2nd floor.</ISDevDesc>
          <ISDevUsername>yuanca</ISDevUsername>
          <ISDevAddr>
            <Xipv4Address>1.5.49.80</Xipv4Address>
          </ISDevAddr>
          <ISPhoneLineList>
            <ExtMapStatus>
              <LineId> 1</LineId>
              <ExtId>1</ExtId>
            </ExtMapStatus>
          </ISPhoneLineList>
        </ISDevice>
      </response>
    </axl>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



```

        <ExtNumber>1234</ExtNumber>
        <ExtStatus>true</ExtStatus>
        <LineState>Idle</LineState>
    </ExtMapStatus>
    <ExtMapStatus>
        <LineId>3</LineId>
        <ExtId>5</ExtId>
        <ExtNumber>5678</ExtNumber>
        <ExtStatus>true</ExtStatus>
    </ExtMapStatus>
</ISPhoneLineList>
<ISKeyPhone>False</ISKeyPhone>
<ISTapiClientAddr>
    <Xipv4Address>172.19.153.129</Xipv4Address>
</ISTapiClientAddr>
<ISDevStatus>Registered</ISDevStatus>
<ISDevLastStatus>New</ISDevLastStatus>
<ISDevChangeTime>12000</ISDevChangeTime>
<ISDevKeepAlives>10</ISDevKeepAlives>
<ISTapiCStatus>Registered</ISTapiCStatus>
<ISTapiCLastStatus>New</ISTapiCLastStatus>
<ISTapiCChangeTime>18000</ISTapiCChangeTime>
<ISTapiCKeepAlives>8</ISTapiCKeepAlives>
</ISDevice>
</response>
</axl>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Transaction to get Extension Event

Request (Server CME/SRST)

```

POST /ISapi/AXL/V1/soapisapi.is
Host: server-w2k:80
Accept: text/xml
Authorization: Basic Q2lzy287VHJpdmVauQ==
Content-type: text/xml
Content-length: 0

```

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/19992000/10/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/1999/XMLSchema">
<SOAP-ENV:Body>
    <axl xsi:type="request"
xmlns="http://www.cisco.com/AXL/1.0"
xsi:schemaLocation="http://www.cisco.com/AXL/1.0
http://gkar.cisco.com/schemaxml/xml/axlsoap.xsd">
        <request xsi:type="ISgetExtEvts">
            <ISExtNumber>1234</ISExtNumber>
        </request>
    </axl>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Response (CME/SRST Server)

Twenty minutes after system restart, extension one with number 1234 goes up.

```

HTTP 1.1/ 200 OK
Mime-version: 1.0
Pragma: no-cache

```

```

Content-type: text/xml

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/19992000/10/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/1999/XMLSchema">

<SOAP-ENV:Body>
  <axl xsi:type="response" xmlns="http://www.cisco.com/AXL/1.0"
xsi:schemaLocation="http://www.cisco.com/AXL/1.0
http://gkar.cisco.com/xml/xmlschema/axlsoap.xsd">
    <response xsi:type="ISgetExtEvtsResponse">
      <ISExtEvent>
        <ISExtID>1</ISExtID>
        <ISExtNumber>1234</ISExtNumber>
        <ISExtEventTime>120000</ISExtEventTime>
        <ISExtEvent>
          <ISExtStatusChangeEvent>Up</ISExtStatusChangeEvent>
        </ISExtEvent>
      </ISExtEvent>
    </response>
  </axl>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Transaction to Execute CLI

Request (Server CME/SRST)

Add a Cisco 7940 phone with line 1 to extension 11; line 2 to extension 12.

```

POST /ISApi/AXL/V1/soapisapi.is
Host: server-w2k:80
Accept: text/xml
Authorization: Basis Q21zY287VHJpdmVauQ==
Content-type: text/xml
Content-length: 0

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/1999/XMLSchema">
<SOAP-ENV:Body>
  <axl xsi:type="request" xmlns="http://www.cisco.com/AXL/1.0
http://gkar.cisco.com/schema/axlsoap.xsd">
    <request xsi:type="ISexecCLI">
      <ISCLIList>
        <CLI>ephone 2</CLI>
        <CLI>mac-address 1234.5678.9012</CLI>
        <CLI>type 7940</CLI>
        <CLI>button 1:10 2:11</CLI>
      </ISCLIList>
    </request>
  </axl>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Response (CME/SRST -> Server)

CLI executed successfully.

```

HTTP 1.1/200 OK
Mime-version: 1.0
Pragma: no-cache
Content-type: text/xml

```

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/1999/XMLSchema">
<SOAP-ENV:Body>
  <axl xsi:type="response" xmlns="http://www.cisco.com/AXL/1.0
  http://gkar.cisco.com/schema/axlsoap.xsd">
    <response xsi:type="ISexecCliResponse">
      <ISexecResult>0</ISexecResult>
    </response>
  </axl>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Related Documentation

For more information about TAPI specifications, creating an application to use TAPI, or TAPI administration see:

- Microsoft TAPI 2.1 Features:
<http://www.microsoft.com/NTServer/commserv/exec/feature/tapi21.asp>
- Getting Started with Windows Telephony
<http://www.microsoft.com/NTServer/commserv/deployment/planguides/getstartedtele.asp>
- Windows Telephony API (TAPI)
<http://www.microsoft.com/NTServer/commserv/exec/overview/tapiabout.asp>
- Creating Next Generation Telephony Applications:
<http://www.microsoft.com/NTServer/commserv/techdetails/prodarch/tapi21wp.asp>
- The Microsoft Telephony Application Programming Interface (TAPI) Programmer's Reference

Glossary

API—Application Programming Interface

AXL—AVVID XML Layer

CME—Cisco CallManager Express.

SRST—Survivable Remote Site Telephony.



Note

For a list of other internetworking terms, see *Internetworking Terms and Acronyms* document that is available on the Documentation CD-ROM and on the Cisco Connection Online (CCO) at the following URL: <http://www.cisco.com/univercd/cc/td/doc/cisintwk/ita/index.htm>.

Cisco Connection Online

Cisco Connection Online (CCO) is Cisco Systems' primary, real-time support channel. Maintenance customers and partners can self-register on CCO to obtain additional information and services.

Available 24 hours a day, 7 days a week, CCO provides a wealth of standard and value-added services to Cisco's customers and business partners. CCO services include product information, product documentation, software updates, release notes, technical tips, the Bug Navigator, configuration notes, brochures, descriptions of service offerings, and download access to public and authorized files.

CCO serves a wide variety of users through two interfaces that are updated and enhanced simultaneously: a character-based version and a multimedia version that resides on the World Wide Web (WWW). The character-based CCO supports Zmodem, Kermit, Xmodem, FTP, and Internet e-mail, and it is excellent for quick access to information over lower bandwidths. The WWW version of CCO provides richly formatted documents with photographs, figures, graphics, and video, as well as hyperlinks to related information.

You can access CCO in the following ways:

- WWW: <http://www.cisco.com>
- WWW: <http://www-europe.cisco.com>
- WWW: <http://www-china.cisco.com>
- Telnet: cco.cisco.com
- Modem: From North America, 408 526-8070; from Europe, 33 1 64 46 40 82. Use the following terminal settings: VT100 emulation; databits: 8; parity: none; stop bits: 1; and connection rates up to 28.8 kbps.

For a copy of CCO's Frequently Asked Questions (FAQ), contact cco-help@cisco.com. For additional information, contact cco-team@cisco.com.

**Note**

If you are a network administrator and need personal technical assistance with a Cisco product that is under warranty or covered by a maintenance contract, contact Cisco's Technical Assistance Center (TAC) at 800 553-2447, 408 526-7209, or tac@cisco.com. To obtain general information about Cisco Systems, Cisco products, or upgrades, contact 800 553-6387, 408 526-7208, or cs-rep@cisco.com.

Documentation CD-ROM

Cisco documentation and additional literature are available in a CD-ROM package, which ships with your product. The Documentation CD-ROM, a member of the Cisco Connection Family, is updated monthly. Therefore, it might be more current than printed documentation. To order additional copies of the Documentation CD-ROM, contact your local sales representative or call customer service. The CD-ROM package is available as a single package or as an annual subscription. You can also access Cisco documentation on the World Wide Web at <http://www.cisco.com>, <http://www-china.cisco.com>, or <http://www-europe.cisco.com>.

If you are reading Cisco product documentation on the World Wide Web, you can submit comments electronically. Click **Feedback** in the toolbar and select **Documentation**. After you complete the form, click **Submit** to send it to Cisco. We appreciate your comments.

This document is to be used in conjunction with the documents listed in the [“Related Documentation”](#) section.

Access Registrar, AccessPath, Any to Any, AtmDirector, CCDA, CCDE, CCDP, CCIE, CCNA, CCNP, CCSI, CD-PAC, the Cisco logo, Cisco Certified Internetwork Expert logo, *CiscoLink*, the Cisco Management Connection logo, the Cisco *NetWorks* logo, the Cisco Powered Network logo, Cisco Systems Capital, the Cisco Systems Capital logo, Cisco Systems Networking Academy, the Cisco Systems Networking Academy logo, the Cisco Technologies logo, ConnectWay, Fast Step, FireRunner, GigaStack, IGX, Internet Quotient, Kernel Proxy, MGX, Natural Network Viewer, NetSonar, Network Registrar, *Packet*, PIX, Point and Click Internetworking, Policy Builder, Precept, Secure Script, ServiceWay, SlideCast, SMARTnet, *The Cell*, TrafficDirector, TransPath, ViewRunner, VisionWay, VlanDirector, Workgroup Director, and Workgroup Stack are trademarks; Changing the Way We Work, Live, Play, and Learn, Empowering the Internet Generation, The Internet Economy, and The New Internet Economy are service marks; and ASIST, BPX, Catalyst, Cisco, Cisco IOS, the Cisco IOS logo, Cisco Systems, the Cisco Systems logo, the Cisco Systems Cisco Press logo, Enterprise/Solver, EtherChannel, EtherSwitch, FastHub, FastLink, FastPAD, FastSwitch, GeoTel, IOS, IP/TV, IPX, LightStream, LightSwitch, MICA, NetRanger, Post-Routing, Pre-Routing, Registrar, StrataView Plus, Stratm, TeleRouter, and VCO are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and certain other countries. All other trademarks mentioned in this document are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any of its resellers. (9909R)

Copyright © 2003, Cisco Systems, Inc.
All rights reserved.