Cisco Confidential.

DEVELOPER GUIDE FOR CISCO UNIFIED CM IM AND PRESENCE 10.5(2)

Last Updated: 15/12/2014

Table of Contents

Cisco Unified CM IM and Presence Interfaces

Overview of Interfaces

Cisco Unified CM IM and Presence supports the following interfaces to interoperate with external clients or applications:

SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE)

The SIMPLE-based interfaces for Cisco Unified CM IM and Presence provide the following functionality:

Availability (PUBLISH, SUBSCRIBE and NOTIFY interfaces)

Instant Messaging (Page mode MESSAGE)

Extensible Messaging and Presence Protocol (XMPP)

The XMPP Presence and Instant Messaging protocol is the core protocol on Cisco Unified CM IM and Presence. This interface provides instant messaging, availability and roster management services.

Bidirectional Streams over Synchronous HTTP (BOSH)

The BOSH interface allows allows integration with instant messaging, availability and roster management services from Cisco Unified CM IM and Presence into your web-based applications. Cisco provide the Cisco AJAX XMPP Library (CAXL) as a JavaScript based XMPP client library that allows developers to build web applications which utilize the BOSH interface

Client Configuration Web Service (SOAP)

The Client Configuration Web Service is an interface to Cisco Unified CM IM and Presence that allows client applications to manage user preference information such as contacts, presence rules, access control lists, and calendaring options. This web service is available via a SOAP interface.

Presence Web Service (SOAP/REST)

The Presence Web Service is an open interface that allows client applications to share user presence information with Cisco Unified CM IM and Presence. This interface is used by developers to build client applications that can send and receive user presence state updates. The web service is available via a SOAP interface and a REST (HTTP/XML) interface.

Roster Management via AXL-API (SOAP/REST)

Provide new features for roster management via AXL-API. This feature will enable customers to build roster management into their automated tooling . The AXL-API is the interface of choice on the IM&P associated product CUCM, for providing management interfaces.

Platform Administrative Web Services (PAWS)

The **Platform Administrative Web Service (PAWS)** is a XML/SOAP based interface that allows applications to **initiate and monitor upgrades** on multiple Unified Communications clusters from a single management client. The PAWS interface facilitates large scale Unified Communications deployments and upgrades. PAWS is a new publicly available interface in **Unified Communications 9.0(1).**

The following platform upgrade and administrative tasks can be performed using PAWS:

- Upgrade
- Switch version
- Reboot
- Get node information, such as version
- Cancel an upgrade installation that is in progress.
- Obtain hardware specifications.
- Retrieve installed options on active and inactive partitions.
- Determine which products are deployed.
- Reboot systems without switching partitions.
- Determine if an upgrade or a COP file is valid.

Documentation for PAWS is available at http://developer.cisco.com/web/paws-developer.

Audience

This document is intended for developers who write applications that extend the functionality of the web services and interfaces that are described in this document. This guide assumes the developer has knowledge or experience in the following areas:

http://www.w3.org/XML/ Extensible Markup Language (XML)

http://www.w3.org/Protocols/rfc2616/rfc2616.html Hypertext Transport Protocol (HTTP)

Simple Object Access Protocol (SOAP)

Web Service Definition Language (WSDL) 1.1

Session Initiation Protocol (SIP)

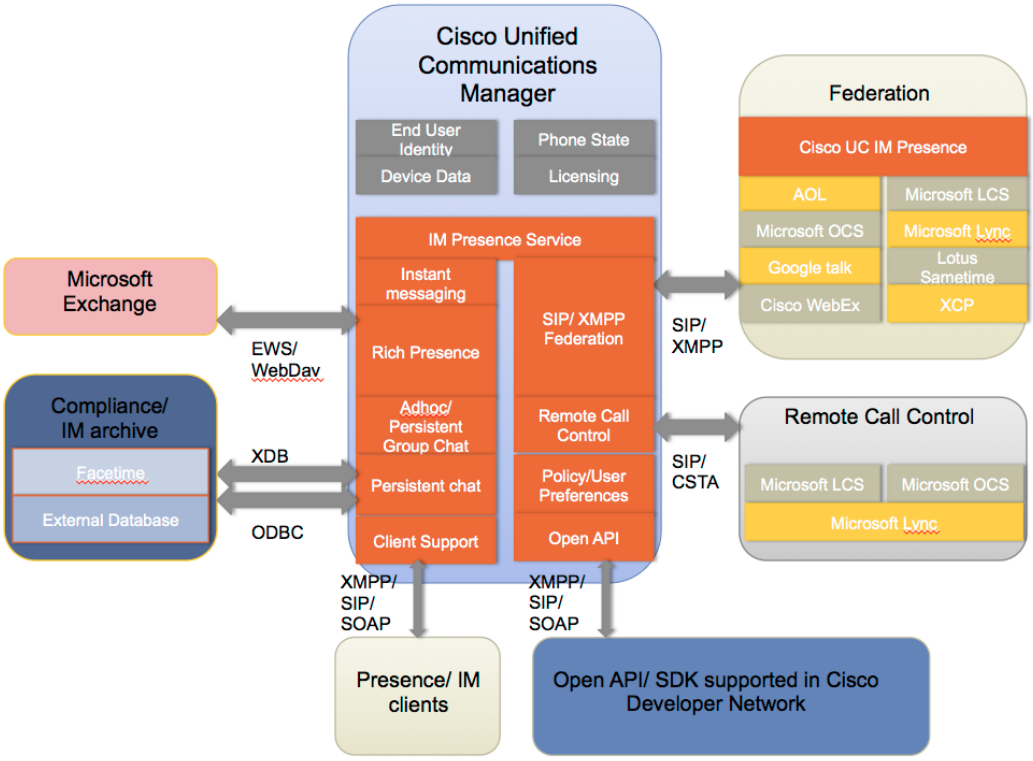
SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE)

Extensible Messaging and Presence Protocol (XMPP)

System Architecture

Figure 1 below shows the overall Cisco Unified CM IM and Presence system architecture.

Figure 1 Cisco Unified CM IM and Presence System Architecture



About the Extensible Messaging and Presence Protocol (XMPP) Interface

The basic XMPP protocol is specified in RFC-3920 and RFC-3921. In addition, there are many extension drafts described in XMPP Extension Protocole (XEPs). The following table lists these drafts and specifies the level of support. “Server Supported” indicates that Cisco Unified CM IM and Presence explicitly implements the XEP. “Server Compatible” indicates that Cisco Unified CM IM and Presence will route XML allowed by policy, including the protocol specified in the XEP.

Table 1 XMPP Extensions Support

XEP Extension	Server Supported	Server Compatible	SIP <> XMPP Conversion Supported	Federated SIP <>XMPP Conversion Supported
XEP-0004 Data Forms	Yes	Yes	No	No
XEP-0012 Last Activity	Yes	Yes	No	No
XEP-0013 Flexibility Offline Message Retrieval	No	No	No	No
XEP-0016 Privacy Lists	Yes	No	No	No
XEP-0020 Feature Negotiation	No	Yes	No	No
XEP-0022	No	Yes	Yes	Yes
XEP-0025 Jabber HTTP Polling	Yes	Yes	No	No
XEP-0027 Current Jabber OpenPGP Usage	No	Yes	No	No
XEP-0030 Service Discovery	Yes	Yes	Yes	Yes
XEP-0033 Extended Stanza Addressing	Yes	Yes	No	No
XEP-0045 Multi-User Chat	Yes	Yes	No	No
XEP-0047 In-Band Bytestreams	No	Yes	No	No
XEP-0048 Bookmarks	Yes	Yes	No	No
XEP-0049 Private XML Storage	Yes	No	No	No
XEP-0050 Ad-Hoc Commands	Yes	Yes	No	No
XEP-0054 Vcard-temp	No	No	No	No
XEP-0055 Jabber Search	Yes	Yes	No	No
XEP-0060 Publish-Subscribe	No	Yes	No	No
XEP-0065 SOCKS5 Bystreams	Yes	Yes	No	No
XEP-0066 Out of Band Data Archive OOB requests	Yes	Yes	No	No
XEP-0068 Field Standardization for Data Forms	Yes	Yes	No	No
XEP-0070 Verifying HTTP Requests via XMPP	Yes	Yes	No	No
XEP-0071 XHTML-IM	Yes	Yes	Yes	TBD
XEP-0072 SOAP over XMPP	Yes	Yes	No	No
XEP-0077 In-Band Registration	No	Yes	No	No
XEP-0080 User Location	Yes	Yes	No	No
XEP-0082 XMPP Date and Time Profiles	Yes	Yes	No	No
XEP-0083 Nested Roster Groups	Yes	No	No	No
XEP-0084 User Avatar	Yes	Yes	No	No
XEP-0085 Chat State Notifications	No	Yes	Yes	Yes

XEP-0092 Software Version	Yes	Yes	No	No
XEP-0093 Roster Item Exchange	No	Yes	No	No
XEP-0094 Agent Information	Yes	Yes	No	No
XEP-0095 Stream Initiation	No	Yes	No	No
XEP-0096 File Transfer	No	Yes	No	No
XEP-0106 JID Escaping	Yes	Yes	Yes	TBD
XEP-0107 User Mood	Yes	Yes	No	No
XEP-0108 User Activity	Yes	Yes	No	No
XEP-0114 Jabber Component Protocol	Yes	No	No	No
XEP-0115 Entity Capabilities	Yes	Yes	Yes	Yes
XEP-0118 User Tune	Yes	Yes	No	No
XEP-0122 Data Forms Validation	No	Yes	No	No
XEP-0124 Bidirectional Streams over Synchronous HTTP (BOSH)	Yes	No	No	No
XEP-0126 Invisibility	Yes	No	No	No
XEP-0127 Common Alerting Protocol (CAP) over XMPP	Yes	Yes	No	No
XEP-0128 Service Discovery Extensions	Yes	Yes	No	No
XEP-0131 Stanza Headers and Internet Metadata (SHIM)	No	Yes	No	No
XEP-0137 Publishing SI Requests	No	Yes	No	No
XEP-0138 Stream Compression	Yes	No	No	No
XEP-0141 Data Forms Layout	No	Yes	No	No
XEP-0144 Roster Item Exchange	No	Yes	No	No
XEP-0146 Remote Controlling Clients	No	Yes	No	No
XEP-0147 XMPP URI Scheme Query Components	No	No	No	No
XEP-0149 Time Periods	No	Yes	No	No
XEP-0153 Vcard-based Avatars	Yes	Yes	No	No
XEP-0155 Stanza Session Negotiation	No	Yes	No	No
XEP-0158 Robot Challenges	No	Yes	No	No
XEP-0160 Best Practices for Handling Offline Messages	Yes	No	No	No
XEP-0163 Personal Eventing Via PubSub	Yes	Yes	No	No
XEP-0166 Jingle	No	Yes	No	No
XEP-0170 Recommended Order of Stream Feature Negotiation	Yes	No	No	No
XEP-0171 Language Translation	No	Yes	No	No
XEP-0172 User Nickname	No	Yes	No	No
XEP-0178 Best Practices for Use of SASL EXTERNAL	Yes	No	No	No
XEP-0181 Jingle DTMF	No	Yes	No	No
XEP-0184 Message Receipts	No	Yes	No	No
XEP-0199 XMPP Ping	No	Yes	No	No
XEP-0201 Best Practices for	No	Yes	No	No

Message Threads				
XEP-0202 Entity Time	No	Yes	No	No
XEP-0203 Delayed Delivery	Yes	Yes	Yes	Not Applicable
XEP-0220 Server Dialback	Yes	No	No	No
XEO-0224 Attention	No	Yes	No	No
XEP-0230 Service Discovery Notifications	No	Yes	No	No
XEP-0233 Use of Domain-Based Service Names in XMPP SASL Negotiation	Yes	Yes	No	No
XEP-0245 The /me Command	No	No	No	No

About XMPP Security

For XMPP based clients such as CUPC 8, Cisco Unified CM IM and Presence will support either secure (default) and unsecure security modes.

- In Secure mode, TLS encryption is required. Cisco Unified CM IM and Presence will therefore send “starttls” to the client in the initial XMPP stream features, with an indication that TLS is required. The TLS version supported on Cisco Unified CM IM and Presence is TLSv1. SSLv3 and lower are not supported.
- In Unsecure mode, TLS encryption is not required, so Cisco Unified CM IM and Presence will not send “starttls” to the client in the initial XMPP stream features.

Cisco Unified CM IM and Presence also provides two different SASL authentication mechanisms when connecting over the XMPP Interface: PLAIN and CISCO_VTG_TOKEN.

- CISCO_VTG_TOKEN is a proprietary SASL mechanism where a one-time password retrieved over the Client Configuration Web Service is passed in SASL authentication over the XMPP interface.
- It provides a more secure authentication mechanism for clients using unencrypted connections. This SASL mechanism is described in more detail later.
- CUPC 8 Clients use the CISCO_VTG_TOKEN SASL mechanism though most 3rd Party XMPP Clients use the SASL PLAIN mechanism.

The Cisco Unified CM IM and Presence authentication component will validate users are enabled from licensing data maintained in the Cisco Unified CM IM and Presence database. An Authentication failure will be returned to the client if that validation fails.

CUPC8 Authentication Sequence

1. CUPC 8 Client will connect to Cisco Unified CM IM and Presence via the Client Configuration Web Service and login.
2. Client will request and receive a token/one time password from Cisco Unified CM IM and Presence over the the Client Configuration Web Service.
3. The Client will then use token/one time password to create the CISCO_VTG_TOKEN : Base64(userid=user@domain, NULL, token=one-time-password).
4. The Client will then connect to Cisco Unified CM IM and Presence as an XMPP Client.
5. When an XMPP Client such as CUPC 8 connects to Cisco Unified CM IM and Presence, the server will return the supported SASL authentication mechanisms in in stream features:

```
<mechanisms>
  <mechanism>PLAIN</mechanism>
  <mechanism>CISCO_VTG_TOKEN</mechanism>
</mechanisms>
```

6. CUPC8 will choose the generated CISCO_VTG_TOKEN during SASL authetication. The Client's request will be of the following format:

```
<auth xmlns='urn:ietf:params:xml:ns:xmpp-sasl' mechanism='CISCO-VTG-TOKEN'>
  dXNlcmlkPWp1bGlldEBjYXB1bGV0LmNvbQB0b2tlbj0yMzQ1Njc4
</auth>
```

Where "dXNlcmlkPWp1bGlldEBjYXB1bGV0LmNvbQB0b2tlbj0yMzQ1Njc4" is Base64(userid=user@domain, NULL, token=one-time-password).

7. The Cisco Unified CM IM and Presence Authentication Component will parse the Base 64 CISCO_VTG_TOKEN received from the Client for the User Id and token to search the database for a match. If there is no matching entry, Cisco Unified CM IM and Presence will respond to Client with an Authentication Failure ERROR. If the token has expired (over 10 minutes old), Cisco Unified CM IM and Presence will respond to the Client with an Authentication Failure ERROR. The authentication error will include a token expiration error code, like this:

```
<failure xmlns='urn:ietf:params:xml:ns:xmpp-sasl'>
  <temporary-auth-failure/>
  <token-expired xmlns='http://protocols.cisco.com/vtg/token'/>
</failure>
```

8. The Cisco Unified CM IM and Presence Authentication Component will query the database for the license status of the User. If the User is not enabled, Cisco Unified CM IM and Presence will respond to the Client with an ERROR.
9. Upon successful completion of Client Authentication, Cisco Unified CM IM and Presence will return success to the client as shown below:

```
<success xmlns='urn:ietf:params:xml:ns:xmpp-sasl'/>
```

10. As this token/1-time password is only good from one use.

3rd Party XMPP Client Authentication Sequence

1. When an XMPP Client connects to Cisco Unified CM IM and Presence, the server will return the supported SASL authentication mechanisms in in stream features:

```
<mechanisms>
    <mechanism>PLAIN</mechanism>
    <mechanism>CISCO_VTG_TOKEN</mechanism>
</mechanisms>
```

2. The Client will reply to these SASL offering by sending an auth element that contains the SASL Plain mechanism and a Base64 encoding of the username and password. An example is shown below where "*AGp1bGlldABwYXNzd29yZA==*" is Base64 (user: juliet, password: password):

```
<auth xmlns='urn:ietf:params:xml:ns:xmpp-sasl' mechanism='PLAIN'>
  AGp1bGlldABwYXNzd29yZA==
</auth>
```

3. The Cisco Unified CM IM and Presence Authentication Component will parse the Base 64 encoded Username and Password pair received from Client for the User Id and password to search the database for a match. If there is no matching entry, Cisco Unified CM IM and Presence will respond to Client with an Authentication Failure ERROR.
4. The Cisco Unified CM IM and Presence Authentication Component will query the database for the license status of the User. If the User is not enabled, Cisco Unified CM IM and Presence will respond to the Client with an ERROR.
5. If the Authentication Component is able to retrieve a match the Authentication Component will reply with a SUCCESS message as follows:

```
<success xmlns='urn:ietf:params:xml:ns:xmpp-sasl'/>
```

About XMPP Presence

An XMPP client send presence stanza with the device status according to RFC-3920 and RFC-3921. From 10.5 XMPP clients are able to connect to IM&P over both IPv4 and IPv6.

The manually set status of a user is accomplished by the semantics in section **XMPP Rich Presence**.

An XMPP client supporting Rich Presence should always display the composed rich presence received from the server as it's own status rather than the device status of the client.

XMPP Rich Presence

Cisco Unified CM IM and Presence supports Rich Presence over the XMPP interface as well as the SIMPLE interface. The standard XMPP protocol will be used to deliver presence information, but it will be enhanced with embedded pidf XML content within the XMPP Presence stanzas. The composed presence will always have a priority of 127 and have a node name of http://cisco,com/cup/caps so that it can be distinguished from other device only XMPP presence stanzas. Clients wishing to take advantage of the Rich Presence information can use this pidf information to accomplish that. The main pidf elements described in the SIMPLE sections of the document apply here as well.

```
<presence from='cupuser@cisco.com/composed'>
  <c hash='sha-1' node='http://cisco.com/cup/caps' ver='uyV+h3Zn3t9eUgEvEULLrTGjzlc='
xmlns='http://jabber.org/protocol/caps'/>
  <priority>127</priority>
  <presence entity='sip:cupuser@cisco.com' xmlns='urn:ietf:params:xml:ns:pidf'>
    <person id='cupuser' xmlns='urn:cisco:params:xml:ns:pidf:rpid'>
      <activities>
        <available/>
        <phone-status>unavailable</phone-status>
        <im-status>available</im-status>
      </activities>
    </person>
    <display-name xmlns='urn:ietf:params:xml:ns:pidf:cipid'>
      yasong@cisco.com
    </display-name>
    <tuple id='JabberJabber MomentIM' xmlns='urn:ietf:params:xml:ns:pidf'>
```

```
      <status>
        <basic>open</basic>
      </status>
      <servcaps xmlns='urn:ietf:params:xml:ns:pidf:servcaps'>
        <type>text/plain</type>
        <type>application/x-cisco-cupc+xml</type>
        <text>true</text>
      </servcaps>
    </tuple>
  </presence>
</presence>
```

XMPP Specification for Manually Set Presence

- A manual status received as embedded pidf in the XMPP will have no expiration and not be tied to the device
- Manual Presence should be sent as directed presence to the full jid of the Cisco Unified CM IM and Presence composed resource for self – so that it is not broadcast to watchers. Cisco Unified CM IM and Presence will then be responsible for sending the resulting composed presence update out to watchers.
    - Full jid resource name is configurable on the Cisco Unified CM IM and Presence system. The client can determine the resource name by looking at the presence stanzas with a hard-coded node name used in the presence stanza. node="http://cisco.com/cup/caps"
- The entity tag in the embedded pidf should have a SIP Url that contains the userID@CUP-domain that corresponds to the user setting the manual presence status
- The XMPP device status need not be sent in the presence stanza, as the manual presence update will be stored separately and does not effect the device presence state.
- The manual status will persist past logout unless the client explicitly clears it as part of it's logout processing. The client should clear DND at logout if it is the device that set it, but not if it is set by some other source. The client should not clear the vacation state at logout. The client should clear any other manually set state upon logout.
- The manual status pidf will only contain <person> information, NO <tuple> or <device> information
- The manual status pidf will have an rpid <class> set to "manual"
- In order to set to " available", a manual PUBLISH will be sent with a <class> of manual, and an "available" indication. This is used as a temporary clearing of ALL other statuses to available(manual or derived), it will not have an infinite expiration time like the other manually set statuses

  In order to explicitly "clear" a manually set status completely (so that only derived status is used, the embedded pidf should be sent with a <class> of manual, and a <clear> element.

Example of embedded pidf for a client to manually set a user to unavailable:

```
<presence entity="sip:donalync@ciscotest.com" xmlns="urn:ietf:params:xml:ns:pidf">
  <person xmlns:dm="urn:cisco:params:xml:ns:pidf:rpid" id="donalync">
    <activities>
```

```
      <unavailable/>
    </activities>
    <class>manual</class>
  </person>
</presence>
```

To change to manually set “busy”, use the above pidf but with the following modification:

```
<activities>
  <busy/>
</activities>
```

To change to manually set “DND”, use the above pidf but with the following modification:

```
<activities>
  <dnd />
</activities>
```

To change to manually set “available”, use the above pidf but with the following modification:

```
<activities>
  <available/>
</activities>
```

An explicit “clear” would look like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<presence entity="sip:donalync@ciscotest.com" xmlns="urn:ietf:params:xml:ns:pidf">
  <person xmlns:dm="urn:cisco:params:xml:ns:pidf:rpid" id="donalync">
    <activities>
      <clear/>
    </activities>
    <class>manual</class>
  </person>
</presence>
```

Privacy Policy

The basis for the privacy implementation is specified in XEP-0016. The following highlight some more specific expectations of clients interfacing with the Cisco Unified CM IM and Presence 8.5 server.

- As the rules are processed in order, the server will stop processing the list when it matches a rule in the list; items in the list must be ordered so that the correct behavior is achieved. For example in the extract below:

```
<item type='jid' value='user@feddomain.com' action='allow' order='1'/>
<item type='jid' value='feddomain.com' action='deny' order='2'/>
```

‘feddomain.com’ is blocked but ‘user@feddomain.com’ is allowed; the desired behavior is that the rule for ‘user@feddomain.com’ be processed first, otherwise the stanza addressed to this federated user will be blocked by a match on ‘feddomain.com’.

- The default policy should have the highest numerical order in the list so that it is only processed if there is no match for any other rule.
- Clients should request and cache the privacy list when it is updated by other client sessions; each active session will receive a privacy list "push" when the other active session has modified the privacy list.

A client session must not delete the 'cisco-default' privacy list, un-set it as the default list; set another list as the default list, or set any list as the active list.

Auto-Authorization

Cisco Unified CM IM and Presence can be configured to auto-authorize presence subscriptions from internal (within domain) users. If this is set, the clients will not receive subscription requests for internal users before presence is sent to the requesting watcher. If this is not set, the client will receive subscription requests and should prompt the user for authorization.

External users requesting presence will not be auto-authorized by the server unless they were previously added to the user's allowed list (via XEP-0016).

Out of Office

The Out of Office status text may be added to an Unavailable/Offline Presence status prior to logging the client out. The status and it's text should override other status updates from the user's other devices/calendar for the duration it is set, until it is cleared by the client that set it. The text also needs to be displayed on all watchers of this user, including 3rd Party XMPP clients. In order to accomplish this, the client sends a manual presence with the pidf of unavailable wrapped in an XMPP presence stanza that contains the XA (Extended Away) state. The XA state was used rather than the unavailable XMPP presence status because:

a) an unavailable presence status does not have a meaningful priority and therefore would not override other devices

b) Status text associated with an unavailable presence stanza is generally not displayed by 3rd party XMPP clients

The following is an example of an out of office presence stanza from the client:

```
<presence from='aa@cisco.com/cucsf39154' id='vijes-14' to='aa@cisco.com/composed'>
  <show>xa</show>
  <presence entity='sip:aa@cisco.com' xmlns='urn:ietf:params:xml:ns:pidf'>
    <person id='aa'>
      <activities>
        <unavailable/>
      </activities>
      <class>manual</class>
      <note>Out of office</note>
    </person>
  </presence>
</presence>
```

The composed presence that results from this setting will also be sent in an XA XMPP presence stanza. This results in a presence status of XA with the Out of office text displayed on 3rd Party XMPP client watchers, and a presence status of Unavailable/Offline displayed on clients that support the embedded pidf presence extensions.

When the user logs into the Out of Office client, it must send a “clear” of the manually set status to remove the Out of Office status and return to available.

Derived Statuses

When a system derived status such as “on-the-phone” or “meeting” are manually overridden by the user on their client, the composed presence will contain a <derived> element within the <person> element that informs the client what the derived status would be if not being overridden by a manual status. The client can use this information to contextually display the derived status as a menu option. If the derived status is then selected from the menu, the client should issue a manual state “clear” stanza so that it is no longer overriding the system derived status.

About XMPP Instant Messaging

The basic XMPP messaging is supported per RFC-3920 and RFC-3921. From 10.5 basic XMPP messaging supports both IPv4 and IPv6.

IM Routing

Cisco Unified CM IM and Presence will route an IM request to all of a user’s IM devices that are logged in an have published presence with a priority attribute greater than -1, not just to the device which has published the highest priority presence. Once one of the user’s devices responds, the initiating device should direct the subsequent IM messages to the full resource jid of the answering client until the IM session is over. In order to achieve this it is important that the XMPP client always includes its full resource jid in the from header of message stanzas.

When the client determines a session is ended, any further utterances from the other party should be sent back to the bare jid, which will cause the messages to be sent to all the user’s devices. An IM session should be considered ended by a client when:

- A XEP-85 “gone” message is received from the remote client
- A presence state change for the remote client is received
- The local IM window of the user is closed
- Note that a <gone> will be generated by the server to SIP clients when there is a presence status change for one of the users involved in the chat.

Keyboard Activity

Keyboard Activity is pass through from a server perspective, Cisco Unified CM IM and Presence clients supporting typing information should be communicating using <composing/> message as defined in Chat State Notifications (XEP-0085) at a minimum.

If a client wishes to maintain compatibility with older clients then it will also need to support <composing/> message as defined in Message Events (XEP-0022), which is a deprecated protocol, replaced by XEP-0085.

IM Message Content

Rich text format content is defined in XHTML-IM (XEP-0071). The Cisco Unified CM IM and Presence server does not support content normalization so any rich text format IM **must** also be sent in plain text form.

Blocking IM

Blocking IMs by the server is supported via XEP-0016. All non-privacy list related IM blocking must be done by the client.

Preventing IM Delivery

If a client does not want to receive IM it should publish a negative presence priority. For Cisco Unified CM IM and Presence 8.5 it is requested clients publish a presence priority of -2 if they do not wish to receive IM. The value of -1 is reserved for the composed presence stanza when there is no logged in IM device. Presence will not work properly if the client uses -1.

Note: This only applies to IM's addressed to a user's bare-jid, if an IM is sent to a user's resource jid it will still be delivered as defined in XMPP RFC.

Delayed/Offline IM

If there are no clients available to receive IM (due to them being logged out or publishing -2 priority) on behalf of a user, the IM's will be stored on the server for delivery when a client is available to receive IM. XEP-0203 is supported to provide this functionality.

Message Priority

Message Priority (a client side feature and not to be confused with presence priority) indicates the priority of an individual IM, similar in concept to priority in email. This is also referred to as Urgency and is defined in XEP-0131: Stanza Headers and Internet Metadata .

Broadcast IM

Broadcast IM is a client-side feature, essentially the client sends a point to point IM to each user individually. For Cisco Unified CM IM and Presence, the recommendation is to use the following format for a broadcast message which shows 2 message stanza's being generated to send a broadcast message to 2 recipients. It is important to note a message containing a <address type='noreply'/>, is identifiable as a broadcast that cannot be replied to. This is purely used to drive the UI of the broadcast message on the recipient clients (e.g. to decide to display a reply button on not) and should not prevent the recipient for sending a point to point IM to the sender if they choose to by normal means.

```
<message id="" to="phession@cisco.com">
  <addresses xmlns="http://jabber.org/protocol/address">
    <address jid="phession@cisco.com" type="to"/>
    <address type="noreply"/>
  </addresses>
  <subject>TEST MESSAGE</subject>
  <body>blah blah blah</body>
  <html xmlns="http://jabber.org/protocol/xhtml-im">
    <body xmlns="http://www.w3.org/1999/xhtml">
      <p>blah blah blah</p>
    </body>
  </html>
</message>
```

```
<message id="" to="pabarry@cisco.com">
  <addresses xmlns="http://jabber.org/protocol/address">
    <address jid="pabarry@cisco.com" type="to"/>
    <address type="noreply"/>
  </addresses>
  <subject>TEST MESSAGE</subject>
  <body>blah blah blah</body>
  <html xmlns="http://jabber.org/protocol/xhtml-im">
    <body xmlns="http://www.w3.org/1999/xhtml">
      <p>blah blah blah</p>
    </body>
  </html>
</message>
```

XMPP Text Chat Restrictions

Cisco Unified CM IM and Presence somewhat restricts the configuration of Text Chat. While most of the more common settings can be configured (database connection information, number of messages to display per room, etc), there are some parameters that are set and cannot be changed that can impact the Client user experience. The following server-side configuration options are pre-configured by CISCO UNIFIED CM IM AND PRESENCE.

Table 2 XMPP Pre-configurred Text Chat Parameters

Parameter Topic	Parameter	Pre-configured Values
Persistenence	Restrict persistent room creation to TC administrators only	No
	How many messages can be retrieved from the archive at once	50
	Can room owners change this setting when configuring a room	No
	Timeout value for persistent rooms	0
Membership	Are rooms for members only by default	No
	Can room owners change this setting when configuring a room	Yes
	Only moderators can invite people to members-only rooms	Yes
	Can room owners change this setting when configuring a room	Yes
	Can users add themselves to rooms as members	No
	Can room owners change this setting when configuring a room	No
Presence	Should members and administrators who are not in a room still be visible in the room	Yes
	Can room owners change this setting when configuring a room	Yes
	Should rooms be backwards compatible with older clients	No
	Can room owners change this setting when configuring a room	No
	Should rooms be anonymous by default	No
	Can room owners change this setting when configuring a room	No
Occupancy	How many users can be in a room at one time	100
	How many hidden users can be in a room	0
	What is the default maximum occupancy for a room	50
	Can room owners change this setting when configuring a room	Yes
Invite	What is the lowest participation level a user can have to invite others to the room	Participant
	Can room owners change this setting when configuring a room	No
Password	Enabled / Disabled	Enabled
Message	What is the lowest participation level a user can have to change a room's subject	Participant
	Can room owners change this setting when configuring a room	Yes
	What is the lowest participation level a user can have to send a private message within the room	Visitor
	Can room owners change this setting when configuring a room	Yes
	Remove all XHTML formatting from messages	No
	Can room owners change this setting when configuring a room	No
Moderation	Are rooms moderated by default	No
	Can room owners change this setting when configuring a room	Yes
History	How many previous messages can display in a room	100
	Can room owners change "How many previous messages display in a room by default" when configuring a room	No

About the Bidirectional-stream Over Synchronous HTTP (BOSH) Interface

The Cisco Unified CM IM and Presence BOSH interface allows allows integration with instant messaging, availability and roster management services from Cisco Unified CM IM and Presence into your web-based applications. It is based on the following XMPP extensions:

- XEP-0124: Bidirectional-streams Over Synchronous HTTP (BOSH)
- XEP-206: XMPP Over BOSH

As This interface provides a HTTP Binding for XMPP communication, all the same capability available to all XMPP clients is also available to BOSH based clients.(BOSH based clients will support both IPv4 and IPv6 addresses in the 10.5 release)

Cisco provide the Cisco AJAX XMPP Library (CAXL) as a JavaScript based XMPP client library that allows developers to build web applications which utilize the BOSH interface.

Cisco AJAX XMPP Library (CAXL)

The Cisco AJAX XMPP Library (CAXL) is a JavaScript based XMPP client library that allows integration with instant messaging, availability and roster management services from Cisco Unified CM IM and Presence into your web-based applications.

CAXL is an object-oriented, client side library that utilizes BOSH (Bidirectional-stream Over Synchronous HTTP) technology as an HTTP binding for XMPP communication. This is useful in situations where a device or client is unable to maintain a long-lived TCP connection to an XMPP server (e.g. a web browser).

For further information or to access this library and associated documentation, please download the Cisco AJAX XMPP Library package from the Cisco Developer Network.

About the Client Configuration Web Service

The Client Configuration Web Service is an interface that allows client applications to manage user preference information on Cisco Unified CM IM and Presence. The Client Configuration Web Service provisions information such as contacts, contact groups, presence rules, access control lists, and calendaring options on Cisco Unified CM IM and Presence.

The Client Configuration Web Service is a Simple Object Access Protocol (SOAP) interface. The request and response messages are sent in the form of XML. A client application sends a SOAP request; the web service processes the request and sends a SOAP response.

Overview of Functions

A client application uses the Client Configuration Web Service to perform the following functions:

- Log in and out of Cisco Unified CM IM and Presence
- Get SASL Cisco-VTG-Token
- Get system configuration information
- Get and set user configurations
- Contact list management
- Get a list of federated domains
- Download dialing rules
- Get licensing features
- Get, add and delete the Access Control Lists (ACL)
- Get and set calendaring options
- Get and set phone presence option
- User public data search
- Get Cluster Information

What has changed in this release of Cisco Unified CM IM and Presence

SOAP interface version 9.0 has the following modifications.

1. Changed the return parameters of
    a. getAllConfig
    b. getSystemConfig
    c. getUserConfig
2. Changed the way the following API is invoked
    a. getAllConfig
3. Deprecated the following APIs
    a. getDiallingRules
    b. getLicensingFeatures

SOAP interface version 10.5 has the following modifications.

1. Changed the return parameter for getAllConfig
2. Modified the procedure – UCSoapProcedures.

These changes does not impact SOAP interface versions 7 & 8 as they are still supported as they were on previous release of CISCO UNIFIED CM IM AND PRESENCE release 9.0. Do refer the section which details each API to understand what has changed.

Login and Authentication

To log in to the Client Configuration Web Service, a client application sends a username and password in the login request for authentication. Cisco Unified CM IM and Presence creates a session key, an opaque string, if the user credentials are verified. The client includes the session key in the SOAP header portion of subsequent SOAP requests to the Client Configuration Web Service, including logout requests.

The Client Configuration Web Service supports the following login scenarios:

The client application sends a username and password for an end user in the login request for authentication, and Cisco Unified CM IM and Presence returns a unique session key for the end user.

The client application logs in to Cisco Unified CM IM and Presence as an "application user" using an application username and password. The client application can then log in an end user by passing the session key for an application user, and the username for an end user in the login request; a unique session key for the end user is returned.

If an application user is disconnected, any associated end users are automatically logged out.

The Client Configuration Web Service supports multiple logins for the same user from different client applications. Cisco Unified CM IM and Presence stores a separate session key per login to each client application for a user. Refer section Client Type under login request for more information.

Contact Management

Using the Client Configuration Web Service, users can manage their non-presence contacts. In addition, users can also add and modify the auxiliary information associated with a contact such as home number, work number, mobile number, email and so on.

The following contact management functionality is supported:

Add a contact

Modify a contact

Delete a contact

Get auxiliary information for a contact

Modify auxiliary information for a contact

Delete auxiliary information for a contact

Federated Domain Support

The Client Configuration Web Service allows users of the client application to interact with users in permitted foreign domains (known as federated domains). Since the Client Configuration Web Service supports contacts from federated domains, not all contacts in the user contact list will be Cisco Unified CM IM and Presence users.

The client application uses the get-federated-domains request to obtain a list of foreign domains (configured on Cisco Unified CM IM and Presence) to which federation is permitted or blocked. This list informs the user which foreign domains, and foreign contacts, they can successfully interact with.

The client application can authorize or deny foreign watchers using the Access Control Lists (ACL). When a user approves or denies a foreign watcher, the client application uses the add-acl request to add the user to the ACL with a policy of allowed or denied, and sends the updated ACL to Cisco Unified CM IM and Presence.

About the SOAP Messages

Accessing the Web Service

To improve backward compatibility, versions of the Client Configuration Web Service are numbered in accordance with Cisco Unified CM IM and Presence releases. Each version of the Client Configuration Web Service is accessed by appending a version number to the following URI:

`https://server_name/EPASSoap/service/<version URI>`

Cisco Unified CM IM and Presence accepts SOAP requests both on the interface version specified in the URI, and on the previous version of the interface. For example, Cisco Unified CM IM and Presence version 9.0 will accept SOAP requests from the following URIs:

https://server_name/EPASSoap/service/v90

https://server_name/EPASSoap/service/v80
https://server_name/EPASSoap/service/v70

Cisco Unified CM IM and Presence 9.0 and above also accepts SOAP requests when an unversioned URI is specified.

https://server_name/EPASSoap/service

When above URI is specified, the oldest SOAP interface of a given release is selected. Howerver different the APIs across each version may be login API is the same across all versions and the login response contains information on the Client Configuration Web Service versions available on the Cisco Unified CM IM and Presence server.

https://server_name/EPASSoap/service/latest

When above URI is specified, the latest SOAP interface of a given release is selected.

Following a successful login response, the client selects which version of the Client Configuration Web Service they wish to use for subsequent methods.

The Client Configuration Web Service is available over HTTPS on port 8443 for IPv4 and on port 443 for IPv6 in the 10.5 release.

Accessing the SOAP Schema

To access a description of the latest Cisco Unified CM IM and Presence SOAP schema, enter the following URL at a computer that has access to your Cisco Unified CM IM and Presence server:

http://server_name/EPASSoap/service/version?wsdl

http://server_name/EPASSoap/service?wsdl

http://server_name/EPASSoap/service/latest?wsdl

Note: Server_name is the the hostname or the IPv4/IPv6 (IPv6 support added in 10.5) address of the Cisco Unified CM IM and Presence server.

Eg:

http://server_name/EPASSoap/service/v90?wsdl

Provides the WSDL of SOAP interface version 9.0

http://server_name/EPASSoap/service/version?wsdl

Provides the WSDL of the oldest SOAP interface version for a given Cisco Unified CM IM and Presence release

http://server_name/EPASSoap/service/version?wsdl

Provides the WSDL of the latest SOAP interface version for a given Cisco Unified CM IM and Presence release

SOAP Header

All SOAP requests, except login requests, must insert a session-key element into the SOAP header portion as shown in Example 1.

Example 1 SOAP Header

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
          xmlns:xsd="http://www.w3.org/2001/XMLSchema"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xmlns:epas="urn:cisco:epas:soap"
          xmlns="urn:cisco:epas:soap">
  <SOAP:Header>
    <epas:session-key>session key</epas:session-key>
  </SOAP:Header>
  <SOAP:Body>
    <!-- a non login request body here -->
  </SOAP:Body>
</SOAP:Envelope>
```

SOAP Fault

A SOAP fault can be returned in the response body due to either a non-conformant SOAP request or an internal server error, for example, failure to connect to database.

Example 2 indicates the client is sending an invalid request.

Example 2 SOAP Fault Caused by an Invalid Request

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
    <env:Fault>
      <env:Code><env:Value>env:Sender</env:Value></env:Code>
      <env:Reason>
        <env:Text xml:lang="en">Invalid request.</env:Text>
      </env:Reason>
    </env:Fault>
  </env:Body>
</env:Envelope>
```

Example 3 indicates the Cisco Unified CM IM and Presence server is having difficulty processing requests because of database failure.

Example 3 SOAP Fault Caused by a Database Failure

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
    <env:Fault>
      <env:Code><env:Value>env:Receiver</env:Value></env:Code>
      <env:Reason>
        <env:Text xml:lang="en">Could not connect to database.</env:Text>
      </env:Reason>
    </env:Fault>
  </env:Body>
</env:Envelope>
```

About the Client Configuration Web Service Functions

Login Request

Client-Type

From Cisco Unified CM IM and Presence 8.6.1 above, Clients including application servers **MUST** include the client type attribute when a login request is made. This is important with regards to

licensing verification, and to allow a user to login from multiple clients, at the same time allow a single login from the same client.

1. Application Servers logging in as admin users SHOULD use '**thirdpartyapp'** as the client type
2. End user logging in by passing in session key of an application user SHOULD use '**thirdclient'** as the client-type.
3. If applications chooses to login to as a standalone client, SHOULD apply use '**thirdstandalone'** as the client-type.

Note: A user logging in over 2 clients using the same client-type could result in an existing session being invalidated or undesirable data writes if 2 clients happen to use the same session id. Refer 'FORCE' section below to understand how a new session id is created for each login if not session id be reused. Through the SOAP interface handles a user logged in from multiple clients of the same type sharing the same session id, its recommended only once client per type be logged in at any given time.

This request logs in an application user to Cisco Unified CM IM and Presence.

Example 4 shows a sample login request made by an application user.

Example 4 Login Request

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:epas="urn:cisco:epas:soap"
            xmlns="urn:cisco:epas:soap">
  <soapenv:Body>
    <login client-type="thirdpartyapp">
      <username>thirdpartyappadmin</username>
      <password>12345</password>
    </login>
  </soapenv:Body>
</soapenv:Envelope>
```

User '**thirdpartyadmin'** is an admin user with certain role previleges provided by Cisco Unified CM IM and Presence Administrator. When an application user logs into Cisco Unified CM IM and Presence, as suggested above, '**thirdpartyapp'** SHOULD be used as the client type. Application users should be added and or updated from Cisco Unified CM IM and Presence Administration using Application User Configuration page. Refer Cisco Unified CM IM and Presence Configuration guide or Cisco Unified CM IM and Presence Administration online help to understand how an application user is added. Supported application user roles are.

1) Admin-3rd Party API
2) Admin-CUMA

Item 2 is reserved for a Cisco internal application server application and SHOULD NOT be used for applications developers of Cisco Developer Network but **Item 1** MAY be used by application

developers of CDN. An application user SHOULD NOT be assigned to more than one group which will result in undesired behaviour.

A successful login response contains a "service-version" parameter indicating the available versions of the Client Configuration Web Service that the client application can connect to. The client then selects the service version to use for all subsequent methods by appending the service version to the URI e.g. https://cup:8443/EPASSoap/service/v90 - for IPv4
https://[cup]:443/EPASSoap/service/v90 - for IPv6

If login is successful, the client receives a response that contains a session key, as shown in Example 5.

Example 5 Successful Login Response

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
    <epas:login-resp  xmlns:epas="urn:cisco:epas:soap" server-version="9.0.1.10000(30)">
      <epas:success>
        <epas:session-key>
          b189aa60-24e3-4e66-a3b6-d8488235ba47
        </epas:session-key>
        <epas:service-versions>
          <epas:version>v70</epas:version>
                <epas:version>v70a</epas:version>
          <epas:version>v80</epas:version>
                <epas:version>v90</epas:version>
        </epas:service-versions>
      </epas:success>
    </epas:login-resp>
  </env:Body>
</env:Envelope>
```

The request below is an example of an end user logging into Cisco Unified CM IM and Presence.

Example 6 Login Request

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:epas="urn:cisco:epas:soap"
            xmlns="urn:cisco:epas:soap">
  <soapenv:Body>
    <login client-type="thirdstandalone">
      <username>joseseba</username>
      <password>12345</password>
    </login>
  </soapenv:Body>
```

```
</soapenv:Envelope>
```

User ‘joseseba’ is an enduser who is already licensed for Cisco Unified CM IM and Presence by the administrator. When an end user logs into Cisco Unified CM IM and Presence as a standalone client, as suggested above, ‘**thirdstandalone’** SHOULD be used as the client type.

If the authentication is unsuccessful, the client receives a login response as shown in Example 7.

Example 7 Unsuccessful Login Response

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
    <epas:login-resp  cup-version="9.0.1.10000(30)" xmlns:epas="urn:cisco:epas:soap" >
      <epas:failure>
        <epas:reason>Wrong username/password</epas:reason>
      </epas:failure>
    </epas:login-resp>
  </env:Body>
</env:Envelope>
```

Client-Version

Clients MAY include a client version. Supported client version is a four part version eg: 1.0.0.0. Cisco Unified CM IM and Presence recommends all clients use this attribute which has the following benefits

1. Possiblity to limit access to client versions unsupported in a given release if needed. If you wish to enable login restriction for a specific client version, please contact your Cisco Support representative.
2. Cisco Unified CM IM and Presence Administration displays the version of logged in clients for a given user which is retrived from the version send within the SOAP login request.

Force

Clients MAY include a force flag for the following reasons.

1. Setting “force=true” will clean up an old session for a user for a given client-type and Cisco Unified CM IM and Presence will return a new session id after deleting an existing session
2. Setting force-false, Cisco Unified CM IM and Presence returns an existing session id if exists if not return a new session key as long as user credentials are correct..

   Example 8 shows a login request with the "force" attribute.

Example 8 Login Request with the Force Attribute

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

```
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:epas="urn:cisco:epas:soap"
            xmlns="urn:cisco:epas:soap">
  <soapenv:Body>
    <login client-type="thirdstandalone" force="true">
      <username>xmzhou</username>
      <password>12345</password>
    </login>
  </soapenv:Body>
</soapenv:Envelope>
```

A redirect login response is issued if a login request is sent to a Cisco Unified CM IM and Presence node that does not contain the presence information for a client. The redirect login response contains a primary and backup server address that can be used by the client to log in to the correct node. The redirect login response consists of a login failure with the failure reason set to "redirection".

Example 9 Login Redirect

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
    <epas:login-resp cup-version="9.0.1.10000(30)"   xmlns:epas="urn:cisco:epas:soap" >
      <epas:failure server.primary="otherclusternode1" server.backup="otherclusternode2" >
        <epas:reason>redirect [server.primary="otherclusternode1"
server.backup="otherclusternode2" ]</epas:reason>
      </epas:failure>
    </epas:login-resp>
  </env:Body>
</env:Envelope>
```

For failover purposes, the client may be configured with a primary server and a backup server on Cisco Unified CM IM and Presence.

Login Request (Application User logging in End User)

This request is used for an application user logging in an end user to Cisco Unified CM IM and Presence. The application user must first have logged into Cisco Unified CM IM and Presence and obtained a session key. This session key is then passed in the request to login the end user.

Example 10 Login request (End User logged in by Application User)

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:urn="urn:cisco:epas:soap">
  <soap:Header></soap:Header>
```

```
  <soap:Body>
    <login client-type="thirdclient">
      <urn:username>josdugga</urn:username>
      <urn:app-session-id>
        12d8d6d9-9099-05fb-f8fb-f82a6a68cecd
      </urn:app-session-id>
    </login>
  </soap:Body>
</soap:Envelope>
```

Example 11 Login response (End User logged in by Application User)

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
    <epas:login-resp cup-version="9.0.1.10000(30)" xmlns:epas="urn:cisco:epas:soap" >
      <epas:success>
        <epas:session-key>
          b189aa60-24e3-4e66-a3b6-d8488235ba47
        </epas:session-key>
        <epas:service-versions>
          <epas:version>v70</epas:version>
          <epas:version>v80</epas:version>
                <epas:version>v90</epas:version>
              </epas:service-versions>
      </epas:success>
    </epas:login-resp>
  </env:Body>
</env:Envelope>
```

Logout Request

To log out, the client sends a session key in the header portion of the SOAP request in addition to the logout element in the body portion. Example 12 shows a sample logout request.

Example 12 Logout Request

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:epas="urn:cisco:epas:soap"
            xmlns="urn:cisco:epas:soap">
  <soapenv:Header>
    <session-key>b189aa60-24e3-4e66-a3b6-d8488235ba47</session-key>
  </soapenv:Header>
  <soapenv:Body>
    <logout tuple-id="cupc-override"/>
  </soapenv:Body>
</soapenv:Envelope>
```

If logout is successful, the client receives a response similar to Example 13.

Example 13 Successful Logout Response

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
    <epas:logout-resp xmlns:epas="urn:cisco:epas:soap" >
      <epas:success/>
    </epas:logout-resp>
  </env:Body>
</env:Envelope>
```

SASL Cisco VTG-Token Functions

Cisco Unified CM IM and Presence 8.0 supports integration with Extensible Messaging and Presence Protocol (XMPP) enabled clients. Such clients will be authenticated using Simple Authentication and Security Layer (SASL) mechanisms.

Cisco Unified CM IM and Presence implements a specific SASL mechanism called Cisco-VTG-Token, which involves passing a one time password in the SASL authentication.

Clients can use the Client Configuration Web Service to get this one time password for use in SASL authentication.

For further details on XMPP client authentication, please refer to section 3rd Party XMPP Client Authentication Sequence of this document

Get One Time Password Request

Example 14 shows a sample get-onetime-password request.

Example 14 Get-Onetime-Password request

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:epas="urn:cisco:epas:soap"
            xmlns="urn:cisco:epas:soap">
  <soapenv:Header>
    <session-key> b189aa60-24e3-4e66-a3b6-d8488235ba47</session-key>
  </soapenv:Header>
  <soapenv:Body>
    <get-onetime-password/>
  </soapenv:Body>
</soapenv:Envelope>
```

Example 15 shows a sample get-onetime-password response.

Example 15 Get-Onetime-Password response

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
    <epas:get-onetime-password-resp xmlns:epas="urn:cisco:epas:soap">
      <epas:success>
        <epas:password>5191906</epas:password>
      </epas:success>
    </epas:get-onetime-password-resp>
  </env:Body>
</env:Envelope>
```

Get Configuration Request

After successfully logging in, the client sends a get-all-config request to retrieve all configuration information from Cisco Unified CM IM and Presence. This request will retrieve system configuration, user configuration, licensing features*, contact list, presence rules, application dial rules* and federated domains.

> * Note: This API has been impacted in SOAP interface version 9.0. For SOAP interface version 9.0, you can no longer specify licensing features and application dial rules within the SOAP body.

An optional "include-contact-list" tag is used to request auxiliary information for the contacts in the contact list.

Example below shows a sample get-all-config request for **SOAP interface version 7.0 and 8.0**.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:epas="urn:cisco:epas:soap"
            xmlns="urn:cisco:epas:soap">
  <soapenv:Header>
    <session-key>b189aa60-24e3-4e66-a3b6-d8488235ba47</session-key>
  </soapenv:Header>
  <soapenv:Body>
    <get-all-config>
      <system-config/>
      <user-config/>
      <licensing-features/>
      <dialing-rules/>
      <federated-domains auth-policy="all"/>
      <non-presence-aware-contacts/>
      <contact-info/>
    </get-all-config>
  </soapenv:Body>
</soapenv:Envelope>
```

Example below shows a sample get-all-config response for **SOAP interface version 7.0 and 8.0**.

```
<?xml version="1.0"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <env:Body>
                <epas:get-all-config-resp xmlns:epas="urn:cisco:epas:soap">
                        <epas:get-system-config-resp xmlns:epas="urn:cisco:epas:soap">
                                <epas:property
name="Directory.Map.UserID">sAMAccountName</epas:property>
                                <epas:property name="Directory.Map.FirstName">givenName
</epas:property>
                                <epas:property
name="Directory.Map.LastName">sn</epas:property>
                                <epas:property
name="Directory.Map.MiddleName">middleName</epas:property>
                                <epas:property
name="Directory.Map.Nickname">nickname</epas:property>
                                <epas:property name="Directory.Map.Photo"/>
                                <epas:property
name="Directory.Map.Title">title</epas:property>
                                <epas:property
name="Directory.Map.DisplayName">displayName</epas:property>
                                <epas:property
name="Directory.Map.NamePrefix">namePrefix</epas:property>
                                <epas:property name="Directory.Map.NameSuffix"/>
                                <epas:property
name="Directory.Map.Gender">gender</epas:property>
                                <epas:property
name="Directory.Map.BusinessEMail">mail</epas:property>
                                <epas:property
name="Directory.Map.BusinessPhoneNumber">telephoneNumber</epas:property>
                                <epas:property name="Directory.Map.BusinessVoiceMail"/>
                                <epas:property
name="Directory.Map.BusinessMobilePhone">mobile</epas:property>
                                <epas:property
name="Directory.Map.BusinessPager">pager</epas:property>
                                <epas:property
name="Directory.Map.BusinessFax">facsimileTelephoneNumber</epas:property>
                                <epas:property
name="Directory.Map.BusinessOtherPhone">otherTelephone</epas:property>
                                <epas:property name="Directory.Map.HomeEMail"/>
                                <epas:property name="Directory.Map.HomeMobilePhone"/>
                                <epas:property name="Directory.Map.HomeFax"/>
                                <epas:property
name="Directory.Map.URL">url</epas:property>
                                <epas:property
name="Directory.Map.Organization">Company</epas:property>
                                <epas:property
name="Directory.Map.PrimaryPhoneNumber">telephoneNumber</epas:property>
```

```
                                    <epas:property
name="Directory.Map.AddressStreet">streetAddress</epas:property>
                                    <epas:property
name="Directory.Map.AddressLocation">l</epas:property>
                                    <epas:property
name="Directory.Map.AddressState">st</epas:property>
                                    <epas:property
name="Directory.Map.AddressPostalCode">postalCode</epas:property>
                                    <epas:property
name="Directory.Map.AddressCountry">co</epas:property>
                                    <epas:property name="Security">0</epas:property>
                                    <epas:property name="VoiceMail.Primary.Address"/>
                                    <epas:property name="VoiceMail.Primary.Port"/>
                                    <epas:property name="VoiceMail.Primary.Protocol"/>
                                    <epas:property name="VoiceMail.Backup1.Address"/>
                                    <epas:property name="VoiceMail.Backup1.Port"/>
                                    <epas:property name="VoiceMail.Backup1.Protocol"/>
                                    <epas:property name="VoiceMail.Backup2.Address"/>
                                    <epas:property name="VoiceMail.Backup2.Port"/>
                                    <epas:property name="VoiceMail.Backup2.Protocol"/>
                                    <epas:property name="VoiceMail.PilotNumber"/>
                                    <epas:property name="MeetingPlace.Primary.Address"/>
                                    <epas:property name="MeetingPlace.Primary.Port"/>
                                    <epas:property name="MeetingPlace.Primary.Protocol"/>
                                    <epas:property name="MeetingPlace.Primary.ServerType"/>
                                    <epas:property name="MeetingPlace.Primary.SiteID"/>
                                    <epas:property name="MeetingPlace.Primary.PartnerID"/>
                                    <epas:property name="MeetingPlace.Backup1.Address"/>
                                    <epas:property name="MeetingPlace.Backup1.Port"/>
                                    <epas:property name="MeetingPlace.Backup1.Protocol"/>
                                    <epas:property name="MeetingPlace.Backup1.ServerType"/>
                                    <epas:property name="MeetingPlace.Backup2.Address"/>
                                    <epas:property name="MeetingPlace.Backup2.Port"/>
                                    <epas:property name="MeetingPlace.Backup2.Protocol"/>
                                    <epas:property name="MeetingPlace.Backup2.ServerType"/>
                                    <epas:property name="MeetingPlace.CertLevel"/>
                                    <epas:property name="CallControl.Primary.Address"/>
                                    <epas:property name="CallControl.Primary.Port"/>
                                    <epas:property name="CallControl.Primary.Protocol"/>
                                    <epas:property name="CallControl.Backup1.Address"/>
                                    <epas:property name="CallControl.Backup1.Port"/>
                                    <epas:property name="CallControl.Backup1.Protocol"/>
                                    <epas:property name="CallControl.Backup2.Address"/>
                                    <epas:property name="CallControl.Backup2.Port"/>
                                    <epas:property name="CallControl.Backup2.Protocol"/>
                                    <epas:property name="Directory.DN"></epas:property>
                                    <epas:property name="Directory.Password"></epas:property>
                                    <epas:property
name="Directory.AnonymousBind">FALSE</epas:property>
                                    <epas:property name="Directory.ConfigurationName"/>
                                    <epas:property
name="Directory.SearchContext1"></epas:property>
                                    <epas:property
name="Directory.SearchRecursive1">TRUE</epas:property>
```

```
                                    <epas:property
name="Directory.SearchContext2"></epas:property>
                                    <epas:property
name="Directory.SearchRecursive2">TRUE</epas:property>
                                    <epas:property
name="Directory.SearchContext3"></epas:property>
                                    <epas:property
name="Directory.SearchRecursive3">TRUE</epas:property>
                                    <epas:property
name="Directory.Primary.Address"></epas:property>
                                    <epas:property
name="Directory.Primary.Port"></epas:property>
                                    <epas:property
name="Directory.Primary.Protocol"></epas:property>
                                    <epas:property name="Directory.Backup1.Address"/>
                                    <epas:property name="Directory.Backup1.Port"/>
                                    <epas:property name="Directory.Backup1.Protocol"/>
                                    <epas:property name="Directory.Backup2.Address"/>
                                    <epas:property name="Directory.Backup2.Port"/>
                                    <epas:property name="Directory.Backup2.Protocol"/>
                                    <epas:property
name="Presence.Primary.Address"></epas:property>
                                    <epas:property
name="Presence.Primary.Port"></epas:property>
                                    <epas:property
name="Presence.Primary.Protocol"></epas:property>
                                    <epas:property
name="SOAP.Primary.Address"></epas:property>
                                    <epas:property name="Presence.Backup.Address"/>
                                    <epas:property name="Presence.Backup.Port"/>
                                    <epas:property name="Presence.Backup.Protocol"/>
                                    <epas:property name="SOAP.Backup.Address"/>
                                    <epas:property name="Presence.Domain"></epas:property>
                                    <epas:property name="TFTP.Primary"></epas:property>
                                    <epas:property name="TFTP.Backup1"/>
                                    <epas:property name="TFTP.Backup2"/>
                                    <epas:property
name="CallRecord.MaxAge">UNLIMITED</epas:property>
                                    <epas:property name="IM.enable">TRUE</epas:property>
                                    <epas:property
name="Presence.enableGlobal">TRUE</epas:property>
                                    <epas:property
name="OfflineIM.suppress">FALSE</epas:property>
                                    <epas:property
name="PhoneDND.enable">FALSE</epas:property>
                                    <epas:property
name="MeetingDND.enable">FALSE</epas:property>
                                    <epas:property name="Calendar.Primary"></epas:property>
                                    <epas:property name="Calendar.Backup"/>
                                    <epas:property
name="CUP.ProxyUDPListener.Port"></epas:property>
                                    <epas:property
name="CUP.ProxyTCPListener.Port"></epas:property>
```

```
                                    <epas:property
name="CUP.ProxyTLSListenerPeerAuth.Port"></epas:property>
                                    <epas:property
name="CUP.ProxyTLSListenerServerAuth.Port"></epas:property>
                                    <epas:property name="CUP.DecomposedLists"/>
                                    <epas:property name="CCMCIP.Host"/>
                                    <epas:property name="CCMCIP.Host.Backup"/>
                                    <epas:property name="CCMCIP.Host.CertLevel"/>
                                    <epas:property
name="Audio.UseAGC">FALSE</epas:property>
                                    <epas:property name="Audio.AGCType"/>
                                    <epas:property name="Audio.UseNS">FALSE</epas:property>
                                    <epas:property name="Audio.NSMode"/>
                                    <epas:property name="Audio.UseVAD">FALSE</epas:property>
                                    <epas:property name="Audio.VADMode"/>
                                    <epas:property name="Audio.UseEC">FALSE</epas:property>
                                    <epas:property name="Audio.ECMode"/>
                                    <epas:property name="Audio.ECType"/>
                                    <epas:property
name="IM.AllowCutAndPaste">TRUE</epas:property>
                                    <epas:property
name="IM.AllowLocalTranscript">TRUE</epas:property>
                                    <epas:property
name="AdhocSubscriptions.Enabled">TRUE</epas:property>
                                    <epas:property
name="AdhocSubscriptions.MaxNum">100</epas:property>
                                    <epas:property
name="AdhocSubscriptions.TTL">90</epas:property>
                                    <epas:property name="Security.CertificateDirectory"/>
                                    <epas:property
name="Security.VoicemailServiceCredentialsSource"></epas:property>
                                    <epas:property
name="Security.WebConfServiceCredentialsSource"></epas:property>
                                    <epas:property
name="Video.ExplicitCameraEnabled">FALSE</epas:property>
                                    <epas:property
name="CUP.SRM.LowerFailoverLoginRetryLimit"></epas:property>
                                    <epas:property
name="CUP.SRM.UpperFailoverLoginRetryLimit"></epas:property>
                                    <epas:property name="DecomposedList">
                                        <clusters>
                                            <cluster>
                                                <SubCluster id="" PrimaryServer=""
SecondaryServer=""/>
                                            </cluster>
                                        </clusters>
                                    </epas:property>
                                </epas:get-system-config-resp>
                                <epas:get-user-config-resp xmlns:epas="urn:cisco:epas:soap">
                                    <epas:property name="Directory.MaxResults" is-
public="false">1000</epas:property>
                                    <epas:property name="Directory.MaxTime" is-
public="false">30</epas:property>
```

```
                                    <epas:property name="Presence.inPersistentState" is-
public="false">false</epas:property>
                                    <epas:property name="Presence.displayName" is-
public="false">Jose Sebastian</epas:property>
                                    <epas:property name="Presence.persistAwayWhenOfflineFlag"
is-public="false">false</epas:property>
                                    <epas:property
name="CUCSF.Custom.cupPreferredDefaultGroup" is-public="true">General</epas:property>
                                    <epas:property name="CUCSF.Custom.enableDNDNoNotify" is-
public="true">true</epas:property>
                                    <epas:property name="Presence.userName" is-
public="false">jose</epas:property>
                                    <epas:property name="Presence.listName" is-
public="false">jose-contacts</epas:property>
                                    <epas:property name="Presence.calendar" is-
public="false">disabled</epas:property>
                            </epas:get-user-config-resp>
                            <epas:get-licensing-features-resp xmlns:epas="urn:cisco:epas:soap">
                                    <epas:base>false</epas:base>
                                    <epas:im>false</epas:im>
                                    <epas:audio>false</epas:audio>
                                    <epas:video>false</epas:video>
                            </epas:get-licensing-features-resp>
                            <epas:get-dialing-rules-resp
xmlns:epas="urn:cisco:epas:soap"></epas:get-dialing-rules-resp>
                            <get-federated-domains-resp>
                                    <domain name="cisco.com" auth-policy="allowed">
                                            <description>Cisco Systems</description>
                                    </domain>
                                    <domain name="DomainA.net" auth-policy="blocked">
                                            <description>DomainA Users</description>
                                    </domain>
                                    <domain name="corporateX.com" auth-policy="allowed">
                                            <description>CorporateX Network</description>
                                    </domain>
                            </get-federated-domains-resp>
                            <epas:get-non-presence-aware-contacts-resp>
                                    <epas:group name="General123">
                                            <epas:persona-id nickname="Test Again">6c840874-
3a65-48d4-ace4-67e311e6c039</epas:persona-id>
                                            <epas:persona-id nickname="test4 PG 4 -
changed">afcb5f28-c4ff-4532-a6cb-abf3ac9df8cc</epas:persona-id>
                                    </epas:group>
                                    <epas:group name="Group44">
                                            <epas:persona-id nickname="John B Doe">37dbe62e-
c030-48d6-b2f8-28650495a428</epas:persona-id>
                                            <epas:persona-id nickname="Someone
Else">c0644330-ba71-4b2e-913e-67eea7e7d9a0</epas:persona-id>
                                    </epas:group>
                                    <epas:group name="another - changed">
                                            <epas:persona-id nickname="Someone
Else">c0644330-ba71-4b2e-913e-67eea7e7d9a0</epas:persona-id>
                                            <epas:persona-id nickname="PG 5 ">bbc70767-
1457-4192-9e5b-746d3ba8cafb</epas:persona-id>
```

```
                                    </epas:group>
                                </epas:get-non-presence-aware-contacts-resp>
                                <epas:get-contact-info-resp xmlns="urn:cisco:epas:soap">
                                    <epas:contact-info persona-id="37dbe62e-c030-48d6-b2f8-
28650495a428" domain="" contentType="text/directory" version="3.0">BEGIN:VCARD
                                    FN:John Doe
                                    N:Doe;John;;;
                                    TEL;TYPE=X-CUSTOM1,VOICE;X-LABEL=Fax:1234567
                                    VERSION:3.0
                                    END:VCARD</epas:contact-info>
                                    <epas:contact-info persona-id="bbc70767-1457-4192-9e5b-
746d3ba8cafb" domain="" contentType="text/directory" version="3.0">BEGIN:VCARD
                                    FN:PG 5
                                    VERSION:3.0
                                    END:VCARD</epas:contact-info>
                                    <epas:contact-info persona-id="c0644330-ba71-4b2e-913e-
67eea7e7d9a0" domain="" contentType="text/directory" version="3.0">BEGIN:VCARD
                                    FN:Someone Else
                                    N:Else;Someone;;;
                                    TEL;TYPE=X-CUSTOM1,VOICE;X-LABEL=Fax:123456789
                                    VERSION:3.0
                                    END:VCARD</epas:contact-info>
                                    <epas:contact-info persona-id="6c840874-3a65-48d4-ace4-
67e311e6c039" domain="" contentType="text/directory" version="3.0">BEGIN:VCARD
                                    FN:Test Again
                                    N:Again;Test;;;
                                    VERSION:3.0
                                    END:VCARD</epas:contact-info>
                                    <epas:contact-info persona-id="afcb5f28-c4ff-4532-a6cb-
abf3ac9df8cc" domain="" contentType="text/directory" version="3.0">BEGIN:VCARD
                                    FN:test4 PG 4
                                    N:PG 4;test4;;;
                                    VERSION:3.0
                                    END:VCARD</epas:contact-info>
                                </epas:get-contact-info-resp>
                            </epas:get-all-config-resp>
                        </env:Body>
</env:Envelope>
```

Example below shows a sample get-all-config request for **SOAP interface version 9.0**.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                  xmlns:epas="urn:cisco:epas:soap"
                  xmlns="urn:cisco:epas:soap">
  <soapenv:Header>
    <session-key>b189aa60-24e3-4e66-a3b6-d8488235ba47</session-key>
  </soapenv:Header>
  <soapenv:Body>
    <get-all-config>
      <system-config/>
```

```
      <user-config/>
      <federated-domains auth-policy="all"/>
      <non-presence-aware-contacts/>
      <contact-info/>
    </get-all-config>
  </soapenv:Body>
</soapenv:Envelope>
```

Example below shows a sample get-all-config response for **SOAP interface version 9.0**.

```
<?xml version="1.0"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<env:Body>
   <epas:get-all-config-resp xmlns:epas="urn:cisco:epas:soap">
      <epas:get-system-config-resp xmlns:epas="urn:cisco:epas:soap">
         <epas:property name="Directory.Map.UserID">sAMAccountName</epas:property>
         <epas:property name="Directory.Map.FirstName">givenName</epas:property>
         <epas:property name="Directory.Map.LastName">sn</epas:property>
         <epas:property name="Directory.Map.MiddleName">middleName</epas:property>
         <epas:property name="Directory.Map.Nickname">nickname</epas:property>
         <epas:property name="Directory.Map.Photo"/>
         <epas:property name="Directory.Map.Title">title</epas:property>
         <epas:property name="Directory.Map.DisplayName">displayName</epas:property>
         <epas:property name="Directory.Map.NamePrefix">namePrefix</epas:property>
         <epas:property name="Directory.Map.NameSuffix"/>
         <epas:property name="Directory.Map.Gender">gender</epas:property>
         <epas:property name="Directory.Map.BusinessEMail">mail</epas:property>
         <epas:property
name="Directory.Map.BusinessPhoneNumber">telephoneNumber</epas:property>
         <epas:property name="Directory.Map.BusinessVoiceMail"/>
         <epas:property name="Directory.Map.BusinessMobilePhone">mobile</epas:property>
         <epas:property name="Directory.Map.BusinessPager">pager</epas:property>
         <epas:property
name="Directory.Map.BusinessFax">facsimileTelephoneNumber</epas:property>
         <epas:property
name="Directory.Map.BusinessOtherPhone">otherTelephone</epas:property>
         <epas:property name="Directory.Map.HomeEMail"/>
         <epas:property name="Directory.Map.HomeMobilePhone"/>
         <epas:property name="Directory.Map.HomeFax"/>
         <epas:property name="Directory.Map.URL">url</epas:property>
         <epas:property name="Directory.Map.Organization">Company</epas:property>
         <epas:property
name="Directory.Map.PrimaryPhoneNumber">telephoneNumber</epas:property>
         <epas:property name="Directory.Map.AddressStreet">streetAddress</epas:property>
         <epas:property name="Directory.Map.AddressLocation">l</epas:property>
         <epas:property name="Directory.Map.AddressState">st</epas:property>
         <epas:property name="Directory.Map.AddressPostalCode">postalCode</epas:property>
         <epas:property name="Directory.Map.AddressCountry">co</epas:property>
         <epas:property name="Presence.Primary.Address">gwydlvm226</epas:property>
         <epas:property name="Presence.Primary.Port">5060</epas:property>
```

```
        <epas:property name="Presence.Primary.Protocol">TCP</epas:property>
        <epas:property name="SOAP.Primary.Address">gwydlvm226</epas:property>
        <epas:property name="Presence.Backup.Address"/>
        <epas:property name="Presence.Backup.Port"/>
        <epas:property name="Presence.Backup.Protocol"/>
        <epas:property name="SOAP.Backup.Address"/>
        <epas:property name="Presence.Domain">cisco.com</epas:property>
        <epas:property name="IM.enable">TRUE</epas:property>
        <epas:property name="AdhocSubscriptions.Enabled">TRUE</epas:property>
        <epas:property name="AdhocSubscriptions.MaxNum">100</epas:property>
        <epas:property name="AdhocSubscriptions.TTL">90</epas:property>
        <epas:property name="Presence.enableGlobal">TRUE</epas:property>
        <epas:property name="OfflineIM.suppress">FALSE</epas:property>
        <epas:property name="Calendar.Primary"/>
        <epas:property name="Calendar.Backup"/>
        <epas:property name="Video.ExplicitCameraEnabled">FALSE</epas:property>
        <epas:property name="CUP.SRM.LowerFailoverLoginRetryLimit">40</epas:property>
        <epas:property name="CUP.SRM.UpperFailoverLoginRetryLimit">207</epas:property>
    </epas:get-system-config-resp>
    <epas:get-user-config-resp xmlns:epas="urn:cisco:epas:soap">
        <epas:property name="Directory.MaxResults" is-public="false">1000</epas:property>
        <epas:property name="Directory.MaxTime" is-public="false">30</epas:property>
        <epas:property name="Presence.inPersistentState" is-public="false">false</epas:property>
        <epas:property name="Presence.displayName" is-public="false">Jose
Sebastian</epas:property>
        <epas:property name="Presence.persistAwayWhenOfflineFlag" is-
public="false">false</epas:property>
        <epas:property name="CUCSF.Custom.cupPreferredDefaultGroup" is-
public="true">General</epas:property>
        <epas:property name="CUCSF.Custom.enableDNDNoNotify" is-
public="true">true</epas:property>
        <epas:property name="Presence.userName" is-public="false">jose</epas:property>
        <epas:property name="Presence.listName" is-public="false">jose-contacts</epas:property>
        <epas:property name="Presence.calendar" is-public="false">disabled</epas:property>
    </epas:get-user-config-resp>
    <get-federated-domains-resp>
        <domain name="cisco.com" auth-policy="allowed">
            <description>Cisco Systems</description>
        </domain>
        <domain name="DomainA.net" auth-policy="blocked">
            <description>DomainA Users</description>
        </domain>
        <domain name="corporateX.com" auth-policy="allowed">
            <description>CorporateX Network</description>
        </domain>
    </get-federated-domains-resp>
    <epas:get-non-presence-aware-contacts-resp>
        <epas:group name="General123">
            <epas:persona-id nickname="Test Again">6c840874-3a65-48d4-ace4-
67e311e6c039</epas:persona-id>
            <epas:persona-id nickname="test4 PG 4 - changed">afcb5f28-c4ff-4532-a6cb-
abf3ac9df8cc</epas:persona-id>
        </epas:group>
        <epas:group name="Group44">
```

```
            <epas:persona-id nickname="John B Doe">37dbe62e-c030-48d6-b2f8-
28650495a428</epas:persona-id>
            <epas:persona-id nickname="Someone Else">c0644330-ba71-4b2e-913e-
67eea7e7d9a0</epas:persona-id>
         </epas:group>
         <epas:group name="another - changed">
            <epas:persona-id nickname="Someone Else">c0644330-ba71-4b2e-913e-
67eea7e7d9a0</epas:persona-id>
            <epas:persona-id nickname="PG 5 ">bbc70767-1457-4192-9e5b-
746d3ba8cafb</epas:persona-id>
         </epas:group>
      </epas:get-non-presence-aware-contacts-resp>
      <epas:get-contact-info-resp xmlns="urn:cisco:epas:soap">
         <epas:contact-info persona-id="37dbe62e-c030-48d6-b2f8-28650495a428" domain=""
contentType="text/directory" version="3.0">BEGIN:VCARD
         FN:John Doe
         N:Doe;John;;;
         TEL;TYPE=X-CUSTOM1,VOICE;X-LABEL=Fax:1234567
         VERSION:3.0
         END:VCARD</epas:contact-info>
         <epas:contact-info persona-id="bbc70767-1457-4192-9e5b-746d3ba8cafb" domain=""
contentType="text/directory" version="3.0">BEGIN:VCARD
         FN:PG 5
         VERSION:3.0
         END:VCARD</epas:contact-info>
         <epas:contact-info persona-id="c0644330-ba71-4b2e-913e-67eea7e7d9a0" domain=""
contentType="text/directory" version="3.0">BEGIN:VCARD
         FN:Someone Else
         N:Else;Someone;;;
         TEL;TYPE=X-CUSTOM1,VOICE;X-LABEL=Fax:123456789
         VERSION:3.0
         END:VCARD</epas:contact-info>
         <epas:contact-info persona-id="6c840874-3a65-48d4-ace4-67e311e6c039" domain=""
contentType="text/directory" version="3.0">BEGIN:VCARD
         FN:Test Again
         N:Again;Test;;;
         VERSION:3.0
         END:VCARD</epas:contact-info>
         <epas:contact-info persona-id="afcb5f28-c4ff-4532-a6cb-abf3ac9df8cc" domain=""
contentType="text/directory" version="3.0">BEGIN:VCARD
         FN:test4 PG 4
         N:PG 4;test4;;;
         VERSION:3.0
         END:VCARD</epas:contact-info>
      </epas:get-contact-info-resp>
   </epas:get-all-config-resp>
</env:Body>
</env:Envelope>
```

Example below shows a sample get-all-config request for SOAP interface version 10.5.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:urn="urn:cisco:epas:soap">
   <soap:Header>
```

```
    <urn:session-key>${#Project#SessionKey}</urn:session-key>
  </soap:Header>
  <soap:Body>
    <urn:get-all-config>
      <urn:system-config/>
      <urn:user-config/>
      <urn:licensing-features/>
      <urn:dialing-rules/>
      <urn:non-presence-aware-contacts/>
      <urn:contact-info/>
    </urn:get-all-config>
  </soap:Body>
</soap:Envelope>
```

Example below shows a sample get-all-config response for SOAP interface version 10.5.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
    <epas:get-all-config-resp xmlns:epas="urn:cisco:epas:soap">
      <epas:get-system-config-resp>
        <epas:property name="Directory.Map.UserID">sAMAccountName</epas:property>
        <epas:property name="Directory.Map.FirstName">givenName</epas:property>
        <epas:property name="Directory.Map.LastName">sn</epas:property>
        <epas:property name="Directory.Map.MiddleName">middleName</epas:property>
        <epas:property name="Directory.Map.Nickname">nickname</epas:property>
        <epas:property name="Directory.Map.Photo"/>
        <epas:property name="Directory.Map.Title">title</epas:property>
        <epas:property name="Directory.Map.DisplayName">displayName</epas:property>
        <epas:property name="Directory.Map.NamePrefix">namePrefix</epas:property>
        <epas:property name="Directory.Map.NameSuffix"/>
        <epas:property name="Directory.Map.Gender">gender</epas:property>
        <epas:property name="Directory.Map.BusinessEMail">mail</epas:property>
        <epas:property
name="Directory.Map.BusinessPhoneNumber">telephoneNumber</epas:property>
        <epas:property name="Directory.Map.BusinessVoiceMail"/>
        <epas:property name="Directory.Map.BusinessMobilePhone">mobile</epas:property>
        <epas:property name="Directory.Map.BusinessPager">pager</epas:property>
        <epas:property
name="Directory.Map.BusinessFax">facsimileTelephoneNumber</epas:property>
        <epas:property
name="Directory.Map.BusinessOtherPhone">otherTelephone</epas:property>
        <epas:property name="Directory.Map.HomeEMail"/>
        <epas:property name="Directory.Map.HomeMobilePhone"/>
        <epas:property name="Directory.Map.HomeFax"/>
        <epas:property name="Directory.Map.URL">url</epas:property>
        <epas:property name="Directory.Map.Organization">Company</epas:property>
        <epas:property
name="Directory.Map.PrimaryPhoneNumber">telephoneNumber</epas:property>
        <epas:property name="Directory.Map.AddressStreet">streetAddress</epas:property>
        <epas:property name="Directory.Map.AddressLocation">l</epas:property>
        <epas:property name="Directory.Map.AddressState">st</epas:property>
        <epas:property name="Directory.Map.AddressPostalCode">postalCode</epas:property>
        <epas:property name="Directory.Map.AddressCountry">co</epas:property>
```

```
<epas:property name="VoiceMail.Primary.Address"/>
<epas:property name="VoiceMail.Primary.SecureMessaging.Port"/>
<epas:property name="VoiceMail.Primary.SecureMessaging.Protocol"/>
<epas:property name="VoiceMail.Backup1.Address"/>
<epas:property name="VoiceMail.Backup1.SecureMessaging.Port"/>
<epas:property name="VoiceMail.Backup1.SecureMessaging.Protocol"/>
<epas:property name="VoiceMail.Backup2.Address"/>
<epas:property name="VoiceMail.Backup2.SecureMessaging.Port"/>
<epas:property name="VoiceMail.Backup2.SecureMessaging.Protocol"/>
<epas:property name="VoiceMail.PilotNumber"/>
<epas:property name="Mailstore.Primary.Address"/>
<epas:property name="Mailstore.Primary.Port"/>
<epas:property name="Mailstore.Primary.Protocol"/>
<epas:property name="Mailstore.Backup1.Address"/>
<epas:property name="Mailstore.Backup1.Port"/>
<epas:property name="Mailstore.Backup1.Protocol"/>
<epas:property name="Mailstore.Backup2.Address"/>
<epas:property name="Mailstore.Backup2.Port"/>
<epas:property name="Mailstore.Backup2.Protocol"/>
<epas:property name="Mailstore.PollingInterval">60</epas:property>
<epas:property name="Mailstore.Inbox.FolderName">INBOX</epas:property>
<epas:property name="Mailstore.Trash.FolderName">Deleted Items</epas:property>
<epas:property name="Mailstore.DualFolder">true</epas:property>
<epas:property name="CallControl.Primary.Address">10.77.46.90</epas:property>
<epas:property name="CallControl.Primary.Port">2748</epas:property>
<epas:property name="CallControl.Primary.Protocol">TCP</epas:property>
<epas:property name="CallControl.Backup1.Address"/>
<epas:property name="CallControl.Backup1.Port"/>
<epas:property name="CallControl.Backup1.Protocol"/>
<epas:property name="CallControl.Backup2.Address"/>
<epas:property name="CallControl.Backup2.Port"/>
<epas:property name="CallControl.Backup2.Protocol"/>
<epas:property name="Directory.Primary.Address"/>
<epas:property name="Directory.Primary.Port"/>
<epas:property name="Directory.Primary.Protocol"/>
<epas:property name="Directory.Backup1.Address"/>
<epas:property name="Directory.Backup1.Port"/>
<epas:property name="Directory.Backup1.Protocol"/>
<epas:property name="Directory.Backup2.Address"/>
<epas:property name="Directory.Backup2.Port"/>
<epas:property name="Directory.Backup2.Protocol"/>
<epas:property name="Directory.AnonymousBind">false</epas:property>
<epas:property name="Directory.Password"/>
<epas:property name="Directory.SearchRecursive3"/>
<epas:property name="Directory.SearchRecursive2"/>
<epas:property name="Directory.SearchContext1"/>
<epas:property name="Directory.SearchContext2"/>
<epas:property name="Directory.SearchContext3"/>
<epas:property name="Directory.DN"/>
<epas:property name="Directory.ConfigurationName"/>
<epas:property name="Directory.SearchRecursive1">true</epas:property>
<epas:property name="MeetingPlace.Primary.Address"/>
<epas:property name="MeetingPlace.Primary.Port"/>
<epas:property name="MeetingPlace.Primary.Protocol"/>
```

```
        <epas:property name="MeetingPlace.Primary.ServerType"/>
        <epas:property name="MeetingPlace.Backup1.Address"/>
        <epas:property name="MeetingPlace.Backup1.Port"/>
        <epas:property name="MeetingPlace.Backup1.Protocol"/>
        <epas:property name="MeetingPlace.Backup1.ServerType"/>
        <epas:property name="MeetingPlace.Backup2.Address"/>
        <epas:property name="MeetingPlace.Backup2.Port"/>
        <epas:property name="MeetingPlace.Backup2.Protocol"/>
        <epas:property name="MeetingPlace.Backup2.ServerType"/>
        <epas:property name="MeetingPlace.CertLevel">ANY_CERT</epas:property>
        <epas:property name="Presence.Primary.Address">cup-dod96</epas:property>
        <epas:property name="Presence.Primary.Port">5060</epas:property>
        <epas:property name="Presence.Primary.Protocol">TCP</epas:property>
        <epas:property
name="Presence.Primary.Ipv6name">2001:1000:1000:1000:250:56ff:febd:4e78</epas:property>
        <epas:property name="SOAP.Primary.Address">cup-dod96</epas:property>
        <epas:property name="Presence.Backup.Address">cup-dod68</epas:property>
        <epas:property name="Presence.Backup.Port">5060</epas:property>
        <epas:property name="Presence.Backup.Protocol">TCP</epas:property>
        <epas:property
name="Presence.Backup.Ipv6name">2001:1000:1000:1000:20c:29ff:fed8:dc35</epas:property>
        <epas:property name="SOAP.Backup.Address">cup-dod68</epas:property>
        <epas:property name="Presence.Domain">vtgtest.com</epas:property>
        <epas:property name="TFTP.Primary">10.77.46.90</epas:property>
        <epas:property name="TFTP.Backup1"/>
        <epas:property name="TFTP.Backup2"/>
        <epas:property name="Security">0</epas:property>
        <epas:property name="CallRecord.MaxAge">UNLIMITED</epas:property>
        <epas:property name="CUP.DecomposedLists">FALSE</epas:property>
        <epas:property name="IM.enable">TRUE</epas:property>
        <epas:property name="Presence.enableGlobal">TRUE</epas:property>
        <epas:property name="OfflineIM.suppress">FALSE</epas:property>
        <epas:property name="PhoneDND.enable">FALSE</epas:property>
        <epas:property name="MeetingDND.enable">FALSE</epas:property>
        <epas:property name="Calendar.Primary"/>
        <epas:property name="Calendar.Backup"/>
        <epas:property name="CUP.ProxyUDPListener.Port">5060</epas:property>
        <epas:property name="CUP.ProxyTCPListener.Port">5060</epas:property>
        <epas:property name="CUP.ProxyTLSListenerPeerAuth.Port">5062</epas:property>
        <epas:property name="CUP.ProxyTLSListenerServerAuth.Port">5061</epas:property>
        <epas:property name="CCMCIP.Host"/>
        <epas:property name="CCMCIP.Host.Backup"/>
        <epas:property name="CCMCIP.Host.CertLevel"/>
        <epas:property name="Audio.UseAGC">TRUE</epas:property>
        <epas:property name="Audio.AGCType">Digital Only</epas:property>
        <epas:property name="Audio.UseNS">TRUE</epas:property>
        <epas:property name="Audio.NSMode">Low</epas:property>
        <epas:property name="Audio.UseVAD">TRUE</epas:property>
        <epas:property name="Audio.VADMode">Least Aggressive</epas:property>
        <epas:property name="Audio.UseEC">TRUE</epas:property>
        <epas:property name="Audio.ECMode">Normal</epas:property>
        <epas:property name="Audio.ECType">Acoustic Echo Cancellation</epas:property>
        <epas:property name="IM.AllowCutAndPaste">TRUE</epas:property>
        <epas:property name="IM.AllowLocalTranscript">TRUE</epas:property>
```

```
        <epas:property name="AdhocSubscriptions.Enabled">TRUE</epas:property>
        <epas:property name="AdhocSubscriptions.MaxNum">50</epas:property>
        <epas:property name="AdhocSubscriptions.TTL">900</epas:property>
        <epas:property name="Security.CertificateDirectory"/>
        <epas:property
name="Security.WebConfServiceCredentialsSource">NOT_SET</epas:property>
        <epas:property
name="Security.VoicemailServiceCredentialsSource">NOT_SET</epas:property>
        <epas:property name="Video.ExplicitCameraEnabled">FALSE</epas:property>
        <epas:property name="CUP.SRM.LowerFailoverLoginRetryLimit">90</epas:property>
        <epas:property name="CUP.SRM.UpperFailoverLoginRetryLimit">257</epas:property>
        <epas:property name="DecomposedList">
          <clusters>
            <cluster id="NONE">
              <SubCluster id="NONE" PrimaryServer="NONE" SecondaryServer="NONE"/>
            </cluster>
          </clusters>
        </epas:property>
      </epas:get-system-config-resp>
      <epas:get-user-config-resp>
        <epas:property name="CallControl.Lineid" is-public="false">6039</epas:property>
        <epas:property name="CallControl.Devicename" is-public="false"/>
        <epas:property name="Credentials.Digest" is-public="false">1234</epas:property>
        <epas:property name="Presence.userName" is-public="false">testuser-
00003</epas:property>
        <epas:property name="Presence.listName" is-public="false">testuser-00003-
contacts</epas:property>
        <epas:property name="Presence.calendar" is-public="false">disabled</epas:property>
      </epas:get-user-config-resp>
      <epas:get-licensing-features-resp>
        <epas:base>true</epas:base>
        <epas:im>true</epas:im>
        <epas:audio>false</epas:audio>
        <epas:video>true</epas:video>
      </epas:get-licensing-features-resp>
      <epas:get-dialing-rules-resp></epas:get-dialing-rules-resp>
      <epas:get-non-presence-aware-contacts-resp></epas:get-non-presence-aware-contacts-
resp>
      <epas:get-contact-info-resp xmlns="urn:cisco:epas:soap"></epas:get-contact-info-resp>
    </epas:get-all-config-resp>
  </env:Body>
</env:Envelope>
```

Get System Configuration Request

The get-system-config request is used to retrieve system level configuration information.

Note: This API has been impacted in SOAP interface version 9.0. For SOAP interface version 9.0, there are several attributes which has been deprecated. Refer the table below, under Remarks, any attribute which has been marked ‘Deprecated’ has been removed from SOAP interface version 9.0 but retained for SOAP interfaces versions 7.0 & 8.0.

NAME OF ATTRIBUTE	Remarks
Directory.Map.UserID	
Directory.Map.FirstName	
Directory.Map.LastName	
Directory.Map.MiddleName	
Directory.Map.Nickname	
Directory.Map.Photo	
Directory.Map.Title	
Directory.Map.DisplayName	
Directory.Map.NamePrefix	
Directory.Map.NameSuffix	
Directory.Map.Gender	
Directory.Map.BusinessEMail	
Directory.Map.BusinessPhoneNumber	
Directory.Map.BusinessVoiceMail	
Directory.Map.BusinessMobilePhone	
Directory.Map.BusinessPager	
Directory.Map.BusinessFax	
Directory.Map.BusinessOtherPhone	
Directory.Map.HomeEMail	
Directory.Map.HomeMobilePhone	
Directory.Map.HomeFax	
Directory.Map.URL	
Directory.Map.Organization	
Directory.Map.PrimaryPhoneNumber	
Directory.Map.AddressStreet	
Directory.Map.AddressLocation	
Directory.Map.AddressState	
Directory.Map.AddressPostalCode	
Directory.Map.AddressCountry	
Security	Deprecated

VoiceMail.Primary.Address	Deprecated
VoiceMail.Backup1.Address	Deprecated
VoiceMail.Backup2.Address	Deprecated
VoiceMail.PilotNumber	Deprecated
VoiceMail.Primary.SecureMessaging.Port	Deprecated
VoiceMail.Primary.SecureMessaging.Protocol	Deprecated
VoiceMail.Backup1.SecureMessaging.Port	Deprecated
VoiceMail.Backup1.SecureMessaging.Protocol	Deprecated
VoiceMail.Backup2.SecureMessaging.Port	Deprecated
VoiceMail.Backup2.SecureMessaging.Protocol	Deprecated
Mailstore.Primary.Address	Deprecated
Mailstore.Primary.Port	Deprecated
Mailstore.Primary.Protocol	Deprecated
Mailstore.Backup1.Address	Deprecated
Mailstore.Backup1.Port	Deprecated
Mailstore.Backup1.Protocol	Deprecated
Mailstore.Backup2.Address	Deprecated
Mailstore.Backup2.Port	Deprecated
Mailstore.Backup2.Protocol	Deprecated
Mailstore.Inbox.FolderName	Deprecated
Mailstore.Trash.FolderName	Deprecated
Mailstore.PollingInterval	Deprecated
Mailstore.DualFolder	Deprecated
MeetingPlace.Primary.Address	Deprecated
MeetingPlace.Primary.Port	Deprecated
MeetingPlace.Primary.Protocol	Deprecated
MeetingPlace.Primary.ServerType	Deprecated
MeetingPlace.Primary.SiteID	Removed from all versions
MeetingPlace.Primary.PartnerID	Removed from all versions
MeetingPlace.Backup1.Address	Deprecated
MeetingPlace.Backup1.Port	Deprecated

MeetingPlace.Backup1.Protocol	Deprecated
MeetingPlace.Backup1.ServerType	Deprecated
MeetingPlace.Backup1.SiteID	Removed from all versions
MeetingPlace.Backup1.PartnerID	Removed from all versions
MeetingPlace.Backup2.Address	Deprecated
MeetingPlace.Backup2.Port	Deprecated
MeetingPlace.Backup2.Protocol	Deprecated
MeetingPlace.Backup2.ServerType	Deprecated
MeetingPlace.Backup2.SiteID	Removed from all versions
MeetingPlace.Backup2.PartnerID	Removed from all versions
MeetingPlace.CertLevel	Deprecated
CallControl.Primary.Address	Deprecated
CallControl.Primary.Port	Deprecated
CallControl.Primary.Protocol	Deprecated
CallControl.Backup1.Address	Deprecated
CallControl.Backup1.Port	Deprecated
CallControl.Backup1.Protocol	Deprecated
CallControl.Backup2.Address	Deprecated
CallControl.Backup2.Port	Deprecated
CallControl.Backup2.Protocol	Deprecated
Directory.DN	Deprecated
Directory.Password	Deprecated
Directory.AnonymousBind	Deprecated
Directory.ConfigurationName	Deprecated
Directory.SearchContext1	Deprecated
Directory.SearchRecursive1	Deprecated
Directory.SearchContext2	Deprecated
Directory.SearchRecursive2	Deprecated
Directory.SearchContext3	Deprecated
Directory.SearchRecursive3	Deprecated
Directory.Primary.Address	Deprecated

Directory.Primary.Port	Deprecated
Directory.Primary.Protocol	Deprecated
Directory.Backup1.Address	Deprecated
Directory.Backup1.Port	Deprecated
Directory.Backup1.Protocol	Deprecated
Directory.Backup2.Address	Deprecated
Directory.Backup2.Port	Deprecated
Directory.Backup2.Protocol	Deprecated
Presence.Primary.Address	
Presence.Primary.Port	
Presence.Primary.Protocol	
SOAP.Primary.Address	
Presence.Backup.Address	
Presence.Backup.Port	
Presence.Backup.Protocol	
SOAP.Backup.Address	
Presence.Domain	
TFTP.Primary	Deprecated
TFTP.Backup1	Deprecated
TFTP.Backup2	Deprecated
CallRecord.MaxAge	Deprecated
IM.enable	
Presence.enableStealth	Removed from all versions
Presence.enableGlobal	
OfflineIM.suppress	
PhoneDND.enable	Deprecated
MeetingDND.enable	Deprecated
Calendar.Primary	
Calendar.Backup	
CUP.ProxyUDPListener.Port	Deprecated
CUP.ProxyTCPListener.Port	Deprecated

CUP.ProxyTLSListenerPeerAuth.Port	Deprecated
CUP.ProxyTLSListenerServerAuth.Port	Deprecated
CUP.DecomposedLists	Deprecated
CCMCIP.Host	Deprecated
CCMCIP.Host.Backup	Deprecated
CCMCIP.Host.CertLevel	Deprecated
Audio.UseAGC	Deprecated
Audio.AGCType	Deprecated
Audio.UseNS	Deprecated
Audio.NSMode	Deprecated
Audio.UseVAD	Deprecated
Audio.VADMode	Deprecated
Audio.UseEC	Deprecated
Audio.ECMode	Deprecated
Audio.ECType	Deprecated
IM.AllowCutAndPaste	Deprecated
IM.AllowLocalTranscript	Deprecated
AdhocSubscriptions.Enabled	
AdhocSubscriptions.MaxNum	
AdhocSubscriptions.TTL	
Security.CertificateDirectory	Deprecated
Security.VoicemailServiceCredentialsSource	Deprecated
Security.WebConfServiceCredentialsSource	Deprecated
Video.ExplicitCameraEnabled	
CUP.SRM.LowerFailoverLoginRetryLimit	
CUP.SRM.UpperFailoverLoginRetryLimit	
DecomposedList	Deprecated

Example below shows a get-system-config request for **SOAP interface version 7.0 & 8.0.**

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

```
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:epas="urn:cisco:epas:soap"
            xmlns="urn:cisco:epas:soap">
  <soapenv:Header>
    <session-key>b189aa60-24e3-4e66-a3b6-d8488235ba47</session-key>
  </soapenv:Header>
  <soapenv:Body>
    <get-system-config/>
  </soapenv:Body>
</soapenv:Envelope>
```

Example below shows a get-system-config response for **SOAP interface version 7.0 & 8.0.**

Get-System-Config response

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
     <epas:get-system-config-resp  xmlns:epas="urn:cisco:epas:soap" >
   <epas:property name="Directory.Map.UserID">sAMAccountName</epas:property>
   <epas:property name="Directory.Map.FirstName">givenName</epas:property>
   <epas:property name="Directory.Map.LastName">sn</epas:property>
   <epas:property name="Directory.Map.MiddleName">middleName</epas:property>
   <epas:property name="Directory.Map.Nickname">nickname</epas:property>
   <epas:property name="Directory.Map.Photo"/>
   <epas:property name="Directory.Map.Title">title</epas:property>
   <epas:property name="Directory.Map.DisplayName">displayName</epas:property>
   <epas:property name="Directory.Map.NamePrefix">namePrefix</epas:property>
   <epas:property name="Directory.Map.NameSuffix"/>
   <epas:property name="Directory.Map.Gender">gender</epas:property>
   <epas:property name="Directory.Map.BusinessEMail">mail</epas:property>
   <epas:property
name="Directory.Map.BusinessPhoneNumber">telephoneNumber</epas:property>
   <epas:property name="Directory.Map.BusinessVoiceMail"/>
   <epas:property name="Directory.Map.BusinessMobilePhone">mobile</epas:property>
   <epas:property name="Directory.Map.BusinessPager">pager    </epas:property>
   <epas:property name="Directory.Map.BusinessFax">facsimileTelephoneNumber</epas:property>
   <epas:property name="Directory.Map.BusinessOtherPhone">otherTelephone</epas:property>
   <epas:property name="Directory.Map.HomeEMail"/>
   <epas:property name="Directory.Map.HomeMobilePhone"/>
   <epas:property name="Directory.Map.HomeFax"/>
   <epas:property name="Directory.Map.URL">url</epas:property>
   <epas:property name="Directory.Map.Organization">Company</epas:property>
   <epas:property name="Directory.Map.PrimaryPhoneNumber">telephoneNumber</epas:property>
   <epas:property name="Directory.Map.AddressStreet">streetAddress</epas:property>
   <epas:property name="Directory.Map.AddressLocation">l</epas:property>
   <epas:property name="Directory.Map.AddressState">st</epas:property>
   <epas:property name="Directory.Map.AddressPostalCode">postalCode</epas:property>
   <epas:property name="Directory.Map.AddressCountry">co</epas:property>
   <epas:property name="Security">0</epas:property>
   <epas:property name="VoiceMail.Primary.Address"/>
   <epas:property name="VoiceMail.Primary.Port"/>
   <epas:property name="VoiceMail.Primary.Protocol"/>
   <epas:property name="VoiceMail.Backup1.Address"/>
```

```
<epas:property name="VoiceMail.Backup1.Port"/>
<epas:property name="VoiceMail.Backup1.Protocol"/>
<epas:property name="VoiceMail.Backup2.Address"/>
<epas:property name="VoiceMail.Backup2.Port"/>
<epas:property name="VoiceMail.Backup2.Protocol"/>
<epas:property name="VoiceMail.PilotNumber"/>
<epas:property name="MeetingPlace.Primary.Address"/>
<epas:property name="MeetingPlace.Primary.Port"/>
<epas:property name="MeetingPlace.Primary.Protocol"/>
<epas:property name="MeetingPlace.Primary.ServerType"/>
<epas:property name="MeetingPlace.Primary.SiteID"/>
<epas:property name="MeetingPlace.Primary.PartnerID"/>
<epas:property name="MeetingPlace.Backup1.Address"/>
<epas:property name="MeetingPlace.Backup1.Port"/>
<epas:property name="MeetingPlace.Backup1.Protocol"/>
<epas:property name="MeetingPlace.Backup1.ServerType"/>
<epas:property name="MeetingPlace.Backup2.Address"/>
<epas:property name="MeetingPlace.Backup2.Port"/>
<epas:property name="MeetingPlace.Backup2.Protocol"/>
<epas:property name="MeetingPlace.Backup2.ServerType"/>
<epas:property name="MeetingPlace.CertLevel"/>
<epas:property name="CallControl.Primary.Address"/>
<epas:property name="CallControl.Primary.Port"/>
<epas:property name="CallControl.Primary.Protocol"/>
<epas:property name="CallControl.Backup1.Address"/>
<epas:property name="CallControl.Backup1.Port"/>
<epas:property name="CallControl.Backup1.Protocol"/>
<epas:property name="CallControl.Backup2.Address"/>
<epas:property name="CallControl.Backup2.Port"/>
<epas:property name="CallControl.Backup2.Protocol"/>
<epas:property name="Directory.DN"></epas:property>
<epas:property name="Directory.Password"></epas:property>
<epas:property name="Directory.AnonymousBind">FALSE</epas:property>
<epas:property name="Directory.ConfigurationName"/>
<epas:property name="Directory.SearchContext1"></epas:property>
<epas:property name="Directory.SearchRecursive1">TRUE</epas:property>
<epas:property name="Directory.SearchContext2"></epas:property>
<epas:property name="Directory.SearchRecursive2">TRUE</epas:property>
<epas:property name="Directory.SearchContext3"></epas:property>
<epas:property name="Directory.SearchRecursive3">TRUE</epas:property>
<epas:property name="Directory.Primary.Address"></epas:property>
<epas:property name="Directory.Primary.Port"></epas:property>
<epas:property name="Directory.Primary.Protocol"></epas:property>
<epas:property name="Directory.Backup1.Address"/>
<epas:property name="Directory.Backup1.Port"/>
<epas:property name="Directory.Backup1.Protocol"/>
<epas:property name="Directory.Backup2.Address"/>
<epas:property name="Directory.Backup2.Port"/>
<epas:property name="Directory.Backup2.Protocol"/>
<epas:property name="Presence.Primary.Address"></epas:property>
<epas:property name="Presence.Primary.Port"></epas:property>
<epas:property name="Presence.Primary.Protocol"></epas:property>
<epas:property name="SOAP.Primary.Address"></epas:property>
<epas:property name="Presence.Backup.Address"/>
```

```
    <epas:property name="Presence.Backup.Port"/>
    <epas:property name="Presence.Backup.Protocol"/>
    <epas:property name="SOAP.Backup.Address"/>
    <epas:property name="Presence.Domain"></epas:property>
    <epas:property name="TFTP.Primary"></epas:property>
    <epas:property name="TFTP.Backup1"/>
    <epas:property name="TFTP.Backup2"/>
    <epas:property name="CallRecord.MaxAge">UNLIMITED</epas:property>
    <epas:property name="IM.enable">TRUE</epas:property>
    <epas:property name="Presence.enableGlobal">TRUE</epas:property>
    <epas:property name="OfflineIM.suppress">FALSE</epas:property>
    <epas:property name="PhoneDND.enable">FALSE</epas:property>
    <epas:property name="MeetingDND.enable">FALSE</epas:property>
    <epas:property name="Calendar.Primary"></epas:property>
    <epas:property name="Calendar.Backup"/>
    <epas:property name="CUP.ProxyUDPListener.Port"></epas:property>
    <epas:property name="CUP.ProxyTCPListener.Port"></epas:property>
    <epas:property name="CUP.ProxyTLSListenerPeerAuth.Port"></epas:property>
    <epas:property name="CUP.ProxyTLSListenerServerAuth.Port"></epas:property>
    <epas:property name="CUP.DecomposedLists"/>
    <epas:property name="CCMCIP.Host"/>
    <epas:property name="CCMCIP.Host.Backup"/>
    <epas:property name="CCMCIP.Host.CertLevel"/>
    <epas:property name="Audio.UseAGC">FALSE</epas:property>
    <epas:property name="Audio.AGCType"/>
    <epas:property name="Audio.UseNS">FALSE</epas:property>
    <epas:property name="Audio.NSMode"/>
    <epas:property name="Audio.UseVAD">FALSE</epas:property>
    <epas:property name="Audio.VADMode"/>
    <epas:property name="Audio.UseEC">FALSE</epas:property>
    <epas:property name="Audio.ECMode"/>
    <epas:property name="Audio.ECType"/>
    <epas:property name="IM.AllowCutAndPaste">TRUE</epas:property>
    <epas:property name="IM.AllowLocalTranscript">TRUE</epas:property>
    <epas:property name="AdhocSubscriptions.Enabled">TRUE</epas:property>
    <epas:property name="AdhocSubscriptions.MaxNum"></epas:property>
    <epas:property name="AdhocSubscriptions.TTL"></epas:property>
    <epas:property name="Security.CertificateDirectory"/>
    <epas:property name="Security.VoicemailServiceCredentialsSource"></epas:property>
    <epas:property name="Security.WebConfServiceCredentialsSource"></epas:property>
    <epas:property name="Video.ExplicitCameraEnabled">FALSE</epas:property>
    <epas:property name="CUP.SRM.LowerFailoverLoginRetryLimit"></epas:property>
    <epas:property name="CUP.SRM.UpperFailoverLoginRetryLimit"></epas:property>
    <epas:property name="DecomposedList">
      <clusters>
        <cluster>
          <SubCluster id="" PrimaryServer="" SecondaryServer=""/>
        </cluster>
      </clusters>
    </epas:property>
  </epas:get-system-config-resp>
 </env:Body>
</env:Envelope>
```

Example below shows a get-system-config response for **SOAP interface version 9.0.**

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <env:Body>
      <epas:get-system-config-resp xmlns:epas="urn:cisco:epas:soap">
         <epas:property name="Directory.Map.UserID">sAMAccountName</epas:property>
         <epas:property name="Directory.Map.FirstName">givenName</epas:property>
         <epas:property name="Directory.Map.LastName">sn</epas:property>
         <epas:property name="Directory.Map.MiddleName">middleName</epas:property>
         <epas:property name="Directory.Map.Nickname">nickname</epas:property>
         <epas:property name="Directory.Map.Photo"/>
         <epas:property name="Directory.Map.Title">title</epas:property>
         <epas:property name="Directory.Map.DisplayName">displayName</epas:property>
         <epas:property name="Directory.Map.NamePrefix">namePrefix</epas:property>
         <epas:property name="Directory.Map.NameSuffix"/>
         <epas:property name="Directory.Map.Gender">gender</epas:property>
         <epas:property name="Directory.Map.BusinessEMail">mail</epas:property>
         <epas:property
name="Directory.Map.BusinessPhoneNumber">telephoneNumber</epas:property>
         <epas:property name="Directory.Map.BusinessVoiceMail"/>
         <epas:property name="Directory.Map.BusinessMobilePhone">mobile</epas:property>
         <epas:property name="Directory.Map.BusinessPager">pager</epas:property>
         <epas:property
name="Directory.Map.BusinessFax">facsimileTelephoneNumber</epas:property>
         <epas:property
name="Directory.Map.BusinessOtherPhone">otherTelephone</epas:property>
         <epas:property name="Directory.Map.HomeEMail"/>
         <epas:property name="Directory.Map.HomeMobilePhone"/>
         <epas:property name="Directory.Map.HomeFax"/>
         <epas:property name="Directory.Map.URL">url</epas:property>
         <epas:property name="Directory.Map.Organization">Company</epas:property>
         <epas:property
name="Directory.Map.PrimaryPhoneNumber">telephoneNumber</epas:property>
         <epas:property name="Directory.Map.AddressStreet">streetAddress</epas:property>
         <epas:property name="Directory.Map.AddressLocation">l</epas:property>
         <epas:property name="Directory.Map.AddressState">st</epas:property>
         <epas:property name="Directory.Map.AddressPostalCode">postalCode</epas:property>
         <epas:property name="Directory.Map.AddressCountry">co</epas:property>
         <epas:property name="Presence.Primary.Address">gwydlvm226</epas:property>
         <epas:property name="Presence.Primary.Port">5060</epas:property>
         <epas:property name="Presence.Primary.Protocol">TCP</epas:property>
         <epas:property name="SOAP.Primary.Address">gwydlvm226</epas:property>
         <epas:property name="Presence.Backup.Address"/>
         <epas:property name="Presence.Backup.Port"/>
         <epas:property name="Presence.Backup.Protocol"/>
         <epas:property name="SOAP.Backup.Address"/>
         <epas:property name="Presence.Domain">cisco.com</epas:property>
         <epas:property name="IM.enable">TRUE</epas:property>
         <epas:property name="AdhocSubscriptions.Enabled">TRUE</epas:property>
         <epas:property name="AdhocSubscriptions.MaxNum">100</epas:property>
         <epas:property name="AdhocSubscriptions.TTL">90</epas:property>
         <epas:property name="Presence.enableGlobal">TRUE</epas:property>
         <epas:property name="OfflineIM.suppress">FALSE</epas:property>
```

```
        <epas:property name="Calendar.Primary"/>
        <epas:property name="Calendar.Backup"/>
        <epas:property name="Video.ExplicitCameraEnabled">FALSE    </epas:property>
        <epas:property name="CUP.SRM.LowerFailoverLoginRetryLimit">40</epas:property>
        <epas:property name="CUP.SRM.UpperFailoverLoginRetryLimit">207</epas:property>
      </epas:get-system-config-resp>
   </env:Body>
</env:Envelope>
```

Get User Configuration Request

The get-user-config request retrieves per-user configuration information.

Note: This API has been impacted in SOAP interface version 9.0. For SOAP interface version 9.0, there are several attributes which has been deprecated. Refer the table below, under Remarks, any attribute which has been marked 'Deprecated' has been removed from SOAP interface version 9.0 but retained for SOAP interfaces versions 7.0 & 8.0.

NAME OF ATTRIBUTE	Remarks
Directory.MaxTime	
Directory.MaxResults	
Presence.inPersistentState	
Presence.displayName	
Reachability.listOfCustomAnnotations	
CUCSF.Custom.cupPreferredDefaultGroup	
CUCSF.Custom.enableDNDNoNotify	
Presence.persistAwayWhenOfflineFlag	
Presence.calendar	
Reachability.autoDetectIdle	
MeetingPlace.password.encrypted	Deprecated
MeetingPlace.userName	Deprecated
VoiceMail.password.encrypted	Deprecated
VoiceMail.userName	Deprecated
Reachability.idleAfterElapsed	
IM.minimizeWindow	
IM.showWindow	
IM.showTimestamp	

CallControl.Lineid	Deprecated
CallControl.Devicename	Deprecated
Credentials.Digest	Deprecated
Presence.userName	
Presence.listName	
Presence.calendar	

Example below shows a get-user-config request.

Get-User-Config request

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                  xmlns:epas="urn:cisco:epas:soap"
                  xmlns="urn:cisco:epas:soap">
  <soapenv:Header>
    <session-key>b189aa60-24e3-4e66-a3b6-d8488235ba47</session-key>
  </soapenv:Header>
  <soapenv:Body>
    <get-user-config/>
  </soapenv:Body>
</soapenv:Envelope>
```

Example below shows a get-user-config response for **SOAP interface version 7.0 & 8.0.**

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
    <epas:get-user-config-resp xmlns:epas="urn:cisco:epas:soap">
      <epas:property name="Directory.MaxResults" is-public="false">1000</epas:property>
      <epas:property name="Directory.MaxTime" is-public="false">30</epas:property>
      <epas:property name="Presence.inPersistentState" is-public="false">false</epas:property>
      <epas:property name="Presence.displayName" is-public="false">Jose
Sebastian</epas:property>
      <epas:property name="Presence.persistAwayWhenOfflineFlag" is-
public="false">false</epas:property>
      <epas:property name="CUCSF.Custom.cupPreferredDefaultGroup" is-
public="true">General</epas:property>
      <epas:property name="CUCSF.Custom.enableDNDNoNotify" is-
public="true">true</epas:property>
      <epas:property name="CallControl.Lineid" is-public="false">12345</epas:property>
      <epas:property name="CallControl.Devicename" is-public="false"/>
```

```
      <epas:property name="Credentials.Digest" is-public="false"/>
      <epas:property name="Presence.userName" is-public="false">jose</epas:property>
      <epas:property name="Presence.listName" is-public="false">jose-contacts</epas:property>
      <epas:property name="Presence.calendar" is-public="false">disabled</epas:property>
    </epas:get-user-config-resp>
  </env:Body>
</env:Envelope>
```

Example below shows a get-user-config response for **SOAP interface version 9.0.**

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
    <epas:get-user-config-resp xmlns:epas="urn:cisco:epas:soap">
      <epas:property name="Directory.MaxResults" is-public="false">1000</epas:property>
      <epas:property name="Directory.MaxTime" is-public="false">30</epas:property>
      <epas:property name="Presence.inPersistentState" is-public="false">false</epas:property>
      <epas:property name="Presence.displayName" is-public="false">Jose
Sebastian</epas:property>
      <epas:property name="Presence.persistAwayWhenOfflineFlag" is-
public="false">false</epas:property>
      <epas:property name="CUCSF.Custom.cupPreferredDefaultGroup" is-
public="true">General</epas:property>
      <epas:property name="CUCSF.Custom.enableDNDNoNotify" is-
public="true">true</epas:property>
      <epas:property name="Presence.userName" is-public="false">jose</epas:property>
      <epas:property name="Presence.listName" is-public="false">jose-contacts</epas:property>
      <epas:property name="Presence.calendar" is-public="false">disabled</epas:property>
    </epas:get-user-config-resp>
  </env:Body>
</env:Envelope>
```

The set-user-config request sets per-user configuration information.

Note: Deprecated parameters should not be set from SOAP interface version 9.0

Example below shows a set-user-config request.

Set-User-Config request

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                  xmlns:epas="urn:cisco:epas:soap"
                  xmlns="urn:cisco:epas:soap">
  <soapenv:Header>
    <session-key>e3b018a6-9a30-ec91-043a-f0c386a2579c</session-key>
  </soapenv:Header>
  <soapenv:Body>
    <epas:set-user-config>
      <epas:property is-public="false" name="Presence.MRUListenPort">50007</epas:property>
    </epas:set-user-config>
```

```
  </soapenv:Body>
</soapenv:Envelope>
```


Example below shows a set-user-config response.

Set-User-Config response

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
    <epas:set-user-config-resp  xmlns:epas="urn:cisco:epas:soap" >
      <epas:property name="Presence.MRUListenPort" status-code="0"/>
    </epas:set-user-config-resp>
  </env:Body>
</env:Envelope>
```

About the Contact Management Functions

Add Contact Request

Example 16 shows an add-contact request. A new group will automatically be created if the group name specified in this add-contact request does not already exist.

Example 16 Add-Contact request

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:epas="urn:cisco:epas:soap"
            xmlns="urn:cisco:epas:soap">
  <soapenv:Header>
    <session-key>b189aa60-24e3-4e66-a3b6-d8488235ba47</session-key>
  </soapenv:Header>
  <soapenv:Body>
    <add-contact>
      <group name="work">
        <persona-id index="200" display-on-phone="false" nickname="user2
nickname">soapuser2</persona-id>
        <persona-id nickname="user3's nickname">soapuser3</persona-id>
        <persona-id display-on-phone="true">soapuser4</persona-id>
        <persona-id index="-1">soapuser5</persona-id>
        <persona-id>nonexisting</persona-id>
      </group>
      <group name="new customers">
        <persona-id index="202" display-on-phone="true" nickname="another user2's
nickname">soapuser2</persona-id>
        <persona-id display-on-phone="false">soapuser3</persona-id>
```

```
        <persona-id index="200" display-on-phone="false" nickname="non
existing">nonexisting</persona-id>
      </group>
      <group name="friends">
        <persona-id nickname="what">soapuser2</persona-id>
        <persona-id>whatever</persona-id>
      </group>
      <group name="">
        <persona-id display-on-phone="false" nickname="what">soapuser4</persona-id>
        <persona-id nickname="dave" index="32">soapuser5</persona-id>
        <persona-id>soapuser3</persona-id>
        <persona-id display-on-phone="true" index="-33.3">soapuser1</persona-id>
      </group>
    </add-contact>
  </soapenv:Body>
</soapenv:Envelope>
```

Example 17 shows an add-contact response.

Example 17 Add-Contact response

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
     <epas:add-contact-resp  xmlns:epas="urn:cisco:epas:soap" >
        <epas:status persona-id="soapuser2" group="work" index="200" nickname="user2
nickname" display-on-phone="false">succeeded</epas:status>
        <epas:status persona-id="soapuser3" group="work" nickname="user3&apos;s
nickname">succeeded</epas:status>
        <epas:status persona-id="soapuser4" group="work" display-on-
phone="true">succeeded</epas:status>
        <epas:status persona-id="soapuser5" group="work" index="-1">succeeded</epas:status>
        <epas:status persona-id="nonexisting" group="work">Contact does not exist
in.</epas:status>
        <epas:status persona-id="soapuser2" group="new customers" index="202"
nickname="another user2&apos;s nickname" display-on-phone="true">succeeded</epas:status>
        <epas:status persona-id="soapuser3" group="new customers" display-on-
phone="false">succeeded</epas:status>
        <epas:status persona-id="nonexisting" group="new customers" index="200"
nickname="non existing" display-on-phone="false">Contact does not exist in.</epas:status>
        <epas:status persona-id="soapuser2" group="friends" nickname="what">The Contact Group
does not exist.</epas:status>
        <epas:status persona-id="whatever" group="friends">Contact does not exist
in.</epas:status>
        <epas:status persona-id="soapuser4" group="" nickname="what" display-on-
phone="false">succeeded</epas:status>
        <epas:status persona-id="soapuser5" group="" index="32"
nickname="dave">succeeded</epas:status>
        <epas:status persona-id="soapuser3" group="">succeeded</epas:status>
```

```
        <epas:status persona-id="soapuser1" group="" index="-33.3" display-on-
phone="true">succeeded</epas:status>
      </epas:add-contact-resp>
   </env:Body>
</env:Envelope>
```

Delete Contact Request

The example below shows a delete-contact request. If this is the last contact in any of the groups specified in this delete-contact request, then the group will be automatically deleted.

Example 18 Delete-Contact request

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:epas="urn:cisco:epas:soap"
            xmlns="urn:cisco:epas:soap">
  <soapenv:Header>
    <session-key>b189aa60-24e3-4e66-a3b6-d8488235ba47</session-key>
  </soapenv:Header>
  <soapenv:Body>
    <delete-contact>
      <group name="work">
        <persona-id>soapuser2</persona-id>
      </group>
      <group name="non-existing">
        <persona-id>one</persona-id>
      </group>
      <group name="Friends">
        <persona-id>soapuser5</persona-id>
        <persona-id>soapuser1</persona-id>
        <persona-id>nonexisting</persona-id>
      </group>
    </delete-contact>
  </soapenv:Body>
</soapenv:Envelope>
```

Example below shows a delete-contact response.

Example 19 Delete-Contact response

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
    <epas:delete-contact-resp xmlns:epas="urn:cisco:epas:soap">
      <epas:status ret-code="0" persona-id="soapuser2" group="work">succeeded</epas:status>
      <epas:status ret-code="0" persona-id="one" group="non-existing">succeeded</epas:status>
```

```
      <epas:status ret-code="0" persona-id="soapuser5"
group="Friends">succeeded</epas:status>
      <epas:status ret-code="0" persona-id="soapuser1"
group="Friends">succeeded</epas:status>
      <epas:status ret-code="0" persona-id="nonexisting"
group="Friends">succeeded</epas:status>
    </epas:delete-contact-resp>
  </env:Body>
</env:Envelope>
```

Get Contact Information Request

Cisco Unified CM IM and Presence can store auxiliary contact information about a user on behalf of the client application, for example information such as work phone number, home phone number, mobile phone number, email address and so on. The client application can retrieve. modify and delete the auxiliary contact information stored on Cisco Unified CM IM and Presence using the Client Configuration Web Service.

Cisco Unified CM IM and Presence only stores the auxiliary information on behalf of the client application. No auxiliary information is generated on the Cisco Unified CM IM and Presence server. The get-contact-info request allows the client application to retrieve auxiliary information about a contact, The contact information is sent as an opaque text document.

Example 20 shows a get-contact-info request.

Example 20 Get-Contact-Info request

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:epas="urn:cisco:epas:soap"
            xmlns="urn:cisco:epas:soap">
  <soapenv:Header>
    <session-key>b189aa60-24e3-4e66-a3b6-d8488235ba47</session-key>
  </soapenv:Header>
  <soapenv:Body>
    <get-contact-info>
      <persona-id domain="">4279252c-9827-46fb-8071-1d138eaaef4a</persona-id>
      <persona-id >soapuser1</persona-id>
      <persona-id domain="federatedcompany.com">john</persona-id>
      <persona-id domain="cisco.com">soapuser5</persona-id>
      <persona-id>nonexisting</persona-id>
    </get-contact-info>
  </soapenv:Body>
</soapenv:Envelope>
```

Example 21 shows a get-contact-info response.

Example 21 Get-Contact-Info response

```
<?xml version='1.0' ?>
 <env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <env:Body>
      <epas:get-contact-info-resp  xmlns:epas="urn:cisco:epas:soap"  xmlns="urn:cisco:epas:soap"
>
        <epas:contact-info persona-id="4279252c-9827-46fb-8071-1d138eaaef4a" domain=""
contentType="text/directory" version="3.0" >BEGIN:VCARD
        FN:pizz guy
        TEL;TYPE=WORK,VOICE:+1999888777
        TEL;TYPE=X-CUSTOM1,VOICE;X-LABEL=Home:+81311115555
        VERSION:3.0
        END:VCARD
        </epas:contact-info>
        <epas:contact-info persona-id="soapuser1" contentType="text/directory" version="3.0"
>BEGIN:VCARD
        FN:soapuser1
        TEL;TYPE=X-CUSTOM1,VOICE;X-LABEL=Home:+81311111111
        VERSION:3.0
        END:VCARD
        </epas:contact-info>
        <epas:contact-info persona-id="john" domain="federatedcompany.com"
contentType="text/directory" version="3.0" >BEGIN:VCARD
        FN:john
        TEL;TYPE=X-CUSTOM1,VOICE;X-LABEL=Home:+81311115555
        TEL;TYPE=X-CUSTOM2,VOICE;X-LABEL=Fax:+81311115556
        VERSION:3.0
        END:VCARD
        </epas:contact-info>
        <epas:contact-info persona-id="soapuser5" domain="cisco.com"
contentType="text/directory" version="3.0" >BEGIN:VCARD
        FN:user5
        TEL;TYPE=X-CUSTOM1,VOICE;X-LABEL=Home:+81311115555
        TEL;TYPE=X-CUSTOM2,VOICE;X-LABEL=Fax:+81311115556
        VERSION:3.0
        END:VCARD
        </epas:contact-info>
      </epas:get-contact-info-resp>
   </env:Body>
</env:Envelope>
```

Modify Contact Information Request

The modify-contact-info request allows the client application to modify auxiliary information about a contact. The contact information is sent as an opaque text document.

Example below shows a modify-contact-info request.

Example 22 Modify-Contact-Info request

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:epas="urn:cisco:epas:soap"
            xmlns="urn:cisco:epas:soap">
   <soapenv:Header>
      <session-key>b189aa60-24e3-4e66-a3b6-d8488235ba47</session-key>
   </soapenv:Header>
   <soapenv:Body>
      <modify-contact-info >
         <contact-info persona-id="soapuser2" contentType="text/directory"
version="3.0">begin:vcard
         fn:John Doe
         n:John Doe
         tel;type=cell:+1555334018
         version:3.0
         end:vcard
         </contact-info>
         <contact-info persona-id="one" domain="another.com" contentType="text/directory"
>begin:vcard
         fn:Joe Bloggs
         n:Bloggs, Joe
         tel:+3535551234
         version:3.0
         end:vcard
         </contact-info>
      </modify-contact-info>
   </soapenv:Body>
</soapenv:Envelope>
```

Example 23 shows a modify-contact-info response.

Example 23 Modify-Contact-Info response

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
     <modify-contact-info-resp>
        <status ret-code="0" persona-id="soapuser2" group="work">succeeded</status>
              <status ret-code="18001" persona-id="one" domain="another.com" group="non-
existing">failed</status>
           </modify-contact-info-resp>
  </env:Body>
</env:Envelope>
```

Delete Contact Information Request

The delete-contact-info request allows the client application to delete auxiliary information about a contact.

The example below shows a delete-contact-info request.

Example 24 Delete-Contact-Info request

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                  xmlns:epas="urn:cisco:epas:soap"
                  xmlns="urn:cisco:epas:soap">
  <soapenv:Header>
    <session-key>b189aa60-24e3-4e66-a3b6-d8488235ba47</session-key>
  </soapenv:Header>
  <soapenv:Body>
    <delete-contact-info>
      <persona-id domain="">4279252c-9827-46fb-8071-1d138eaaef4a</persona-id>
      <persona-id >soapuser1</persona-id>
      <persona-id domain="federatedcompany.com">john</persona-id>
      <persona-id domain="cisco.com">soapuser5</persona-id>
      <persona-id>nonexisting</persona-id>
    </delete-contact-info>
  </soapenv:Body>
</soapenv:Envelope>
```

The example below shows a delete-contact-info response.

Example 25 Delete-Contact-Info response

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
    <epas:delete-contact-info-resp xmlns:epas="urn:cisco:epas:soap"  xmlns="urn:cisco:epas:soap"
>
      <status ret-code="0" persona-id="4279252c-9827-46fb-8071-1d138eaaef4a" domain=""
>succeeded</status>
      <status ret-code="0" persona-id="soapuser1" >succeeded</status>
      <status ret-code="0" persona-id="john" domain="federatedcompany.com"
>succeeded</status>
      <status ret-code="0" persona-id="soapuser5" domain="cisco.com" >succeeded</status>
      <status ret-code="18112" persona-id="nonexisting" >failed</status>
    </epas:delete-contact-info-resp>
  </env:Body>
</env:Envelope>
```

Get Non Presence Aware Contacts

Non Presence Aware Contacts are contacts for which no presence can be retrieved. These contacts are useful for storing personal contact information, even when it is not possible to get presence for a contact. Clients can use the phone number stored by the user to initiate phone calls.

This message allows the client application to retrieve a list of non presence aware contacts.

The example below shows a get-non-presence-aware-contacts request

Example 26 Get-Non-Presence-Aware-Contacts request

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:urn="urn:cisco:epas:soap">
  <soap:Header>
    <urn:session-key>1b2a83e2-55cb-3f64-fe10-aad4e7152bf0</urn:session-key>
  </soap:Header>
  <soap:Body>
    <urn:get-non-presence-aware-contacts/>
  </soap:Body>
</soap:Envelope>
```

The example below shows a get-non-presence-aware-contacts response

Example 27 Get-Non-Presence-Aware-Contacts response

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <env:Body>
      <epas:get-non-presence-aware-contacts-resp xmlns:epas="urn:cisco:epas:soap">
         <epas:group name="General123">
            <epas:persona-id nickname="Test Again">6c840874-3a65-48d4-ace4-
67e311e6c039</epas:persona-id>
            <epas:persona-id nickname="test4 PG 4 - changed">afcb5f28-c4ff-4532-a6cb-
abf3ac9df8cc</epas:persona-id>
         </epas:group>
         <epas:group name="Group44">
            <epas:persona-id nickname="John B Doe">37dbe62e-c030-48d6-b2f8-
28650495a428</epas:persona-id>
            <epas:persona-id nickname="Someone Else">c0644330-ba71-4b2e-913e-
67eea7e7d9a0</epas:persona-id>
         </epas:group>
         <epas:group name="another - changed">
            <epas:persona-id nickname="Someone Else">c0644330-ba71-4b2e-913e-
67eea7e7d9a0</epas:persona-id>
         </epas:group>
      </epas:get-non-presence-aware-contacts-resp>
   </env:Body>
</env:Envelope>
```

Get Federated Domains Request

This message allows the client application to retrieve either a list of the permitted federated domains, a list of blocked federated domains, or a list of all federated domains configured on Cisco

Unified CM IM and Presence. The type of list requested is determined using the "auth-policy" parameter. The valid values of the "auth-policy" parameter are "allowed", "blocked" or "all".

Federated domains with an authorization policy state of "pending" on Cisco Unified CM IM and Presence will be sent as "allowed" for this command.

Example below shows a get-federated-domains request.

Example 28 Get-Federated-Domains request

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                  xmlns:epas="urn:cisco:epas:soap"
                  xmlns="urn:cisco:epas:soap">
  <soapenv:Header>
    <session-key>b189aa60-24e3-4e66-a3b6-d8488235ba47</session-key>
  </soapenv:Header>
  <soapenv:Body>
    <get-federated-domains auth-policy="all"/>
  </soapenv:Body>
</soapenv:Envelope>
```

The example below shows a get-federated-domains response.

Example 29 Get-Federated-Domains response

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
    <get-federated-domains-resp>
      <domain name="cisco.com" auth-policy="allowed">
        <description>Cisco Systems</description>
      </domain>
      <domain name="DomainA.net" auth-policy="blocked">
        <description>DomainA Users</description>
      </domain>
      <domain name="corporateX.com" auth-policy="allowed">
        <description>CorporateX Network</description>
      </domain>
    </get-federated-domains-resp>
  </env:Body>
</env:Envelope>
```

Get Dialing Rules Request

The get-dialing-rules request is used to download application dialing rules from the associated Cisco Unified Communications Manager server. On receipt of the get-dialing-rules request Cisco

Unified CM IM and Presence queries the associated Cisco Unified Communications Manager to retrieve the dialing rules for the client application.

Note: This API has been deprecated in SOAP interface version 9.0, trying to invoke this API from 9.0 version and above will fail.

Example below shows a get-dialing-rules request.

Example 30 Get-Dialing-Rules request

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                  xmlns:epas="urn:cisco:epas:soap"
                  xmlns="urn:cisco:epas:soap">
  <soapenv:Header>
    <session-key>b189aa60-24e3-4e66-a3b6-d8488235ba47</session-key>
  </soapenv:Header>
  <soapenv:Body>
    <get-dialing-rules/>
  </soapenv:Body>
</soapenv:Envelope>
```

The example below shows a get-dialing-rules response.

Example 31 Get-Dialing-Rules response

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <env:Body>
      <epas:get-dialing-rules-resp xmlns:epas="urn:cisco:epas:soap">
         <epas:rule>
            <epas:type>1</epas:type>
            <epas:name>App dialing rule</epas:name>
            <epas:description>App dialing rule</epas:description>
            <epas:priority>1</epas:priority>
            <epas:number-begins-with>89</epas:number-begins-with>
            <epas:number-of-digits>1</epas:number-of-digits>
            <epas:digits-to-be-removed>0</epas:digits-to-be-removed>
            <epas:prefix>7</epas:prefix>
         </epas:rule>
         <epas:rule>
            <epas:type>2</epas:type>
            <epas:name>Directory dialing rule</epas:name>
            <epas:description>Directory dialing rule</epas:description>
            <epas:priority>1</epas:priority>
            <epas:number-begins-with>97876</epas:number-begins-with>
            <epas:number-of-digits>2</epas:number-of-digits>
            <epas:digits-to-be-removed>2</epas:digits-to-be-removed>
            <epas:prefix>98</epas:prefix>
         </epas:rule>
      </epas:get-dialing-rules-resp>
```

```
   </env:Body>
</env:Envelope>
```

Get Licensing Features Request

The get-licensing-features request is used to retrieve the licensing features assigned to a user.

Note: This API has been deprecated in SOAP interface version 9.0, trying to invoke this API from 9.0 version and above will fail.

The example below shows a get-licensing-features request.

Example 32 Get-Licensing-Features request

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:epas="urn:cisco:epas:soap"
            xmlns="urn:cisco:epas:soap">
  <soapenv:Header>
    <session-key>b189aa60-24e3-4e66-a3b6-d8488235ba47</session-key>
  </soapenv:Header>
  <soapenv:Body>
    <get-licensing-features/>
  </soapenv:Body>
</soapenv:Envelope>
```

The example below shows a get-licensing-features response.

Example 33 Get-Licensing-Features Response

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
    <epas:get-licensing-features-resp  xmlns:epas="urn:cisco:epas:soap" >
      <epas:base>true</epas:base>
      <epas:im>true</epas:im>
      <epas:audio>false</epas:audio>
      <epas:video>false</epas:video>
    </epas:get-licensing-features-resp>
  </env:Body>
</env:Envelope>
```

About the Access Control List Functions

Cisco Unified CM IM and Presence associates an Access Control List (ACL) with each presence rule. The list contains a group of watchers who are granted the privilege of watching the user (the owner of the presence rule).

The presence rule name can one of the following reserved rule names:

"allowed" - This rule retrieves the list of watchers who are allowed to view the presence of the user.

"politeblocking" - This rule retrieves the list of watchers who are not allowed view the presence of the user. The presence status of the user will always be "unavailable" for these watchers.

Get ACL Request

The get-acl request is used to get all watchers of a presence rule.

The example below shows the get-acl request.

Example 34 Get-ACL request

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                  xmlns:epas="urn:cisco:epas:soap"
                  xmlns="urn:cisco:epas:soap">
  <soapenv:Header>
    <session-key>396fa7d8-56eb-46ed-ad8d-39df966c85d2</session-key>
  </soapenv:Header>
  <soapenv:Body>
    <get-acl>
      <rule name="allowed"/>
      <rule name="politeblocking"/>
    </get-acl>
  </soapenv:Body>
</soapenv:Envelope>
```

The example below shows the get-acl response.

Example 35 Get-ACL response

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
    <epas:get-acl-resp xmlns:epas="urn:cisco:epas:soap" xmlns="urn:cisco:epas:soap">
      <rule name="allowed">
        <acl>sip:john@federatedcompany.com</acl>
        <acl>sip:paul@federatedcompany.com</acl>
        <acl>sip:john@anotherfederatedcompany.com</acl>
      </rule>
      <rule name="politeblocking"></rule>
    </epas:get-acl-resp>
  </env:Body>
</env:Envelope>
```

Add ACL Request

The add-acl request adds watchers to the Access Control List associated with the presence rule. If the watcher already belongs to the ACL of the presence rule specified, no change is made. If the watcher already belongs to the ACL of a different presence rule, it is removed from the existing rule and added to the specified rule.

The watcher can be passed in as a full URI, for example sip:soapuser2@mydomain.com, or can be passed without the sip prefix, for example, soapuser3@cisco.com. The full *user@domain.com* string should be used, even when referring to a local Cisco Unified CM IM and Presence user.

If a watcher string is passed in without the '@' symbol, it is treated as a domain, for example *yourdomain.com* .

If the username of the watcher contains any special characters, these should be escaped using URL escape codes, for example, the '@' symbol is replaced with '%40' in soapuser4%40mydepartment@mydomain.com.

The example below shows an add-acl request.

Example 36 Add-ACL request

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:epas="urn:cisco:epas:soap"
            xmlns="urn:cisco:epas:soap">
  <soapenv:Header>
    <session-key>396fa7d8-56eb-46ed-ad8d-39df966c85d2</session-key>
  </soapenv:Header>
  <soapenv:Body>
    <add-acl>
      <rule name="allowed">
        <watcher>sip:soapuser2@mydomain.com</watcher>
        <watcher>soapuser3@cisco.com</watcher>
        <watcher>soapuser4%40mydepartment@mydomain.com</watcher>
        <watcher>soapuser5@cisco.com</watcher>
        <watcher>g@cisco.com</watcher>
        <watcher>j@cisco.com</watcher>
      </rule>
    </add-acl>
  </soapenv:Body>
</soapenv:Envelope>
```

The example below shows the add-acl response.

Example 37 Add-ACL response

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <env:Body>
      <epas:add-acl-resp xmlns:epas="urn:cisco:epas:soap">
         <epas:rule name="allowed">
```

```
            <epas:status ret-code="0"
watcher="sip:soapuser2@mydomain.com">succeeded</epas:status>
            <epas:status ret-code="0" watcher="soapuser3@cisco.com">succeeded</epas:status>
            <epas:status ret-code="0"
watcher="soapuser4%40mydepartment@mydomain.com">succeeded</epas:status>
            <epas:status ret-code="0" watcher="soapuser5@cisco.com">succeeded</epas:status>
            <epas:status ret-code="18065" watcher="g@cisco.com">failed</epas:status>
            <epas:status ret-code="18065" watcher="j@cisco.com">failed</epas:status>
          </epas:rule>
        </epas:add-acl-resp>
    </env:Body>
</env:Envelope>
```

Delete ACL Request

The example below shows a delete-acl request.

Example 38 Delete-ACL request

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
              xmlns:xsd="http://www.w3.org/2001/XMLSchema"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xmlns:epas="urn:cisco:epas:soap"
              xmlns="urn:cisco:epas:soap">
  <soapenv:Header>
    <session-key>396fa7d8-56eb-46ed-ad8d-39df966c85d2</session-key>
  </soapenv:Header>
  <soapenv:Body>
    <delete-acl>
      <rule name="politeblocking">
        <watcher> sip:soapuser2@mydomain.com </watcher>
        <watcher>soapuser5@cisco.com</watcher>
      </rule>
    </delete-acl>
  </soapenv:Body>
</soapenv:Envelope>
```

This example shows a delete-acl response.

Example 39 Delete-ACL response

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
    <epas:delete-acl-resp xmlns:epas="urn:cisco:epas:soap">
      <epas:rule name="politeblocking">
        <epas:status ret-code="0"
watcher="sip:soapuser2@mydomain.com">succeeded</epas:status>
```

```
        <epas:status ret-code="0" watcher="soapuser5@cisco.com">succeeded</epas:status>
      </epas:rule>
    </epas:delete-acl-resp>
  </env:Body>
</env:Envelope>
```

About the Calendaring Functions

The user can get and set calendaring options through the Client Configuration Web Service. The get-user-config-resp returns the calendaring setting for a user.

Get Calendaring Request

The example below shows a get-calendaring request.

Example 40 Get-Calendaring

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                  xmlns:epas="urn:cisco:epas:soap"
                  xmlns="urn:cisco:epas:soap">
  <soapenv:Header>
    <session-key>2a08cef6-0f85-4ab3-871c-663269f1d49d</session-key>
  </soapenv:Header>
  <soapenv:Body>
    <get-calendaring/>
  </soapenv:Body>
</soapenv:Envelope>
```

This shows a get-calendaring response.

Example 41 Get-Calendaring Response

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
    <epas:get-calendaring-resp  xmlns:epas="urn:cisco:epas:soap" >
      true
    </epas:get-calendaring-resp>
  </env:Body>
</env:Envelope>
```

Set Calendaring Request

This example shows a set-calendaring request.

Example 42 Set-Calendaring

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:epas="urn:cisco:epas:soap"
            xmlns="urn:cisco:epas:soap">
  <soapenv:Header>
    <session-key>2a08cef6-0f85-4ab3-871c-663269f1d49d</session-key>
  </soapenv:Header>
  <soapenv:Body>
    <set-calendaring>true</set-calendaring>
  </soapenv:Body>
</soapenv:Envelope>
```

This example shows a set-calendaring response.

Example 43 Set-Calendaring Response

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
    <epas:set-calendaring-resp  xmlns:epas="urn:cisco:epas:soap" >
      succeeded
    </epas:set-calendaring-resp>
  </env:Body>
</env:Envelope>
```

Phone Presence Preferences Functions

A user can get or set their phone presence preferences through the Client Configuration Web Service.

When set to “Always”, it is an indication that the user wants to include his/her phone presence in the composed presence for the user. When set to “Never”, it is an indication that the user does not want to include his/her phone presence in the composed presence. When set to “When Logged In”, it is an indication that the user wants his phone presence to be included when he/she is logged into the presence client.

Get Phone Presence Preferences Request

The example below shows a get-phone-preference request.

Example 44 Get-Phone-Preference request

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
            xmlns:epas="urn:cisco:epas:soap"
            xmlns="urn:cisco:epas:soap">
  <soapenv:Header>
    <session-key>2a08cef6-0f85-4ab3-871c-663269f1d49d</session-key>
  </soapenv:Header>
  <soapenv:Body>
    <get-phone-preference/>
  </soapenv:Body>
</soapenv:Envelope>
```

The example below shows a get-phone-preference response

Example 45 Get-Phone-Preference response

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
    <epas:get-phone-preference-resp  xmlns:epas="urn:cisco:epas:soap"
xmlns="urn:cisco:epas:soap" >
      <include-phone-presence>Always</include-phone-presence>
    </epas:get-phone-preference-resp>
  </env:Body>
</env:Envelope>
```

Set Phone Presence Preferences Request

The example below shows a set-phone-preference request.

Example 46 Set-Phone-Preference request

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:urn="urn:cisco:epas:soap">
  <soap:Header>
    <urn:session-key>f349f85f-e938-0129-e730-8ddfaa7c3eba</urn:session-key>
  </soap:Header>
  <soap:Body>
    <urn:set-phone-preference>
      <urn:include-phone-presence>Always</urn:include-phone-presence>
    </urn:set-phone-preference>
  </soap:Body>
</soap:Envelope>
```

The example below shows a set-phone-preference response

Example 47 Set-Phone-Preference response

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
```

```
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
    <epas:set-phone-preference-resp  xmlns:epas="urn:cisco:epas:soap"
xmlns="urn:cisco:epas:soap" >
      <status ret-code="0" >succeeded</status>
    </epas:set-phone-preference-resp>
  </env:Body>
</env:Envelope>
```

Profile Search Request

The profile search service allows a user to search the public data of other users.

The example below shows a search-profile request.

Example 48 Search-Profile request

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:epas="urn:cisco:epas:soap"
            xmlns="urn:cisco:epas:soap">
  <soapenv:Header>
    <session-key>b189aa60-24e3-4e66-a3b6-d8488235ba47</session-key>
  </soapenv:Header>
  <soapenv:Body>
    <search-profile>
      <persona id="soapuser2"/>
      <persona id="unknown"/>
      <persona id="soapuser3"/>
      <property name="Preferred.Contact"/>
      <property name="UI.Color.Background"/>
      <property name="PhoneNumber.Primary"/>
    </search-profile>
  </soapenv:Body>
</soapenv:Envelope>
```

Example below shows a search-profile response.

Example 49 Search-Profile response

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
    <epas:search-profile-resp  xmlns:epas="urn:cisco:epas:soap" >
      <epas:persona id="soapuser2">
        <epas:property name="Preferred.Contact">soapuser 2&amp;apos;s preferred contact is
email</epas:property>
```

```
        <epas:property name="PhoneNumber.Primary">soapuser 2&apos;s primary phone
number</epas:property>
      </epas:persona>
      <epas:persona id="unknown"></epas:persona>
      <epas:persona id="soapuser3">
        <epas:property name="Preferred.Contact">soapuser 3&apos;s preferred contact is
phone</epas:property>
        <epas:property name="PhoneNumber.Primary">soapuser 3&apos;s primary phone
number</epas:property>
      </epas:persona>
    </epas:search-profile-resp>
  </env:Body>
</env:Envelope>
```

Get Cluster Information Request

This request allows a user to retrieve information about the Cisco Unified CM IM and Presence cluster, such as Cisco Unified CM IM and Presence node names and their associated realms.

Example below shows a get-cluster-info request

Example 50 Get-Cluster-Info request

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:urn="urn:cisco:epas:soap">
  <soap:Header>
    <urn:session-key> b189aa60-24e3-4e66-a3b6-d8488235ba47</urn:session-key>
  </soap:Header>
  <soap:Body>
    <urn:get-cluster-info/>
  </soap:Body>
</soap:Envelope>
```

Example below shows a get-cluster-info response

Example 51 Get-Cluster-Info response

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:urn="urn:cisco:epas:soap">
  <soap:Header/>
  <soap:Body>
    <urn:get-cluster-info-resp>
      <!--1 to 6 repetitions:-->
      <urn:Node Name="?" isPub="?" subclusterName="?">
        <!--1 to 2 repetitions:-->
        <urn:Realm isActive="?">?</urn:Realm>
      </urn:Node>
    </urn:get-cluster-info-resp>
  </soap:Body>
</soap:Envelope>
```

About the Roster Management AXL-API Services

The Roster Management Web Service is an interface that allows client applications to manage user migration and onboarding on Cisco Unified CM IM and Presence.

The Roster Management Web Service is a Simple Object Access Protocol (SOAP) interface. The request and response messages are sent in the form of XML. A client application sends a SOAP request, the web service processes the request and sends a SOAP response.

Overview of Interfaces

A client application uses the Roster Management AXL-API Web Service to perform the following functions:

Contact Rename

Contact Add

The wdsl for conatact add, and rename may be obtained from an IM&P server. Using the following URL, using your hostname,

https:// **hostname** /axl/services/AXLAPIService?wsdl

Contact Rename

For contact rename, the userid or the domain name may be changed, or both. Example soap request/response shown below.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="http://www.cisco.com/AXL/API/10.5">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:updateContact sequence="?">
      <oldContact>john@ibm.com</oldContact>
      <newContact>john@cisco.com</newContact>
    </ns:updateContact>
  </soapenv:Body>
</soapenv:Envelope>
```

Request : An example to rename the contact from john@ibm.com to john@cisco.com

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns:updateContactRespons xmlns:ns="http://www.cisco.com/AXL/API/10.5">
      <totalRows>100</totalRows>
      <failedRows>0</failedRows>
      <successfulRows>0</successfulRows>
    </ns:updateContactResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Response : returns totalRows,failedRows and successfulRows, total 100 contacts requested to be be updated, but 100 updates passed and 0 failed. The number of total rows updated indicates the contact is in 100 buddy lists

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>soapenv:Server</faultcode>
      <faultstring>Cisco XCP router must be started on all nodes in the
            local cluster</faultstring>
      <detail>
        <axlError>
          <axlcode>5102</axlcode>
          <axlmessage>Cisco XCP router must be started on all nodes in
                the local cluster</axlmessage>
          <request>updateContact</request>
        </axlError>
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

Response (Error Case) : The XCP router service is down, client logic may be included to retry after starting the XCP service

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>soapenv:Server</faultcode>
      <faultstring>Invalid Jid: newContact must be a valid
            JID</faultstring>
      <detail>
        <axlError>
          <axlcode>5101</axlcode>
          <axlmessage>Invalid Jid: newContact must be a valid
                JID</axlmessage>
          <request>updateContact</request>
        </axlError>
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

Response (Error Case) : An invalid JID is supplied for old / new contact

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>soapenv:Server</faultcode>
      <faultstring>oldContact contactname@jid.com and newContact
            contactname@jid.com be must different</faultstring>
      <detail>
```

```
        <axlError>
          <axlcode>5101</axlcode>
          <axlmessage>oldContact contactname@jid.com and newContact
                contactname@jid.com be must different</axlmessage>
          <request>updateContact</request>
        </axlError>
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

Response (Error Case) : oldContact and newContact are the same JID

Contact Add

Below is an example to add 'james@cisco.com' to jack@cisco.com 's contact list in the group 'clientDev'. Note : The nickname is an optional parameter.

Add Contact Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="http://www.cisco.com/AXL/API/10.5">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:addContact sequence="?">
      <userJid>jack@cisco.com</userJid>
      <contact>james@cisco.com</contact>
      <groupName>clientDev</groupName>
      <!--Optional:-->
      <nickname>james2</nickname>
    </ns:addContact>
  </soapenv:Body>
</soapenv:Envelope>
```

Add contact Response

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns:addContactResponse sequence="?"
                           xmlns:ns="http://www.cisco.com/AXL/API/10.5">
      <message>Success</message>
    </ns:addContactResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Response (Error Case) : cannot find user, userJid "jack@cisco.com" in this example

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>soapenv:Server</faultcode>
```

```
      <faultstring>Unable to find user</faultstring>
      <detail>
        <axlError>
          <axlcode>5003</axlcode>
          <axlmessage>User not found</axlmessage>
          <request>addContact</request>
        </axlError>
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

Response (Error Case) : service is down, client logic may be included to retry after starting the XCP service

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>soapenv:Server</faultcode>
      <faultstring>XCP service is down</faultstring>
      <detail>
        <axlError>
          <axlcode>5003</axlcode>
          <axlmessage>XCP service is down</axlmessage>
          <request>addContact</request>
        </axlError>
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

Performance Considerations

For both Contact Add and Contact Rename, a rate of 1 contact per second results in a 5% to 7% CPU, up to a recommended maximum rate of 4 per second. The recommended maximum rate of 2 contacts per second may be updated in a non-maintenance window and maximum rate of 4 contacts per second in a maintenance window.

Troubleshooting Help

Via IM&P admin console, it is possible to check the input AXP-API SOAP request is valid, along with other debug information, by checking the AXL Web service log.

Once logged into the IM&P admin CLI. Type

```
> admin : file list  activelog tomcat/logs/axl/log4j/
```

This will list the existing log files, choose the latest axl log file (example shown axl00001.log)

```
> admin : file view  activelog tomcat/logs/axl/log4j/axl00001.log
```


About the Presence Web Service

The Presence Web Service is an open interface that allows client applications to share user presence information with Cisco Unified CM IM and Presence. Third party developers use this interface to build client applications that can send and retrieve updates to the presence state of a user.

The Presence Web Service manages user presence on a Cisco Unified CM IM and Presence server. Users can set their own presence states and receive notifications of changes to the presence state of their contacts.

The Presence Web Service supports the exchange of user presence information from a Cisco Unified CM IM and Presence node using either a Simple Object Access Protocol (SOAP) interface or a REST (XML/HTTP) interface. A client application sends an XML request; the web service processes the request and sends a response. The Presence Web Service sends all messages securely over HTTPS ; it can also be accessed over HTTP.

In 10.5, IPv6 support has been added to the Presence Web Services.

Login and Authentication

A client application logs in to Cisco Unified CM IM and Presence as an application user using a valid username and password. On successful login, the application user is passed a unique session key from Cisco Unified CM IM and Presence.

A registered application user can log in multiple end users of the client application to Cisco Unified CM IM and Presence. An application user logs in an end user by passing the session key of the application user and the username of the end user; a unique session key for the end user is returned. This login method is authenticated only if the session key that is passed back belongs to an application user, and the username of the end user is configured in the database on Cisco Unified CM IM and Presence.

On successful login, the session key is passed as a parameter of all subsequent request messages to Cisco Unified CM IM and Presence.

Each client application should have its own application user configured on Cisco Unified CM IM and Presence. If an application user is disconnected, any associated end users are automatically logged out.

Set Presence Status

The Presence Web Service allows users to set their own presence status. There are two types of presence supported, basic presence and rich presence. Rich presence is exchanged by passing a PIDF file in the message. The following basic presence states are supported:

- Available
- Busy

- Do Not Disturb
- Away
- Unavailable
- Unknown

Note that the "Unknown" state cannot be set through the Presence Web Service.

Event Notifications

A client application is notified when changes occur to the presence information of a user (such as changes to the presence status of a contact).

The client application registers a HTTP endpoint with Cisco Unified CM IM and Presence. The client application sets up a subscription for event notifications types; the subscription is associated with the registered HTTP endpoint. Cisco Unified CM IM and Presence sends a HTTP notify message to alert the client application when a subscribed event occurs. The client calls the getSubscribedPresence method on Cisco Unified CM IM and Presence to retrieve the updated data.

Note: IM and Presence will only send HTTP notify messages to well known HTTP ports as listed below, so the client application should ensure it specifies one of these ports in it's URL when registering a HTTP endpoint.

Well Known HTTP Ports

Ports	80, 443, 8080, 8081, 8443

To reduce any impact to performance that may occur on the client with large bursts of presence events, no further events are sent until the getSubscribedPresence method is called. Any further updates that occur while waiting for the getSubscribedPresence request are stored on Cisco Unified CM IM and Presence and incorporated into the getSubscribedPresence response.

The notify callback is transported over HTTP and is therefore insecure; no sensitive data is passed over this link.

The subscription and endpoint registration requests both have associated expiration timers. These timers are absolute in that the timer starts when Cisco Unified CM IM and Presence receives the request. The timers are refreshed by sending an updated expiration value (with either the subscription ID or the endpoint ID) to Cisco Unified CM IM and Presence.

The client application can also terminate the subscription to an event type, and unregister a HTTP endpoint.

While getSubscribedPresence will always return the latest data, any delays in calling the getSubscribedPresence method may result in some state transitions being missed.

Changes to the presence information of a user can also be retrieved by continually polling for presence using the getPolledPresence request; the messages are throttled so this method may not be as responsive.

About the Presence Web Service Interfaces

Client applications can exchange presence information with Cisco Unified CM IM and Presence using the following interfaces:

SOAP Interface -- This interface, based on the Simple Object Access Protocol (SOAP), passes XML request and response messages. A client application sends a SOAP request; the web service processes the request and sends a response.

REST Interface -- This REST interface, short for Representational State Transfer, passes XML data over HTTP. The data is transferred using a HTTP query parameter or by putting XML in the message body.

The Presence Web Service is available over the following ports:

over HTTPS on port 8083 (SOAP and REST for both IPv4 and Ipv6)

over HTTP on port 8082 (SOAP and REST for both IPv4 and Ipv6)

Cisco views REST as the interface mechanism of choice for the future and has incorporated this early adopter REST interface into the Cisco Unified CM IM and Presence solution.

About the SOAP Interface Methods

All SOAP requests for the Presence Web Service must be sent to the following URL:
http://<cuphost>:<port>/presence-service/soap

Login Request (Application User)

This request logs in an application user to Cisco Unified CM IM and Presence.

Input Parameters

Parameter	Description
username	The username of the application user.
password	The password of the application user.
force	This flag is used to override any existing login sessions for a user. The valid values are "true" and "false".

Output Parameters

Parameter	Description
session-key	The session key of the application user.
backup	The backup (failover) node of application user. This parameter is only present in the

Server	response if there is a backup server available.

Example 52 shows a login request for an application user.

Example 52 Login request for an application user

```
<?xml version="1.0" encoding="http://www.w3.org/2003/05/soap-envelope" standalone="no"?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
        <soapenv:Body>
                <login xmlns="urn:cisco:cup:presence:soap" force="true">
                        <username>{app-user-username}</username>
                        <password>{app-user-password}</password>
                </login>
        </soapenv:Body>
</soapenv:Envelope>
```

Example 53 shows a login response for an application user.

Example 53 Login response for an application user

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENC="http://www.w3.org/2003/05/soap-envelope" xmlns:SOAP-
ENV="http://www.w3.org/2003/05/soap-envelope/" xmlns:cup="urn:cisco:cup:presence:soap"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <SOAP-ENV:Body>
                <loginResponse>
                        <success>
                                <success backupServer="cupserver2">
                                <session-key>{app-user-session-key}</cup:session-key>
                        </success>
                </loginResponse>
        </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The session key is passed as a parameter in the SOAP header of all subsequent request messages to Cisco Unified CM IM and Presence. The session key is used as an authorization mechanism for each request.

Login Request (Application User Logging in End User)

This request is used for an application user logging in an end user to Cisco Unified CM IM and Presence; the session key of the application user is passed in the request.

Input Parameters

Parameter	Description

username	The username of the end user.
app-session-id	The session key of the application user.
force	This flag is used to override any existing login sessions for a user. The valid values are "true" and "false".

Output Parameters

Parameter	Description
session-key	The session key of the end user.
backupServer	The backup (failover) node of end user. This parameter is only present in the response if there is a backup server available.
primaryServer	The primary node of the end user. This parameter is part of the redirect login response only.

For failover purposes, an end user is configured with a primary server and a backup server on Cisco Unified CM IM and Presence. On a successful login, in addition to a session key, the end user is returned the address of the backup server. If the primary node becomes unavailable, the end user can log in to the backup server.

Example 54 shows a login request for an application user logging in an end user.

Example 54 Login request (application user logging in end user)

```
<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
  <soapenv:Body>
    <login xmlns="urn:cisco:cup:presence:soap" force="true">
      <username>{end-user-username}</username>
      <app-session-id>
        {app-user-session-key}
      </app-session-id>
    </login>
  </soapenv:Body>
</soapenv:Envelope>
```

Example 55 shows a successful login response for an application user logging in an end user.

Example 55 Login response (application user logging in end user)

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope" xmlns:SOAP-
ENC="http://www.w3.org/2003/05/soap-encoding"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:cup="urn:cisco:cup:presence:soap">
  <SOAP-ENV:Body>
    <cup:loginResponse>
      <cup:success backupServer="cupserver2">
```

```
        <cup:session-key>
          {end-user-session-key}
        </cup:session-key>
      </cup:success>
    </cup:loginResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

A redirect login response is issued if users attempt to log in to a Cisco Unified CM IM and Presence node where they are not a registered as users. The redirect login response contains information on the primary server and backup server assigned to the user. The application can log in the end user to the primary server specified in the redirect login response.

Example 56 shows a redirect login response.

Example 56 Redirect login response

```
<?xml version='1.0' ?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
  <soapenv:Body>
    <loginResponse  xmlns="urn:cisco:cup:presence:soap" >
      <redirect primaryServer="cupserver10" backupServer="cupserver11">
      </redirect>
    </loginResponse >
  </soapenv:Body>
</soapenv:Envelope>
```

Logout Request

This request logs out an application user or an end user.

Input Parameters

Parameter	Description
session-key	The session key of the user.

Example 57 shows a logout request.

Example 57 Logout request

```
<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
  <soapenv:Header>
    <session-key xmlns="urn:cisco:cup:presence:soap">
      {end-user-session-key}
    </session-key>
  </soapenv:Header>
  <soapenv:Body>
    <logout xmlns="urn:cisco:cup:presence:soap" />
```

```
  </soapenv:Body>
</soapenv:Envelope>
```


Example 58 shows a successful logout response.

Example 58 Logout response

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope" xmlns:SOAP-
ENC="http://www.w3.org/2003/05/soap-encoding"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:cup="urn:cisco:cup:presence:soap">
  <SOAP-ENV:Header></SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <cup:logoutResponse>
      <cup:status>SUCCESS</cup:status>
    </cup:logoutResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Register Endpoint Request

The registerEndPoint request registers a HTTP endpoint to be used for presence notifications. This method is only used by an application user. The session key in the request must be associated with a logged-in application user.

Input Parameters

Parameter	Description
url	The callback URL for the HTTP endpoint.
endpointID	The registration ID for the HTTP endpoint. This is passed as a parameter when extending the expiration time for the endpoint registration.
expiration	The time (in seconds) that the registration is valid for. The maximum value is 86400 seconds (24 hours). The minimum value is 3600 seconds (1 hour).

Output Parameters

Parameter	Description
endpointID	The registration ID for the HTTP endpoint.

To register a new HTTP endpoint, an *endpointID* value of zero (0) is passed in the request. The subsequent response contains the new ID number for the endpoint.

To extend the expiration time for an endpoint, call the registerEndpoint request again, passing in the existing endpoint ID and the updated expiration time; the URL should be empty.

Example 59 shows a registerEndPoint request.

Example 59 RegisterEndPoint request

```
<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
  <soapenv:Header>
    <session-key xmlns="urn:cisco:cup:presence:soap">
      {app-user-session-key}
    </session-key>
  </soapenv:Header>
  <soapenv:Body>
    <registerEndPoint xmlns="urn:cisco:cup:presence:soap">
      <url>http://appserver1@cisco.com:8080/refApp/notify.do</url>
      <endPointID>0</endPointID>
      <expiration>3600</expiration>
    </registerEndPoint>
  </soapenv:Body>
</soapenv:Envelope>
```

Example 60 shows a registerEndPoint response.

Example 60 RegisterEndPoint response

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope" xmlns:SOAP-
ENC="http://www.w3.org/2003/05/soap-encoding"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:cup="urn:cisco:cup:presence:soap">
  <SOAP-ENV:Header></SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <cup:registerEndPointResponse>
      <cup:endPointID>1</cup:endPointID>
    </cup:registerEndPointResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Unregister Endpoint Request

The unregisterEndPoint request unregisters a HTTP endpoint that is used for presence notifications. This method is only used by an application user. The session key in the request must be associated with a logged-in application user. The session key used in the unregister endpoint request must be the same session key used in the original register request.

Input Parameters

Parameter	Description

endpointID	The registration ID for the HTTP endpoint. In this case it is the endpoint that the client wishes to unregister.

Example 61 shows an unregisterEndPoint request.

Example 61 UnregisterEndPoint request

```
<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
  <soapenv:Header>
    <session-key xmlns="urn:cisco:cup:presence:soap">
      {app-user-session-key}
    </session-key>
  </soapenv:Header>
  <soapenv:Body>
    <unregisterEndPoint xmlns="urn:cisco:cup:presence:soap">
      <endPointID>1</endPointID>
    </unregisterEndPoint>
  </soapenv:Body>
</soapenv:Envelope>
```

Example 62 shows an unregisterEndPoint response.

Example 62 UnregisterEndPoint response

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope" xmlns:SOAP-
ENC="http://www.w3.org/2003/05/soap-encoding"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:cup="urn:cisco:cup:presence:soap">
  <SOAP-ENV:Header></SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <cup:unregisterEndPointResponse>
      <cup:status>SUCCESS</cup:status>
    </cup:unregisterEndPointResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Subscribe Request

The subscribe request sets up a subscription with Cisco Unified CM IM and Presence for event notifications. The notifications are sent to the HTTP endpoint registered for the client application.

Input Parameters

Parameter	Description
contactsList	The list of contact URIs that are part of the subscription.

subscriptionType	The type of event that is being subscribed to; currently only PRESENCE_NOTIFICATION is supported. The value is case sensitive.
expiration	The time (in seconds) that is subscription is valid for. The maximum value is 86400 seconds (24 hours). The minimum value is 3600 seconds (1 hour). A value of zero can be used if you are adding contacts to an existing subscription.
endpointID	The registration ID for the HTTP endpoint.
subscriptionID	The ID number of the subscription. To create a new subscription for a contact list, a subscriptionID value of zero (0) is passed in the request. The subsequent response contains the new ID of the subscription.

Output Parameters

Parameter	Description
subscriptionID	The ID number of the subscription.

To add new contacts to an existing subscription, call the subscribe request again, passing in the subscriptionID, the endpointID and a list of the additional contact URIs.

To refresh the expiration time of a subscription, call the subscribe request again, passing in the subscriptionID, the endpointID and the updated expiration time. Example 63 shows a subscribe request.

Example 63 Subscribe request

```
<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
  <soapenv:Header>
    <session-key xmlns="urn:cisco:cup:presence:soap">
      {end-user-session-key}
    </session-key>
  </soapenv:Header>
  <soapenv:Body>
    <subscribe xmlns="urn:cisco:cup:presence:soap">
      <contactsList>
        <contact contactURI="enduser1@cisco.com" />
        <contact contactURI="enduser2@cisco.com" />
        <contact contactURI="enduser3@cisco.com" />
      </contactsList>
      <subscriptionType>PRESENCE_NOTIFICATION</subscriptionType>
      <expiration>3600</expiration>
      <endPointID>1</endPointID>
      <subscriptionID>0</subscriptionID>
    </subscribe>
  </soapenv:Body>
</soapenv:Envelope>
```

Example 64 shows a subscribe response.

Example 64 Subscribe response

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope" xmlns:SOAP-
ENC="http://www.w3.org/2003/05/soap-encoding"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:cup="urn:cisco:cup:presence:soap">
  <SOAP-ENV:Header></SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <cup:subscribeResponse>
      <cup:subscriptionID>1</cup:subscriptionID>
    </cup:subscribeResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Unsubscribe Request

The unsubscribe request terminates the subscription of the client applications with Cisco Unified CM IM and Presence for event notifications. The subscription can be terminated for specified list of contacts, or for all contacts. The session key used in the unsubscribe request must be the same session key used in the original subscribe request.

Input Parameters

Parameter	Description
subscriptionID	The ID number of the subscription. In this case it is the subscription which the client wishes to unsubscribe from.
contactsList	The list of contact URIs that you wish to remove from the subscription. You can remove all contacts by either including all contacts in the contactsList or by using the unsubscribeAll parameter. See Example 66.

Example 65 shows an unsubscribe request for a specified list of contacts.

Example 65 Unsubscribe request for a specified list of contacts

```
<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
  <soapenv:Header>
    <session-key xmlns="urn:cisco:cup:presence:soap">
      {end-user-session-key}
    </session-key>
  </soapenv:Header>
  <soapenv:Body>
    <unsubscribe xmlns="urn:cisco:cup:presence:soap">
      <unsubscribeRequest>
        <subscriptionID>1</subscriptionID>
        <contactsList>
          <contact contactURI="enduser1@cisco.com" />
          <contact contactURI="enduser3@cisco.com" />
        </contactsList>
      </unsubscribeRequest>
```

```
    </unsubscribe>
  </soapenv:Body>
</soapenv:Envelope>
```

Example 66 shows an unsubscribe request for all contacts.

Example 66 Unsubscribe request for all contacts

```
<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
  <soapenv:Header>
    <session-key xmlns="urn:cisco:cup:presence:soap">
      {end-user-session-key}
    </session-key>
  </soapenv:Header>
  <soapenv:Body>
    <unsubscribe xmlns="urn:cisco:cup:presence:soap">
      <unsubscribeRequest>
        <subscriptionID>1</subscriptionID>
        <unsubscribeAll/>
      </unsubscribeRequest>
    </unsubscribe>
  </soapenv:Body>
</soapenv:Envelope>
```

Example 67 shows an unsubscribe response.

Example 67 Unsubscribe response

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope" xmlns:SOAP-
ENC="http://www.w3.org/2003/05/soap-encoding"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:cup="urn:cisco:cup:presence:soap">
  <SOAP-ENV:Header></SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <cup:unsubscribeResponse>
      <cup:status>SUCCESS</cup:status>
    </cup:unsubscribeResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Get Subscribed Presence Request

The getSubscribedPresence request returns presence information for a subscription. This method is called following a notify message from Cisco Unified CM IM and Presence. The Presence Web Service supports two types of presence, basic and rich presence. For a request for rich presence data, a PIDF document is passed in the response.

Input Parameters

Parameter	Description
subscriptionID	The ID number of the subscription.
presenceType	The type of presence being requested. The valid values are BASIC_PRESENCE or RICH_PRESENCE. The values are case sensitive.

Output Parameters

Parameter	Description
presenceType	The type of presence being requested. The valid values are BASIC_PRESENCE or RICH_PRESENCE. The values are case sensitive.
basicPresenceList	A list of contact URIs and their associated presence states.
richPresenceList	A PIDF document describing rich presence for a contact. There is a PIDF document sent for each end user.

Example 68 shows a getSubscribedPresence request for basic presence.

Example 68 getSubscribedPresence request for basic presence

```
<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
  <soapenv:Header>
    <session-key xmlns="urn:cisco:cup:presence:soap">
      {end-user-session-key}
    </session-key>
  </soapenv:Header>
  <soapenv:Body>
    <getSubscribedPresence xmlns="urn:cisco:cup:presence:soap">
      <subscriptionID>1</subscriptionID>
      <presenceType>BASIC_PRESENCE</presenceType>
    </getSubscribedPresence>
  </soapenv:Body>
</soapenv:Envelope>
```

Example 69 shows a getSubscribedPresence response for basic presence.

Example 69 getSubscribedPresence response for basic presence

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope" xmlns:SOAP-
ENC="http://www.w3.org/2003/05/soap-encoding"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:cup="urn:cisco:cup:presence:soap">
```

```
  <SOAP-ENV:Header></SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <cup:getSubscribedPresenceResponse>
      <cup:presenceResponse>
        <cup:presenceType>BASIC_PRESENCE</cup:presenceType>
        <cup:basicPresenceList>
          <cup:contact presenceStatus="AWAY" contactURI="enduser1@cisco.com"></cup:contact>
          <cup:contact presenceStatus="AVAILABLE"
contactURI="enduser2@cisco.com"></cup:contact>
          <cup:contact presenceStatus="UNAVAILABLE"
contactURI="enduser3@cisco.com"></cup:contact>
        </cup:basicPresenceList>
        <richPresenceList xmlns="urn:cisco:cup:presence:soap">
        </richPresenceList>
      </cup:presenceResponse>
    </cup:getSubscribedPresenceResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example 70 shows a getSubscribedPresence response for rich presence.

Example 70 getSubscribedPresence response for rich presence

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope" xmlns:SOAP-
ENC="http://www.w3.org/2003/05/soap-encoding"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:cup="urn:cisco:cup:presence:soap">
  <SOAP-ENV:Header></SOAP-ENV:Header>
  <soapenv:Body>
    <getSubscribedPresenceResponse xmlns="urn:cisco:cup:presence:soap">
      <PresenceResponse>
        <presenceType>RICH_PRESENCE</presenceType>
        <richPresenceList>
          {multiple PIDF documents}
        </richPresenceList>
                    <basicPresence></basicPresence>
      </PresenceResponse>
    </getSubscribedPresenceResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Get Polled Presence Request

The getPolledPresence request returns presence information for a list of users. The Presence Web Service supports two types of presence supported, basic and rich presence. For a request for rich presence data, a PIDF document is passed in the response.

Input Parameters

Parameter	Description
contactsList	The list of contact URIs that are part of the polled presence request.
presenceType	The type of presence being requested. The valid values are BASIC_PRESENCE or RICH_PRESENCE. The values are case sensitive.

Output Parameters

Parameter	Description
presenceType	The type of presence being requested. The valid values are BASIC_PRESENCE or RICH_PRESENCE. The values are case sensitive.
basicPresenceList	A list of contact URIs and their associated presence states.
richPresenceList	A PIDF document describing rich presence for a contact. There is a PIDF document sent for each end user.

Example 71 shows a getPolledPresence request for basic presence.

Example 71 getPolledPresence request for basic presence

```
<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
  <soapenv:Header>
    <session-key xmlns="urn:cisco:cup:presence:soap">
      {end-user-session-key}
    </session-key>
  </soapenv:Header>
  <soapenv:Body>
    <getPolledPresence xmlns="urn:cisco:cup:presence:soap">
      <contactsList>
        <contact contactURI="enduser1@cisco.com" />
        <contact contactURI="enduser2@cisco.com" />
        <contact contactURI="enduser3@cisco.com" />
      </contactsList>
      <presenceType>BASIC_PRESENCE</presenceType>
    </getPolledPresence>
  </soapenv:Body>
</soapenv:Envelope>
```

Example 72 shows a getPolledPresence response for basic presence.

Example 72 getPolledPresence response for basic presence

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope" xmlns:SOAP-
ENC="http://www.w3.org/2003/05/soap-encoding"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:cup="urn:cisco:cup:presence:soap">
  <soapenv:Body>
    <getPolledPresenceResponse xmlns="urn:cisco:cup:presence:soap">
      <PresenceResponse>
        <presenceType>BASIC_PRESENCE</presenceType>
        <basicPresenceList>
          <contact contactURI="enduser1@cisco.com" presenceStatus="BUSY"/>
          <contact contactURI="enduser2@cisco.com" presenceStatus="DND"/>
          <contact contactURI="enduser3@cisco.com" presenceStatus="UNKNOWN"/>
        </basicPresenceList>
        <richPresenceList></richPresenceList>
      </PresenceResponse>
    </getPolledPresenceResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Example 73 shows a getPolledPresence response for rich presence.

Example 73 getPolledPresence response for rich presence

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope" xmlns:SOAP-
ENC="http://www.w3.org/2003/05/soap-encoding"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:cup="urn:cisco:cup:presence:soap">
  <SOAP-ENV:Header></SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <cup:getPolledPresenceResponse>
      <cup:presenceResponse>
        <cup:presenceType>RICH_PRESENCE</cup:presenceType>
        <richPresenceList xmlns="urn:cisco:cup:presence:soap">
          {multiple PIDF documents}
        </richPresenceList>
      </cup:presenceResponse>
    </cup:getPolledPresenceResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Set Presence Request

The setPresence request allows users to set their own presence states. Users are identified to Cisco Unified CM IM and Presence by the session key.

Input Parameters

Parameter	Description
presenceType	The type of presence being set. The valid values are BASIC_PRESENCE or

	RICH_PRESENCE. The values are case sensitive.
basicPresence	Used to send basic presence data by passing the presence state in this string. The permitted values are AVAILABLE, BUSY, DND, AWAY, UNAVAILABLE and VACATION. The values are case sensitive.
richPresence	Used to send rich presence data by passing a PIDF document in the message.
override	This flag is used to override the presence state of all end user devices. The valid values are "true" and "false".
expiration	The time (in seconds) that the presence state is valid for. The minimum value is 3600 seconds (1 hour). The maximum value is 86400 seconds (24 hours).

The *basicPresence* value AVAILABLE must not be set as an override state (override flag must be set to false), and the *basicPresence* value of DND can only be set as an override state (override flag must be set to true). The same rules apply when setting rich presence.

Example 74 shows a setPresence request for basic presence with the presence state set to BUSY.

Example 74 setPresence request for basic presence

```
<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
  <soapenv:Header>
    <session-key xmlns="urn:cisco:cup:presence:soap">
      {end-user-session-key}
    </session-key>
  </soapenv:Header>
  <soapenv:Body>
    <setPresence xmlns="urn:cisco:cup:presence:soap">
      <presenceType>BASIC_PRESENCE</presenceType>
      <presenceInfo>
        <basicPresence>BUSY</basicPresence>
        <override>false</override>
      </presenceInfo>
      <expiration>3600</expiration>
    </setPresence>
  </soapenv:Body>
</soapenv:Envelope>
```

The user can send rich presence data by passing a PIDF document in the setPresence request, as shown in Example 75.

Example 75 setPresence request for rich presence

```
<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
  <soapenv:Header>
    <session-key xmlns="urn:cisco:cup:presence:soap">
      {end-user-session-key}
```

```
    </session-key>
  </soapenv:Header>
  <soapenv:Body>
    <setPresence xmlns="urn:cisco:cup:presence:soap">
      <presenceType>RICH_PRESENCE</presenceType>
      <presenceInfo>
        <richPresence>
          {a PIDF document}
        </richPresence>
        <override>false</override>
      </presenceInfo>
      <expiration>3600</expiration>
    </setPresence>
  </soapenv:Body>
</soapenv:Envelope>
```

Example 76 shows a setPresence response.

Example 76 setPresence response

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope" xmlns:SOAP-
ENC="http://www.w3.org/2003/05/soap-encoding"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:cup="urn:cisco:cup:presence:soap">
  <SOAP-ENV:Header></SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <cup:setPresenceResponse>
      <cup:status>SUCCESS</cup:status>
    </cup:setPresenceResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

About the REST Interface Resources

All REST requests for the Presence Web Service must be sent to the following URL:
http://<cup_host>:<cup_port>/presence-service

URI Reference Table

Table 3 provides a reference guide for the URIs in the REST interface. Note that these URIs should be prefixed by " *https:///<cup-server>/presence-service/* " to be complete.

Table 3 REST URI Reference Table

Action	Method	URI
(Sessions)		

Login user	POST	/users/{user}/sessions
Logout user	DELETE	/users/{user}/sessions
(Endpoints)		
Register an endpoint	POST	/endpoints
Unregister an endpoint	DELETE	/endpoints/{endpointId}
(Subscriptions)		
Create a subscription	POST	/users/{user}/subscriptions
Add to a subscription	POST	/users/{user}/subscriptions/{subId}
Remove from a subscription	PUT	/users/{user}/subscriptions/{subId}
Remove a subscription	DELETE	/users/{user}/subscriptions/{subId}
(Presence)		
Subscribed Presence	GET	/users/{user}/presence/{type}/subscriptions/{subId}
Polled Presence	GET	/users/{user}/presence/{type}/contacts/{list}
Set Presence	PUT	/users/{user}/presence/{type}

Login Request (Application User)

This request logs in an application user to Cisco Unified CM IM and Presence.

Input Parameters

Parameter	Description
username	The username of the application user.
password	The password of the application user.

Output Parameters

Parameter	Description
sessionKey	The session key of the application user.
backupServer	The backup (failover) node of application user. This parameter is only present in the response if ther e is a backup server available.

For failover purposes an end user is configured with a primary server and a backup server on Cisco Unified CM IM and Presence. On a successful login, in addition to a session key, the end user is returned the address of the backup server. If the primary node becomes unavailable, the end user can log in to the backup server.

The "username" for the end user is passed as a parameter in the URL of REST request messages.

The parameter "Presence-Session-Key", which holds the session ID of the user, is passed as a HTTP header in all subsequent REST request messages; The session ID is used as an authorization mechanism for each request.

Example 77 shows a forced login request for an application user.

Example 77 Forced Login request (application user)

```
PUT /presence-service/users/{app-userid}/session HTTP/1.1

<?xml version="1.0"?>
<session>
   <password>{app-user-password}</password>
</session>
```

To send a non-forced login request, send a HTTP POST request to /presence-service/users/username/sessions. Note the method type used is POST, and the "sessions" string is plural, as shown in Example 78.

Example 78 Non-Forced Login request (application user)

```
POST /presence-service/users/{app-userid}/sessions HTTP/1.1

<?xml version="1.0"?>
<session>
   <password>{app-user-password}</password>
</session>
```

Example 79 shows a login response for an application user.

Example 79 Login response (application user)

```
HTTP/1.1 201 Created
Location: https://{cup-hostname:port}/presence-service/users/{app-userid}/session
Content-Type: text/xml

<?xml version="1.0" encoding="UTF-8"?>
<session>
  <sessionKey>{app-user-session-key}</sessionKey>
</session>
```

Login Request (Application User Logging in End User)

This request is used for an application user logging in an end user to Cisco Unified CM IM and Presence. The session key of the application user is passed in the request.

Input Parameters

Parameter	Description
username	The username of the end user.
Presence-Session-Key	The session key of the application user.

Output Parameters

Parameter	Description
sessionKey	The session key of the end user.
backupServer	The backup (failover) node of the end user. This parameter is only present in the response if there is a backup server available.
primaryServer	The primary node of the application user. This parameter is part of the redirect response only.

A forced login request is shown in Example 80.

Example 80 Forced Login request (application user logging in end user)

```
PUT /presence-service/users/{end-userid}/session HTTP/1.1
Presence-Session-Key: {app-user-session-key}
```

To send a non-forced login request, send a HTTP POST request to /presence-service/users/username/sessions. Note the method type used is POST, and the "sessions" string is plural, as shown below.

Example 81 Non-Forced Login request (application user logging in end user)

```
POST /presence-service/users/{end-userid}/sessions HTTP/1.1
Presence-Session-Key: {app-user-session-key}
```

A login response is shown in Example 82.

Example 82 Login response (application user logging in end user)

```
HTTP/1.1 201 Created
Location: https://{cup_hostname:port}/presence-service/users/{end-userid}/session
Content-Type: text/xml

<?xml version="1.0" encoding="UTF-8"?>
<session>
  <sessionKey>{end-user-session-key}</sessionKey>
    <backupServer>{end-user-backup-server}</backupServer>
</session>
```

A redirect login response is issued if a users attempt to log in to a Cisco Unified CM IM and Presence node where they are not registered as users. The redirect login response contains information on the primary server and backup server assigned to the user. The application can log in the end user to the primary server specified in the redirect login response.

The redirect login response contains information on the backup server assigned to the user.

Example 83 shows a redirect login response.

Example 83 Redirect login response

```
HTTP/1.1 307 Temporary Redirect
Location: https://{cup_hostname:port}/presence-service/users/{end-userid}/session
Content-Type: text/xml

<?xml version="1.0" encoding="UTF-8"?>
<session>
   <redirect primaryServer="{cup-primary-server}" backupServer="{cup-backup-server}"/>
</session>
```

Logout Request

This request logs out an application user or an end user.

Input Parameters

Parameter	Description
username	The username of the user.
Presence-Session-Key	The session key of the user.

Example 84 shows a logout request.

Example 84 Logout request

```
DELETE /presence-service/users/{userid}/session HTTP/1.1
Presence-Session-Key: {session-key}
```

Example 85 shows a logout response.

Example 85 Logout response

```
HTTP/1.1 200 OK
```

The logout response does not contain a message body.

Register Endpoint Request

The registerEndPoint request registers a new HTTP endpoint to be used for presence notifications. This method is only used by an application user. The session key in the request must be associated with a logged-in application user.

Input Parameters

Parameter	Description
Presence-Session-Key	The session key of the application user.
Presence-Expiry	The time (in seconds) that the registration is valid for. This parameter is passed as a HTTP header in the request. The maximum value is 86400 seconds (24 hours). The minimum value is 3600 seconds (1 hour).
callback-url	The URL for the HTTP endpoint.

Output Parameters

Parameter	Description
endpointId	The registration ID for the HTTP endpoint. This is passed as a parameter when refreshing the endpoint registration.

Example 86 shows a registerEndPoint request.

Example 86 RegisterEndPoint request

```
POST /presence-service/endpoints HTTP/1.1
Presence-Session-Key: {app-user-session-key}
Presence-Expiry: 3600

<?xml version="1.0"?>
<endpoint>
   <url>{callback-url}</url>
</endpoint>
```

Example 87 shows a registerEndPoint response.

Example 87 RegisterEndPoint response

```
HTTP/1.1 201 Created
Location: https://{cup-hostname:port}/presence-service/endpoints/{endpoint-id}
Content-Type: text/plain; charset=ISO-8859-1
```

To renew an endpoint registration, send a HTTP PUT request to *{hostname}/presence-service/endpoints/{endpointId}* , specifying the existing endpoint ID in the request URI and the

updated expiration value in the HTTP header. The request should not contain an XML body. The response code for this request message is a HTTP 200 OK response.

Unregister Endpoint Request

The unregisterEndPoint request unregisters a HTTP endpoint that is used for presence notifications. This method is only used by an application user. The session key in the request must be associated with a logged-in application user. The session key used in the unregister endpoint request must be the same session key used in the original register request.

Input Parameters

Parameter	Description
Presence-Session-Key	The session key of the application user.
endpoint Id	The registration ID for the HTTP endpoint. In this case it is the endpoint that the client wishes to unregister.

Example 88 shows an unregisterEndPoint request.

Example 88 UnregisterEndPoint request

```
DELETE /presence-service/endpoints/{endpoint-id} HTTP/1.1
Presence-Session-Key: {app-user-session-key}
```

Example 89 shows an unsubscribe response.

Example 89 UnregisterEndPoint response

```
HTTP/1.1 200 OK
```

The unregisterEndPoint response does not contain a message body.

Subscribe Request

The subscribe request sets up a subscription with Cisco Unified CM IM and Presence for event notifications. The notifications are sent to the HTTP endpoint registered for the client application.

Input Parameters

Parameter	Description
username	The username of the end user.

Presence-Session-Key	The session key of the end user.
Presence-Expiry	The time (in seconds) for which the subscription is valid. This parameter is passed in the HTTP header of the request. The maximum value is 86400 seconds (24 hours). A non-zero value must be greater than or equal to 3600 seconds (1 hour). If you are adding contacts to an existing subscription this is an optional parameter.
subscriptionType	The type of event that is being subscribed to. Currently the Presence Web Service only supports the value PRESENCE_NOTIFICATION. The value is case sensitive.
endpoint ID	The registration ID for the HTTP endpoint.
subID	The ID number of the subscription. This parameter is only used for updating a subscription.
contactsList	The list of contact URIs that are part of the subscription.

Output Parameters

Parameter	Description
username	The username of the end user.
subID	The ID number of the subscription.

Example 90 shows a subscribe request.

Example 90 Subscribe request

```
POST /presence-service/users/{end-userid}/subscriptions HTTP/1.1
Presence-Session-Key: {end-user-session-key}
Presence-Expiry: 3600
Content-Type: text/xml; charset=UTF-8

<?xml version="1.0"?>
<subscription>
<contactsList>
   <contact contactURI="enduser1@cisco.com"/>
   <contact contactURI="enduser2@cisco.com"/>
   <contact contactURI="enduser3@cisco.com"/>
</contactsList>
<subscriptionType>PRESENCE_NOTIFICATION</subscriptionType>
<endPointID>{endpoint-id}</endPointID>
</subscription>
```

Example 91 shows a subscribe response.

Example 91 Subscribe response

```
HTTP/1.1 201 Created
Location: https://{cup-hostname:port}/presence-service/users/{end-userid}/subscriptions/{sub-id}
Content-Type: text/plain; charset=ISO-8859-1
```

To add new contacts to an existing subscription, send a HTTP POST request to */presence-service/users/{username}/subscriptions/{subId}* and pass the new contact URIs and the endpoint ID in the request body. The response code for this request message is a HTTP 200 OK response.

To refresh the expiration time of a subscription, send a HTTP PUT request to */presence-service/users/{username}/subscriptions/{subId}* and pass the updated expiration value as a HTTP header and the endpoint ID in the request body. The response code for this request message is a HTTP 200 OK response.

Unsubscribe Request

The unsubscribe request terminates the subscription of the client application with Cisco Unified CM IM and Presence for event notifications. The subscription can be terminated for specified list of contacts, or for all contacts. The session key used in the unsubscribe request must be the same session key used in the original subscribe request.

Input Parameters

Parameter	Description
username	The username of the end user.
Presence-Session-Key	The session key of the end user.
subID	The ID number of the subscription. In this case it is the subscription that the client wishes to unsubscribe from.
contactsList	The list of contact URIs that you wish to remove from the subscription.

Example 92 shows an unsubscribe request for all contacts.

Example 92 Unsubscribe request (all contacts)

```
DELETE/presence-service/users/{end-userid}/subscriptions/{sub-id} HTTP/1.1
Presence-Session-Key: {end-user-session-key}
```

To remove a subset of contacts from an existing subscription, send a HTTP PUT request to the subscription URI i.e. */presence-service/users/{username}/subscriptions/{subId},* and pass the contacts to be removed from the subscription in the request body.

Example 93 shows an unsubscribe request for a sub-set of contacts.

Example 93 Unsubscribe request (sub-set of contacts)

```
PUT /presence-service/users/{end-userid}/subscriptions/{sub-id} HTTP/1.1
Presence-Session-Key: {end-user-session-key}
Content-Type: text/xml; charset=UTF-8

<?xml version="1.0"?>
<subscription>
   <contactsList>
      <contact contactURI="enduser1@cisco.com"/>
      <contact contactURI="enduser3@cisco.com"/>
   </contactsList>
</subscription>
```

Example 94 shows an unsubscribe response.

Example 94 Unsubscribe response

```
HTTP/1.1 200 OK
```

Get Subscribed Presence Request

The getSubscribedPresence request returns presence information for a subscription. This method is called following a notify message from Cisco Unified CM IM and Presence. The Presence Web Service supports two types of presence, basic and rich presence. For a request for rich presence data, a PIDF document is passed in the response.

Input Parameters

Parameter	Description
username	The username of the end user.
Presence-Session-Key	The session key of the end user.
type	The type of presence being requested. The valid values are "basic" or "rich".
subID	The ID number of the subscription.

Output Parameters

Parameter	Description
type	The type of presence being returned. The valid values are "basic" or "rich". Basic presence consists of a contact URI and the presenceStatus. Rich presence consists of a PIDF document.
contactURI	The URI of the end user (basic).

presenceStatus	The basic presence state of the end user (basic).
multiple PIDF documents	A PIDF document containing the rich presence of the end user. There is a PIDF document sent for each end user.

Example 95 shows a getSubscribedPresence request for basic presence.

Example 95 getSubscribedPresence request

```
GET /presence-service/users/{end-userid}/presence/basic/subscriptions/{sub-id} HTTP/1.1
Presence-Session-Key: {end-user-session-key}
```

The getSubscribedPresence responses are the same as the responses for getPolledPresence.

Get Polled Presence Request

The getPolledPresence request returns presence information for a list of users. There are two types of presence supported, basic and rich presence. For a request for rich presence data, a PIDF document is passed in the response.

Input Parameters

Parameter	Description
username	The username of the end user.
Presence-Session-Key	The session key of the end user.
type	The type of presence being requested. The valid values are "basic" or "rich".
contactsList	A semi-colon separated list of contact URIs e.g. bob@cisco.com;tom@cisco.com;anne@cisco.com.

Output Parameters

Parameter	Description
type	The type of presence being returned. The valid values are "basic" or "rich". Basic presence consists of a contact URI and the presenceStatus. Rich presence consists of a PIDF document.
contactURI	The URI of the end user. (basic)
presenceStatus	The basic presence state of the end user. (basic)
multiple PIDF	A PIDF document containing the rich presence of the end user. There is a

documents	PIDF document sent for each end user.

Example 96 shows a getPolledPresence request.

Example 96 getPolledPresence request

```
GET /presence-service/users/{end-userid}/presence/{type}/contacts/{contact-list} HTTP/1.1
Presence-Session-Key: {end-user-session-key}
```

Example 97 shows a getPolledPresence response for basic presence.

Example 97 getPolledPresence response for basic presence

```
HTTP/1.1 200 OK
Content-Type: text/xml

<?xml version="1.0" encoding="UTF-8"?>
<presenceList type="basic">
   <contact contactURI="enduser1@cisco.com" presenceStatus="AVAILABLE"/>
   <contact contactURI="enduser2@cisco.com" presenceStatus="AWAY"/>
   .. .. ..
</presenceList>
```

Example 98 shows a getPolledPresence response for rich presence.

Example 98 getPolledPresence response for rich presence

```
HTTP/1.1 200 OK
Content-Type: text/xml

<?xml version="1.0" encoding="UTF-8"?>
<presenceList type="rich">
   {multiple PIDF documents}
</presenceList>
```

Set Presence Request

The setPresence request allows users to set their own presence states. Users are identified to Cisco Unified CM IM and Presence by the session key.

Input Parameters

Parameter	Description
username	The username of the end user.
Presence-	The session key of the end user.

Session-Key	
type	The type of presence being requested. As illustrated in the examples below, the valid values are "basic" or "rich", for example, */presence/basic* or */presence/rich* .
Presence-Override	This flag is used to override the presence state of all end user devices. This value is passed as a HTTP header in the request. The valid values are "true" and "false".
Presence-Expiry	The time (in seconds) that the presence state is valid for. This value is passed as a HTTP header in the request. The minimum value is 3600 seconds (1 hour). The maximum value is 86400 seconds (24 hours).
status	Used to send basic presence data by passing the presence state in this string. The permitted values are AVAILABLE, BUSY, DND, AWAY, UNAVAILABLE and VACATION. The values are case sensitive.

The *presence-status* value AVAILABLE must not be set as an override state (override flag must be set to false), and the *presence-status* value of DND can only be set as an override state (override flag must be set to true). The same rules apply when setting rich presence.

Example 99 shows a setPresence request (basic presence).

Example 99 setPresence request (basic presence)

```
PUT /presence-service/users/{end-userid}/presence/basic HTTP/1.1
Presence-Expiry: {expiration}
Presence-Override: {override}
Presence-Session-Key: {end-user-session-key}

<?xml version="1.0"?>
<presence>{status}</presence>
```

Example 100 shows a setPresence request (rich presence). To send rich presence data, a PIDF document is passed in message body.

Example 100 setPresence request (rich presence)

```
PUT /presence-service/users/{end-userid}/presence/rich HTTP/1.1
Presence-Expiry: {expiration}
Presence-Override: {override}
Presence-Session-Key: {end-user-session-key}
Content-Type: text/xml; charset=UTF-8

{PIDF document}
```

Example 101 shows a setPresence response.

Example 101 setPresence response

```
HTTP/1.1 200 OK
```

The setPresence response does not contain a message body.

Presence API WSDL

The WSDL specification provides the basis for the Web Service Definition Language (WSDL) for the Presence Web Service. You can access the WSDL for the Presence Web Service on a Cisco Unified CM IM and Presence server at:

http://<cuphost>:<port>/presence-service/soap?wsdl

Use this WSDL, and the interface methods that are described in this chapter, to develop customized applications for the Presence Web Service.

Sample PIDF

A number of sample PIDF documents are provided below to show how to set different presence states over the PWS interface. The example provided show how to set both device and manual presence.

Device status is a specific status of an individual IM device belonging to a user. If a user has multiple IM devices, CUCM IM & Presence will compose an overall user status that best represents a user's status across all such devices. When setting device presence over the PWS interface, the appropriate override flag on the SOAP/REST request MUST be set to false:

- For SOAP, the element within the body of the request is called override
- For REST, the HTTP Header is called Presence-Override

Manual Presence data on the other hand is a presence update that is explicitly set by the user. Such manual presence data generally override any system-determined presence and become overall user status. When setting manual presence over the PWS interface, the appropriate override flag on the SOAP/REST request MUST be set to true:

- For SOAP, the element within the body of the request is called override
- For REST, the HTTP Header is called Presence-Override

The tuple-id value must be "pws-override" when setting manual rich presence through the Presence Web Service.

The tuple-id value must be "pws-persistent" when setting manual persistent rich presence, such as 'vacation', through the Presence Web Service.

The tuple-source value must be “Presence Web Service” when setting manual or device rich presence. The exception to this is when setting manual Do Not Disturb (DND) rich presence. When setting manual Do Not Disturb (DND) rich presence the tuple source value must be “Human”.

Setting Available (Device Presence)

This sample PIDF has a presence setting of device “Available”

```
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
xmlns:rp="urn:ietf:params:xml:ns:pidf:rpid"
xmlns:dm="urn:ietf:params:xml:ns:pidf:data-model"
xmlns:pp="urn:ietf:params:xml:ns:pidf:person"
xmlns:et="urn:ietf:params:xml:ns:pidf:rpid:rpid-tuple"
xmlns:ep="urn:ietf:params:xml:ns:pidf:rpid:rpid-person"
xmlns:es="urn:ietf:params:xml:ns:pidf:rpid:status:rpid-status"
xmlns:ci="urn:ietf:params:xml:ns:pidf:cipid"
xmlns:ce="urn:cisco:params:xml:ns:pidf:rpid"
xmlns:sc="urn:ietf:params:xml:ns:pidf:servcaps"
xmlns:so="urn:cisco:params:xml:ns:pidf:source"
entity="sip:testuser1@cisco.com">
   <dm:person id="testuser1@cisco.com">
      <rp:activities>
         <ce:available/>
      </rp:activities>
   </dm:person>
   <tuple id="cisco-pws">
      <contact priority="0.5">sip:testuser1@cisco.com</contact>
      <so:source>Presence Web Service</so:source>
      <sc:servcaps>
         <sc:text>true</sc:text>
         <sc:type>text/plain; charset=UTF-8</sc:type>
      </sc:servcaps>
      <user-input>active</user-input>
      <status>
         <basic>open</basic>
      </status>
   </tuple>
</presence>
```

Setting Away/Idle (Device Presence)

This sample PIDF has a presence setting of device “Away” (idle)

```
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
xmlns:rp="urn:ietf:params:xml:ns:pidf:rpid"
xmlns:dm="urn:ietf:params:xml:ns:pidf:data-model"
xmlns:pp="urn:ietf:params:xml:ns:pidf:person"
xmlns:et="urn:ietf:params:xml:ns:pidf:rpid:rpid-tuple"
xmlns:ep="urn:ietf:params:xml:ns:pidf:rpid:rpid-person"
xmlns:es="urn:ietf:params:xml:ns:pidf:rpid:status:rpid-status"
xmlns:ci="urn:ietf:params:xml:ns:pidf:cipid"
xmlns:ce="urn:cisco:params:xml:ns:pidf:rpid"
xmlns:sc="urn:ietf:params:xml:ns:pidf:servcaps"
xmlns:so="urn:cisco:params:xml:ns:pidf:source"
entity="sip:testuser1@cisco.com">
   <dm:person id="testuser1@cisco.com">
```

```
      <rp:activities>
         <ce:away/>
      </rp:activities>
   </dm:person>
   <tuple id="cisco-pws">
      <contact priority="0.5">sip:testuser1@cisco.com</contact>
      <so:source>Presence Web Service</so:source>
      <sc:servcaps>
         <sc:text>true</sc:text>
         <sc:type>text/plain; charset=UTF-8</sc:type>
      </sc:servcaps>
      <user-input>active</user-input>
      <status>
         <basic>open</basic>
      </status>
   </tuple>
</presence>
```

Setting Phone Status (Device Presence)

This sample PIDF has a presence setting of on the phone

```
<?xml version="1.0" encoding="UTF-8" ?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
xmlns:rp="urn:ietf:params:xml:ns:pidf:rpid"
xmlns:dm="urn:ietf:params:xml:ns:pidf:data-model"
xmlns:pp="urn:ietf:params:xml:ns:pidf:person"
xmlns:et="urn:ietf:params:xml:ns:pidf:rpid:rpid-tuple"
xmlns:ep="urn:ietf:params:xml:ns:pidf:rpid:rpid-person"
xmlns:es="urn:ietf:params:xml:ns:pidf:rpid:status:rpid-status"
xmlns:ci="urn:ietf:params:xml:ns:pidf:cipid"
xmlns:ce="urn:cisco:params:xml:ns:pidf:rpid"
xmlns:sc="urn:ietf:params:xml:ns:pidf:servcaps"
xmlns:so="urn:cisco:params:xml:ns:pidf:source"
entity="sip:testuser1@cisco.com">
   <dm:person xmlns="urn:cisco:params:xml:ns:pidf:rpid" id="testuser1">
      <rp:activities>
         <ce:busy/>
         <ce:on-the-phone/>
      </rp:activities>
   </dm:person>
   <tuple id="cisco-pws">
      <contact priority="0.5">sip:testuser1@cisco.com</contact>
      <so:source>Presence Web Service</so:source>
      <sc:servcaps xmlns="urn:ietf:params:xml:ns:pidf:servcaps">
         <sc:text>true</sc:text>
         <sc:type>text/plain; charset=UTF-8</sc:type>
      </sc:servcaps>
      <user-input>active</user-input>
      <status>
         <basic>open</basic>
      </status>
      <activities>
```

```
        <busy/>
        <on-the-phone/>
      </activities>
   </tuple>
</presence>
```

Setting Meeting Status (Device Presence)

This sample PIDF has a presence setting of 'In A Meeting'

```
<?xml version="1.0" encoding="UTF-8" ?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
xmlns:rp="urn:ietf:params:xml:ns:pidf:rpid"
xmlns:dm="urn:ietf:params:xml:ns:pidf:data-model"
xmlns:pp="urn:ietf:params:xml:ns:pidf:person"
xmlns:et="urn:ietf:params:xml:ns:pidf:rpid:rpid-tuple"
xmlns:ep="urn:ietf:params:xml:ns:pidf:rpid:rpid-person"
xmlns:es="urn:ietf:params:xml:ns:pidf:rpid:status:rpid-status"
xmlns:ci="urn:ietf:params:xml:ns:pidf:cipid"
xmlns:ce="urn:cisco:params:xml:ns:pidf:rpid"
xmlns:sc="urn:ietf:params:xml:ns:pidf:servcaps"
xmlns:so="urn:cisco:params:xml:ns:pidf:source"
entity="sip:testuser1@cisco.com">
   <dm:person xmlns="urn:cisco:params:xml:ns:pidf:rpid" id="testuser1">
      <rp:activities>
         <ce:busy/>
         <ce:meeting/>
      </rp:activities>
   </dm:person>
   <tuple id="cisco-pws">
      <contact priority="0.5">sip:testuser1@cisco.com</contact>
      <so:source>Presence Web Service</so:source>
      <sc:servcaps xmlns="urn:ietf:params:xml:ns:pidf:servcaps">
         <sc:text>true</sc:text>
         <sc:type>text/plain; charset=UTF-8</sc:type>
      </sc:servcaps>
      <user-input>active</user-input>
      <status>
         <basic>open</basic>
      </status>
   </tuple>
</presence>
```

Setting Offline (Device Presence)

This sample PIDF has a presence setting of offline or unavailable

```
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
xmlns:rp="urn:ietf:params:xml:ns:pidf:rpid"
xmlns:dm="urn:ietf:params:xml:ns:pidf:data-model"
xmlns:pp="urn:ietf:params:xml:ns:pidf:person"
xmlns:et="urn:ietf:params:xml:ns:pidf:rpid:rpid-tuple"
```

```
xmlns:ep="urn:ietf:params:xml:ns:pidf:rpid:rpid-person"
xmlns:es="urn:ietf:params:xml:ns:pidf:rpid:status:rpid-status"
xmlns:ci="urn:ietf:params:xml:ns:pidf:cipid"
xmlns:ce="urn:cisco:params:xml:ns:pidf:rpid"
xmlns:sc="urn:ietf:params:xml:ns:pidf:servcaps"
xmlns:so="urn:cisco:params:xml:ns:pidf:source"
entity="sip:testuser1@cisco.com">
   <tuple id="cisco-pws">
      <contact priority="0.5">sip:testuser1@cisco.com</contact>
      <so:source>Presence Web Service</so:source>
      <sc:servcaps>
         <sc:text>false</sc:text>
         <sc:type>text/plain; charset=UTF-8</sc:type>
      </sc:servcaps>
      <status>
         <basic>closed</basic>
      </status>
   </tuple>
</presence>
```

Setting Manual Away

This sample PIDF has a presence setting of manual Away

```
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
xmlns:rp="urn:ietf:params:xml:ns:pidf:rpid"
xmlns:dm="urn:ietf:params:xml:ns:pidf:data-model"
xmlns:pp="urn:ietf:params:xml:ns:pidf:person"
xmlns:et="urn:ietf:params:xml:ns:pidf:rpid:rpid-tuple"
xmlns:ep="urn:ietf:params:xml:ns:pidf:rpid:rpid-person"
xmlns:es="urn:ietf:params:xml:ns:pidf:rpid:status:rpid-status"
xmlns:ci="urn:ietf:params:xml:ns:pidf:cipid"
xmlns:ce="urn:cisco:params:xml:ns:pidf:rpid"
xmlns:sc="urn:ietf:params:xml:ns:pidf:servcaps"
xmlns:so="urn:cisco:params:xml:ns:pidf:source"
entity="sip:testuser1@cisco.com">
   <dm:person id="testuser1@cisco.com">
      <rp:activities>
         <ce:away/>
      </rp:activities>
      <class>manual</class>
   </dm:person>
   <tuple id="pws-override">
      <contact priority="0.5">sip:testuser1@cisco.com</contact>
      <so:source>Presence Web Service</so:source>
      <sc:servcaps>
         <sc:text>true</sc:text>
         <sc:type>text/plain; charset=UTF-8</sc:type>
      </sc:servcaps>
      <status>
         <basic>open</basic>
      </status>
   </tuple>
```

```
</presence>
```

Setting Manual Do Not Disturb (DND)

This sample PIDF has a presence setting of manual Do Not Disturb (DND)

```
<?xml version="1.0" encoding="UTF-8" ?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
xmlns:rp="urn:ietf:params:xml:ns:pidf:rpid"
xmlns:dm="urn:ietf:params:xml:ns:pidf:data-model"
xmlns:pp="urn:ietf:params:xml:ns:pidf:person"
xmlns:et="urn:ietf:params:xml:ns:pidf:rpid:rpid-tuple"
xmlns:ep="urn:ietf:params:xml:ns:pidf:rpid:rpid-person"
xmlns:es="urn:ietf:params:xml:ns:pidf:rpid:status:rpid-status"
xmlns:ci="urn:ietf:params:xml:ns:pidf:cipid"
xmlns:ce="urn:cisco:params:xml:ns:pidf:rpid"
xmlns:sc="urn:ietf:params:xml:ns:pidf:servcaps"
xmlns:so="urn:cisco:params:xml:ns:pidf:source"
entity="sip:testuser1@cisco.com">
   <dm:person xmlns="urn:cisco:params:xml:ns:pidf:rpid" id="testuser1">
      <rp:activities>
         <ce:dnd/>
      </rp:activities>
      <class>manual</class>
   </dm:person>
   <tuple id="pws-override">
      <so:source>Human</so:source>
      <sc:servcaps>
         <sc:text>true</sc:text>
         <sc:type>text/plain; charset=UTF-8</sc:type>
      </sc:servcaps>
      <status>
         <basic>open</basic>
      </status>
   </tuple>
</presence>
```

With any manual status, it is possible to provide a custom text annotation. This annotation will be incorporated into the user's overall status.

Setting Custom Manual Away

This sample PIDF has a presence setting of manual Away with a custom text annotation of 'Gone to Lunch'

```
<?xml version="1.0" encoding="UTF-8" ?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
xmlns:rp="urn:ietf:params:xml:ns:pidf:rpid"
xmlns:dm="urn:ietf:params:xml:ns:pidf:data-model"
xmlns:pp="urn:ietf:params:xml:ns:pidf:person"
xmlns:et="urn:ietf:params:xml:ns:pidf:rpid:rpid-tuple"
```

```
xmlns:ep="urn:ietf:params:xml:ns:pidf:rpid:rpid-person"
xmlns:es="urn:ietf:params:xml:ns:pidf:rpid:status:rpid-status"
xmlns:ci="urn:ietf:params:xml:ns:pidf:cipid"
xmlns:ce="urn:cisco:params:xml:ns:pidf:rpid"
xmlns:sc="urn:ietf:params:xml:ns:pidf:servcaps"
xmlns:so="urn:cisco:params:xml:ns:pidf:source"
entity="sip:testuser1@cisco.com">
   <dm:person id="testuser1@cisco.com">
      <rp:activities>
         <ce:away/>
      </rp:activities>
      <class>manual</class>
      <note>Gone To Lunch</note>
   </dm:person>
   <tuple id="pws-override">
      <contact priority="0.5">sip:testuser1@cisco.com</contact>
      <so:source>Presence Web Service</so:source>
      <sc:servcaps>
         <sc:text>true</sc:text>
         <sc:type>text/plain; charset=UTF-8</sc:type>
      </sc:servcaps>
      <status>
         <basic>open</basic>
      </status>
   </tuple>
</presence>
```

Setting Custom Manual Do Not Disturb (DND)

This sample PIDF has a presence setting of manual Do Not Disturb (DND) with a custom text annotation of ‘Sharing a presentation’

```
<?xml version="1.0" encoding="UTF-8" ?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
xmlns:rp="urn:ietf:params:xml:ns:pidf:rpid"
xmlns:dm="urn:ietf:params:xml:ns:pidf:data-model"
xmlns:pp="urn:ietf:params:xml:ns:pidf:person"
xmlns:et="urn:ietf:params:xml:ns:pidf:rpid:rpid-tuple"
xmlns:ep="urn:ietf:params:xml:ns:pidf:rpid:rpid-person"
xmlns:es="urn:ietf:params:xml:ns:pidf:rpid:status:rpid-status"
xmlns:ci="urn:ietf:params:xml:ns:pidf:cipid"
xmlns:ce="urn:cisco:params:xml:ns:pidf:rpid"
xmlns:sc="urn:ietf:params:xml:ns:pidf:servcaps"
xmlns:so="urn:cisco:params:xml:ns:pidf:source"
entity="sip:testuser1@cisco.com">
   <dm:person xmlns="urn:cisco:params:xml:ns:pidf:rpid" id="testuser1">
      <rp:activities>
         <ce:dnd/>
      </rp:activities>
      <class>manual</class>
      <note>Sharing a presentation</note>
   </dm:person>
   <tuple id="pws-override">
```

```
      <so:source>Human</so:source>
      <sc:servcaps>
         <sc:text>true</sc:text>
         <sc:type>text/plain; charset=UTF-8</sc:type>
      </sc:servcaps>
      <status>
         <basic>open</basic>
      </status>
   </tuple>
</presence>
```

Sample Event Notification

The format of a HTTP event notification for the Presence Web Service is shown below. A sample event notification is also provided.

```
GET {registered endpoint url}?id={subscription id}&eventType=PRESENCE_NOTIFICATION
```

```
GET  http://10.53.45.9:8080/appl/notify.do?id=10eventType=PRESENCE_NOTIFICATION
```

Presence Web Service Error Codes

Error Code	Error Message	Error Fix	HTTP Status Code	
			SOAP	REST
100	Session key not present in request	A session key is required to authenticate the request. It is obtained by logging in.	200 OK	401 Unauthorized
101	Invalid session key	Ensure the user is logged in or try logging in the user again.	200 OK	401 Unauthorized
102	Unable to parse XML request	Ensure that the XML in the request is well formed and the required data is provided.	200 OK	400 Bad Request
103	The XML root element is invalid	Ensure that the XML root element corresponds to the root element expected for this type of request.	200 OK	400 Bad Request
110	The presence type is not valid	Presence type must be either BASIC_PRESENCE or RICH_PRESENCE.	200 OK	400 Bad Request
111	The password is not valid	The password must not be empty	200 OK	400 Bad Request
112	The login type is not valid	Follow the required format for either application user login or end user login.	200 OK	400 Bad Request
113	The username is not valid	The username must not be empty.	200 OK	404 Not Found
114	Failed to login user	Ensure the login data is valid.	200 OK	400 Bad Request
115	Basic presence parameter specified is either null or empty	Specify a basic presence status.	200 OK	400 Bad Request
120	Failed to set presence data	Ensure the presence data is valid.	200 OK	400 Bad Request

121	User's presence status cannot be set to AVAILABLE in override mode	To set the presence status to AVAILABLE, the override flag must be false.	200 OK	400 Bad Request
122	User's presence status cannot be set to DND in non-override mode	To set the presence status to DND, the override flag must be true.	200 OK	400 Bad Request
123	Endpoint URL specified in endpoint registration update	Only expiration times are updated in endpoint registration. Ensure that the endpoint URL field is empty.	200 OK	400 Bad Request
124	An invalid contact URI was provided	Ensure that all contact URIs are correctly formatted.	200 OK	400 Bad Request
125	An invalid override parameter was provided	Ensure that the override parameter is specified as either true or false (case sensitive).	200 OK	400 Bad Request
130	The endpoint URL is null or empty	An endpoint URL is required when registering an endpoint.	200 OK	400 Bad Request
131	The expiry value must be between 3600 and 86400 seconds	The value must not be less than 3600 or greater than 86400 seconds.	200 OK	400 Bad Request
132	The endpoint id does not exist	Ensure the value is not less than 0 or that the endpoint was not previously unregistered.	200 OK	400 Bad Request
133	The subscription id does not exist	Ensure the value is not less than 0 or that the id was not previously unsubscribed.	200 OK	400 Bad Request
134	The subscription type is not valid	Subscription was expecting to receive a type of PRESENCE_NOTIFICATION.	200 OK	400 Bad Request
135	The number of contacts must be between 0 and 5000.	Ensure that the number of contacts is greater than 0 but not greater than 5000.	200 OK	400 Bad Request
136	You do not have permission to access	Ensure the session key associated with this endpoint is used.	200 OK	400 Bad Request

	this endpoint			
137	You do not have permission to access this subscription	Ensure the session key associated with this subscription is used.	200 OK	400 Bad Request
138	There are no free endpoints available	Unregister any unused endpoints.	200 OK	400 Bad Request
139	There are no free subscriptions available	Unsubscribe from any unused subscriptions.	200 OK	400 Bad Request
140	You must be an end user to perform this task	Log in as an end user.	200 OK	400 Bad Request
141	You must be an application user to perform this task	Log in as an application user.	200 OK	400 Bad Request
142	The endpoint URL contains spaces	The endpoint URL is invalid. Remove any spaces.	200 OK	400 Bad Request
143	A null or empty contact was provided	Ensure that all contacts are valid.	200 OK	400 Bad Request
144	At least one of the contacts provided is not part of the subscription	Ensure that all contacts are valid and exist as part of this subscription.	200 OK	400 Bad Request
145	No expiry value or contacts provided	An expiration value or a list of contacts must be specified.	200 OK	400 Bad Request
146	No contact list provided	A list of contacts must be specified.	200 OK	400 Bad Request
147	Invalid element in contact list	Ensure the contact list consists of valid elements.	200 OK	400 Bad Request
148	Invalid contact attribute	Specify a valid contact attribute.	200 OK	400 Bad Request
150	Could not read the message body of the HTTP request	Ensure the HTTP body is correctly formed.	200 OK	400 Bad Request

160	User is already logged in	Log the user out from their session or enable the force option when logging in.	200 OK	409 Conflict
161	Invalid override tuple-id specified in rich presence document	To set override or persistent presence, the tuple-id must be 'pws-override' or 'pws-persistent'.	200 OK	400 Bad Request
162	Invalid source element in rich presence document	Place a 'Presence Web Service' source element in the device tuple.	200 OK	400 Bad Request
200	Server error occurred	Consult the application server logs.	200 OK	500 Internal Server Error
201	Failed to unregister endpoint	Consult the application server logs.	200 OK	500 Internal Server Error
202	Failed to subscribe to contact's presence	Consult the application server logs.	200 OK	500 Internal Server Error
203	Failed to unsubscribe to contact's presence	Consult the application server logs.	200 OK	500 Internal Server Error
204	Failed to set contact's presence status	Consult the application server logs.	200 OK	500 Internal Server Error
205	Failed to logout user	Consult the application server logs	200 OK	500 Internal Server Error
206	Failed to acquire endpointID	Consult the application server logs	200 OK	500 Internal Server Error
207	Failed to acquire subscriptionID	Consult the application server logs	200 OK	500 Internal Server Error
208	Failed to login user	Consult the application server logs	200 OK	500 Internal Server Error
209	Failed to register endpoint	Consult the application server logs	200 OK	500 Internal Server Error
210	Max CPU utilization reached	Wait before sending another request	503 Service Unavailable	503 Service Unavailable

211	No free SIP subscriptions available	Reduce the number of contacts being viewed to reduce the number of underlying SIP subscriptions	200 OK	503 Service Unavailable
212	Max number of requests per second reached	Wait before sending another request	503 Service Unavailable	503 Service Unavailable

About the SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE) Interface

The SIMPLE-based interfaces for Cisco Unified CM IM and Presence provide the following functionality:

- The publication and subscription of presence status
- Instant Message via pager mode MESSAGE request

SIMPLE Industry Standards

The industry standards that describe the SIMPLE interface that is supported by Cisco Unified CM IM and Presence are:

- RFC3261--SIP: Session Initiation Protocol
- RFC3265--SIP-Specific Event Notification
- RFC3856--A Presence Event Package for SIP
- RFC 3863--Presence Information Data Format (PIDF)
- RFC3903--SIP Extension for Event State Publication
- RFC4479--A Data Model for Presence
- RFC4480--RPID: Rich Presence: Extensions to the Presence Information Data Format (PIDF)
- draft-ietf-simple-prescaps-ext-03--User Agent Capability Extension to Presence Information Data Format (PIDF)
- RFC 4662--A Session Initiation Protocol (SIP) Event Notification Extension for Resource Lists
- draft-ietf-sip-subnot-etags-02-- An Extension to Session Initiation Protocol (SIP) Events for Conditional Event Notification

Cisco Unified CM IM and Presence is agnostic to pidf extensions; any Presence User Agent Client (UAC) or Presence User Agent Server (UAS) that interfaces with the Cisco Unified CM IM and Presence Engine must handle these extensions.

About SIMPLE Presence

The SIMPLE Presence interface allows Third Party (client) applications to subscribe to Cisco Unified CM IM and Presence to receive presence status notifications from Cisco Unified CM IM and Presence, and to publish presence states to Cisco Unified CM IM and Presence.

The client application sends a SUBSCRIBE request to Cisco Unified CM IM and Presence to subscribe to the presence of a user or a group of users. Cisco Unified CM IM and Presence authorizes the subscription policy. Cisco Unified CM IM and Presence then transmits the presence status of the user or group of users to the subscriber in a NOTIFY message. The client application can transmit presence states to Cisco Unified CM IM and Presence using the PUBLISH message.

Support for sending/receiving these messages over dual stack, i.e. both IPv4/IPv6 addresses of the Cisco Unified CM IM and Presence server and the user, has been added in 10.5.

A client application can subscribe to the presence state of a user that is not stored locally on Cisco Unified CM IM and Presence i.e. a user in a foreign cluster. In this scenario, Cisco Unified CM IM and Presence creates a back-end subscription to the server responsible for storing the presence state of the foreign user in order to retrieve the presence state.

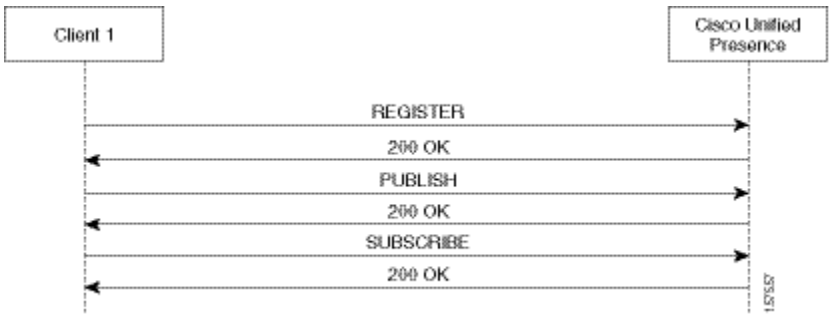
Message Flows

This section provides some common message exchange flows between the client application and Cisco Unified CM IM and Presence using the SIMPLE Presence interface.

Message Flow for Client Login to Cisco Unified CM IM and Presence

Figure 2 shows the messaging exchange that occurs when a client application logs into Cisco Unified CM IM and Presence.

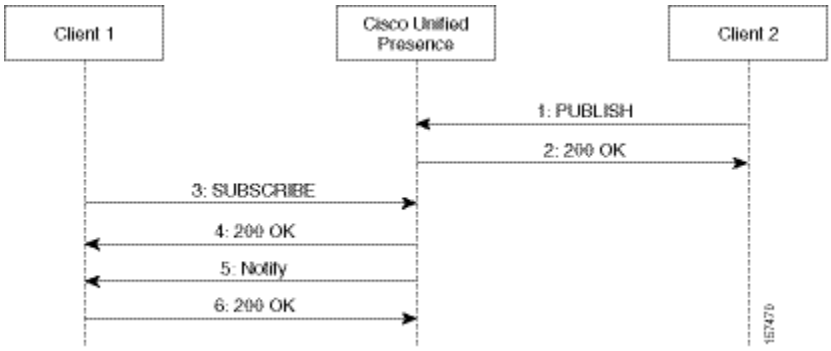
Figure 2 Call Flow for a Client Log In to Cisco Unified CM IM and Presence



Message Flow for PUBLISH and SUBSCRIBE request (to Local Resource)

Figure 3 shows the messaging exchange that occurs when a client application (Client 2) wishes to publish its own presence state to the Cisco Unified CM IM and Presence. It also shows the messaging exchange that occurs when a separate client application (Client 1) wishes to subscribe to the presence state of Client 2.

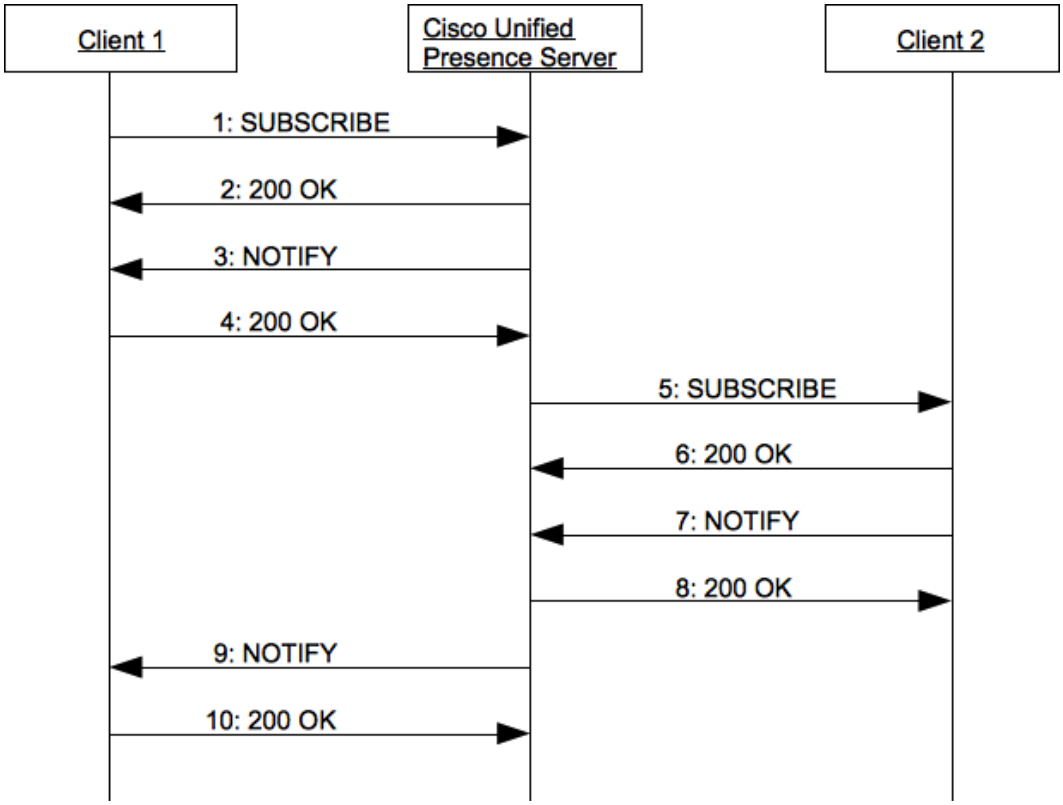
Figure 3 Call flow for PUBLISH and SUBSCRIBE to Local Resource



Message Flow for Back-end SUBCRIBE request (to Foreign Resource)

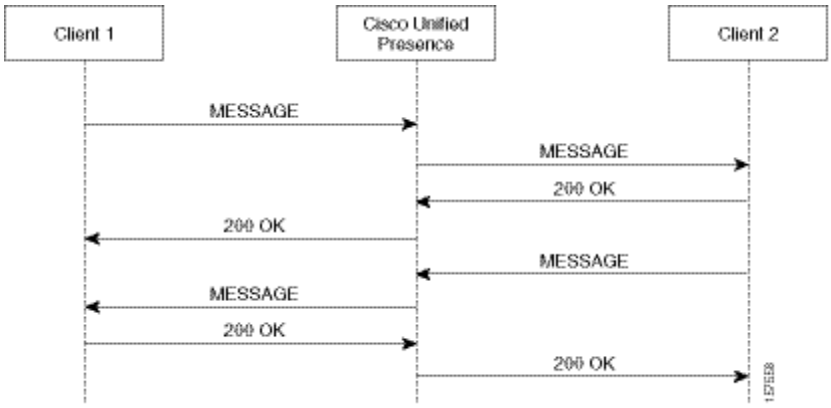
Figure 4 shows the messaging exchange that occurs when a client application (Client 1) subscribes to the presence state of a resource that is not stored locally on Cisco Unified CM IM and Presence, known as a back-end SUBSCRIBE request. Cisco Unified CM IM and Presence creates a back-end subscription to the entity (Client 2) that has the presence state stored locally, and retrieves the presence state. Cisco Unified CM IM and Presence then sends the presence state to the subscriber (Client 1) in a NOTIFY message.

Figure 4 Call Flow for back-end SUBSCRIBE to Foreign Resource



Message Flow for Instant Messages (IM)Figure 5 shows the message flow that occurs between Cisco Unified CM IM and Presence and instant messaging clients. Each instant messaging client registers and publishes its status with the Cisco Unified CM IM and Presence. The instant messaging client then subscribes to the status of its buddy and receives presence notifications. At this point, either client can send an instant message to the other client.

Figure 5 Call Flow for Instance Messages



About the SUBSCRIBE Request

The SUBSCRIBE request allows a client application to subscribe to Cisco Unified CM IM and Presence for the presence of a user or a group of users.

Example SUBSCRIBE Request

Example 102 shows the SUBSCRIBE request message that is sent from the client application to Cisco Unified CM IM and Presence. The fields shown in bold are the parameters that define the subscription and must correspond to the provisioned information for Cisco Unified CM IM and Presence.

On a refresh SUBSCRIBE message, only the expiration time parameter is extended. All other parameters that define the subscription, that is, the other bold fields, do not change in a refresh SUBSCRIBE message. If the client application wishes to change the characteristics of the subscription, it must terminate the existing subscription and create a new subscription.

Example 102 SUBSCRIBE request IPv4

```
SUBSCRIBE sip:xten3@compB.cisco.com:5060;transport=tcp SIP/2.0
Via: SIP/2.0/TCP 57.1.1.15:5060;branch=7d37939e-f68c2040-34226455-fb8872e6-1
Via: SIP/2.0/UDP 57.1.1.15:5051;received=57.1.1.15
From: <sip:ippm4@compB.cisco.com>;tag=82c1000
To: <sip:xten3@compB.cisco.com:5060;transport=TCP>
Call-ID: 4207647f-178-2f538b99-8c4@57.1.1.15
Csque: 166 SUBSCRIBE
Contact: <sip:ippm4@57.1.1.15:5060>
Content-Length: 0
Event: presence
Accept: application/pidf+xml
Expires: 300
```

```
User-Agent: MeetingPlace/5.1
P-Asserted-Identity: <sip:ippm4@compB.cisco.com>
```

Example 103 SUBSCRIBE request IPv6

```
SUBSCRIBE sip:abc@dodtest.com SIP/2.0
Via: SIP/2.0/TCP [2001:1000:1000:1000:20c:29ff:feee:ac35]:5060;branch=z9hG4bKf4cfef72-5d910-e4ae3eb8-dd9c212d-1
Record-Route: <sip:abc.f4cfef72-5d910-e4ae3eb8-dd9c212d@[2001:1000:1000:1000:20c:29ff:feee:ac35]:5060;maddr=[2001:1000:1000:1000:20c:29ff:feee:ac35];transport=tcp;cisco-federation-routing-proxy;lr>
From: <sip:pqr@vtgtest.com>;tag=e668e678-612e4d0a-13d8-45026-5298309c-4ee8b4be-5298309c;epid=1234567890
To: <sip:abc@dodtest.com>
Call-ID: dfa61e40-612e4d0a-13d8-45026-5298309c-ab3de18-5298309c
CSeq: 1 SUBSCRIBE
Via: SIP/2.0/TCP 10.77.46.97:5080;received=10.77.46.97;branch=z9hG4bK-5298309c-a27de48e-4ae13f6f
Expires: 8067
Accept: application/pidf+xml, application/cpim-pidf+xml
User-Agent: Cisco-Systems-Federation 8.0
Max-Forwards: 69
Event: presence
Contact: <sip:10.77.46.97:5080;transport=TCP>
Content-Length: 0
```

SUBSCRIBE Request Header Definitions

Table 4 describes the headers for the SUBSCRIBE request message. These headers define the subscription and must correspond to the provisioned information for Cisco Unified CM IM and Presence. For a description of the other headers in the SUBSCRIBE request, refer to the appropriate SIMPLE/SIP specification.

Table 4 SUBSCRIBE Request Header Descriptions

Header Name	Description	Mapping to Provisioning
RequestURI	This field contains the URI of the presentity to be watched.	The value in this field must correspond to a provisioned alias for a user or persona.
P-Asserted-Identity	This field provides the preferred method for passing the identity of the watcher to Cisco Unified CM IM and Presence. It is either inserted by the proxy, or if inserted by the end-user client, it is validated by the proxy.	If the presentity has a URI ACL list that is defined for authorization, the value in this field is matched with the provisioned URI in that list. In addition, if a domain-based watcher filter is provisioned, the domain portion of this URI is used to match this with the provisioned domain filter.
Remote-	If the P-Asserted-Identity field value is not	Same as P-Asserted-Identity

Party-ID	supported by the sender, this field contains the identity of the watcher. This field is not used if the P-Asserted-Identity value is present. Example: Remote-Party-iD: <sip:ippm4@compB.cisco.com>	
From	If the sender does not support either the P-Asserted-Identity field value or the Remote-Party-ID field value, this field contains the identity of the watcher. This field is not used if the P-Asserted-Identity value or the Remote-Party-ID value is present.	Same as P-Asserted-Identity
User-Agent	This field contains the client type and version information for the sender of the request.	The client type and version information in this field are matched to provisioned filter information for the specified client type and version. For example, if all Cisco Unified MeetingPlace clients should have a phone-only state filter applied, this filter is provisioned for a client type of Cisco Unified MeetingPlace. The phone-only filter is applied to any SUBSCRIBE request that has the User-Agent header value MeetingPlace.
Record-route	Double record routing has been implemented in the sip proxy to overcome the overhead of record modification in the response. This mechanism holds only when the proxy is communicating with two nodes along the network which are connecting over different address families(IPv4 and IPv6).	The 1st record route can be either IPv4 or IPv6.If the address family of the previous hop is different from the next hop, the 2nd record route is created and double record routing is done.
Expires	This field contains the relative expiration time of the subscription. This value must be between the minimum and maximum expiration times that are configured on Cisco Unified CM IM and Presence. If the expiration time is too small, Cisco Unified CM IM and Presence rejects the subscription. If the expiration time is too large, the subscription is accepted, but the expiration of the subscription is set to the	Not applicable

	configured default expiration time on Cisco Unified CM IM and Presence. If the Expires header value is not present, the expiration of the subscription is set to the configured default expiration time on Cisco Unified CM IM and Presence.	
Event	For Enterprise Presence, the value of this field specifies "presence." This specification does not cover any other Event packages.	Not applicable
Accept	This field contains the list of accepted mime types for the subscription. Mime types other than those supported by Cisco Unified CM IM and Presence can exist in the header as long as one or more of the mime types matches those that Cisco Unified CM IM and Presence supports.	The following mime types are used for Cisco Unified CM IM and Presence for an Event type of 'presence': application/pidf+xml multipart/related (used for list subscriptions) application/rlmi+xml (used for list subscriptions) application/cpim-pidf+xml (support of legacy Sametime, will eventually be deprecated) Cisco Unified CM IM and Presence may receive or ignore other mime types.
Supported	This field contains the extensions that are supported.	For list subscriptions, specify the value of "eventlist." “cisco-fetch” may be used to get the presence document in the 200 OK rather than a separate NOTIFY request. “x-cisco-no-list-refresh-notify” may be used to suppress the full state notify on a list subscription. This is necessary for performance optimization. Because the NOTIFY requests are always sent in order and the subscription is terminated if a NOTIFY cannot be delivered, the full state NOTIFY on refresh doesn’t

		serve a useful purpose and adds overhead. Note: “decomposed-list” has been deprecated and is no longer supported.

Back-End SUBSCRIBE Request Header Definitions

Table 5 describes the headers for the back-end SUBSRIBE message request. These are the headers that define the back-end SUBSCRIBE request and must correspond to the provisioned information for Cisco Unified CM IM and Presence.

Table 5 Back-end SUBSCRIBE Request Header Descriptions

Header Name	Description	Mapping to Provisioning
RequestURI	This field contains the URI of the presentity to be watched. This may differ from the URI of the presentity that is received.	Cisco Unified CM IM and Presence provides a provisioned mapping of local presentity URIs to foreign back-end presentity URIs when applicable. An example would include a subscription that is received by Cisco Unified CM IM and Presence for a whole persona, 'sip:joe@cisco.com', that may result in a back-end subscription to obtain a phone state for the phone that is owned by Joe, 'sip:5555@cm.cisco.com'.
P-Asserted-Identity	This header contains the original watcher identity that Cisco Unified CM IM and Presence receives. This watcher identity comes from P-Asserted-Identity, Remote-Party-ID, or from headers as described in Table 8.	Not applicable
From	This header contains the original watcher identity that Cisco Unified CM IM and Presence receives. This watcher identity comes from the P-Asserted-Identity, Remote-Party-ID, or from headers as described in Table 8.	Not applicable
Expires	This field contains the relative expiration time of the back-end subscription.	This value comes from either a foreign, server-specific, provisioned value, or a global default

		value for all back-end subscriptions.
Event	For Enterprise Presence, ensure that the value of this field specifies "presence." This specification does not cover other Event packages.	Not applicable
Accept	This field contains the list of accepted mime types for the subscription that Cisco Unified CM IM and Presence receives from the original watcher.	The following mime types are used for Cisco Unified CM IM and Presence for an Event type of "presence": application/pidf+xml application/cpim-pidf+xml If Cisco Unified CM IM and Presence received additional mime types, they are transmitted to the foreign server.

SUBSCRIBE Response

The response to the SUBSCRIBE request does not contain a message body. The response is included in the NOTIFY request message (complying with SIP specification RFC3903).

User Move

There are conditions where an administrator may perform a “user move” which will require a subscription to be moved from one node to another, or from one sub-cluster to another.

For SIP, a terminating NOTIFY will be sent from the server to the client with a reason of “deactivated” and will specify the server to which the user has been moved. The client should do a full logout and login to the new Cisco Unified CM IM and Presence server or login via SOAP to get the appropriate server.

For XMPP, the session will be terminated and the client must do a full logout and login to the new server.

About the NOTIFY Request

The NOTIFY request allows Cisco Unified CM IM and Presence to transmit the presence status of a user or group of users to the client application (the subscriber).

Example NOTIFY Request

Example 104 shows a NOTIFY request that is sent by Cisco Unified CM IM and Presence to a client application (watcher) for an authorized subscription.

Example 104 NOTIFY Request IPv4

```
NOTIFY sip:ippm4@compB.cisco.com:5060 SIP/2.0
Call-ID: 42078b79-e0-30e78af6-8c4@57.1.1.15
From: <sip:xten3@compB.cisco.com:5060;transport=TCP>;tag=52fc53ae
To: <sip:ippm4@compB.cisco.com>;tag=82d1158
Event: presence
CSeq: 1073741825 NOTIFY
Contact: <sip:57.1.1.14:5060>
Content-Length: 599
Content-Type: application/pidf+xml
Subscription-State: active;expires=300
Via: SIP/2.0/UDP 57.1.1.14:5060;branch=z9hG4bK76d2e702-1dd2-11b2-8fe0-b1c8ef4f8c83
Max-Forwards: 69

<?xml version="1.0" encoding="UTF-8"?>
<presence entity="sip:xten3@compB.cisco.com" xmlns="urn:ietf:params:xml:ns:pidf">
   <dm:person xmlns:dm=" urn:cisco:params:xml:ns:pidf:rpid" id="p1" >
      <activities xmlns:r="urn:ietf:params:xml:pidf:rpid">
         <available xmlns:ce=" urn:cisco:params:xml:ns:pidf:rpid"/>
      </activities>
   </person>
   <tuple xmlns="urn:ietf:params:xml:ns:pidf" id="t31">
      <contact priority="1">sip:xten3@57.1.1.15</contact>
      <sc:sercvaps xmlns:sc="urn:ietf:params:xml:ns:pidf:servcaps">
         <sc:audio>true</sc:audio>
         <sc:video>false</sc:video>
         <sc:text>true</sc:text>
      </sc:servcaps>
      <r:user-input xmlns:r="urn:ietf:params:xml:ns:pidf:rpid:status:rpid">active</r:user-input>
      <status>
         <basic>open</basic>
      </status>
   </tuple>
   <tuple xmlns="urn:ietf:params:xml:ns:pidf" id="t32">
      <contact priority="1">sip:xten4@57.1.1.16</contact>
      <sc:sercvaps xmlns:sc="urn:ietf:params:xml:ns:pidf:servcaps">
         <sc:audio>true</sc:audio>
         <sc:video>false</sc:video>
         <sc:text>true</sc:text>
      </sc:servcaps>
      <ce:model xmlns:ce=urn:cisco:params:xml:pidf:rpid>Cisco 7960</ce:model>
      <r:user-input xmlns:es="urn:ietf:params:xml:ns:pidf:rpid:status:rpid">active</r:user-input>
      <status>
         <basic>open</basic>
      </status>
   </tuple>
</presence>
```

NOTIFY Request Field Descriptions

Table 6 provides a description of the usage portions of the NOTIFY request message that are shown in bold in Example 104.

Table 6 NOTIFY Message Field Descriptions

Field Name	Description
Subscription-State	This field contains the state of the subscription. If the subscription is active, this field includes expiration time of the subscription. If the subscription is not active, this field includes the reason for the termination of the subscription.
Content-Type	This field contains the mime type of the message body. It will correlate to one of the mime types that were sent in the Accept header of the initial SUBSCRIBE request.
Message body	This XML document describes the state of the presentity. It represents the state of the presentity after any composition, privacy filtering, or watcher filtering is applied. Because this document may contain a composed state, the client must be able to accept multiple tuples that correspond to the multiple device states for the presentity. Extensions to the base pidf are sent from Cisco Unified CM IM and Presence, and the client must handle them appropriately; for example, the client can ignore extensions that it does not use. There will always be on <person> element in the composed presence pidf. This element will contain the overall status and summary elements of information about the user. Additionally, there is a set of tuples (one per device/media combination and one for calendar information). These tuples contain media capability and status information about the devices.
<audio>	This element in the published document indicates whether the audio class of service (for example, phone) is available. Multiple classes of services can be published from the same device.
<text>	This element in the published document indicates whether the text class of service (for example, IM) is available. Multiple classes of services can be published from the same device.
<user-input>	This element in the published document indicates activity on the device (for example, keyboard, pointing device, or voice). Additionally, there is a user-input element that can appear as a child of the <person> element in the composed pidf. It will only be present in that element if it has the value of “idle” as an indication that the presence of the user is “idle”.
<model>	This proprietary element provides a way for the model information for a device to be passed in a presence document. It serves the same purpose as a User-Agent header but can be used with the PUBLISH or NOTIFY request that comes from an aggregation server in which the User-Agent header identifies the server and the <model> identifies the device information applicable for the tuple of which it is a child. The namespace used is "urn:cisco:params:xml:ns:pidf:rpid".
<video>	This element in the published document indicates whether the video class of service is available. Multiple classes of services can be published from the same device.

<person>	This element provides information about the reachability status of the persona. It also includes elements that indicate certain activities that are reported from the user devices, such as “meeting”, as defined in the RIPD. The following mapping shows the Cisco Unified CM IM and Presence-defined reachability status values, shown in the format that they are reported by Cisco Unified CM IM and Presence in the <person> element. The id parameter is mandatory for the person element that was introduced in RFC4479. The <person> element, and all of its child elements, will be in the Cisco proprietary namespace (xmlns=”urn:cisco:params:xml:ns:pidf:rpid”) rather than referencing the data-model and rpid namespace. This is to optimize the size of the pidf document and because the elements contained therein comprise the Cisco proprietary persona composed presence. **Away** and **On Vacation** have been deprecated in CISCO UNIFIED CM IM AND PRESENCE 8.5 but could still come from IPPM so are available in the interface. For **Unavailable**, a device is expected to log out after sending an Unavailable status. Legacy clients may still set their status manually to Unavailable and receive presence updates.

Available

```
<person id ="p1"  >
  <activities>
    <available/>
  </activities>
</person>
```

Busy

```
<person id="p2" >
  <activities>
    <busy/>
  </activities>
</person>
```

Do Not Disturb

```
<person id="p3" >
  <activities>
    <dnd/>
  </activities>
</person>
```

Away

```
<person id="p4" >
  <activities>
    <away/>
  </activities>
</person>
```

On Vacation

```
<person id="p5" >
  <activities>
    <vacation/>
  </activities>
</person>
```

Unavailable

```
<person id="p6" >
  <activities>
    <unavailable/>
  </activities>
</person>
```

Unknown

```
<person id="p7" >
  <activities>
    <unknown/>
  </activities>
</person>
```

<phone-status>	This is a child element of <activities> in the <person> section of the composed presence document. It indicates the overall media status of the user's phones. Possible values are "available" and "unavailable". If there is at least one audio capable device available, this will be set to "available". This element is not present when the user is on the phone.

<im-status>	This is a child element of <activities> in the <person> section of the composed presence document. It indicates the overall media status of the user's IM capable clients. Possible values are "available" and "unavailable".
<on-the-phone>	This is a child element of <activities> in the <person> section of the composed presence document. It indicates the overall reason for the busy status.
<class>	A <class> element set to a value of "manual" may be present in the <person> portion of the composed presence document if the presence was manually set.
<note>	Any number of <note> elements may be present in the <person> element or tuple elements. They are indications of more information about the user's status.
<derived>	Within the <person> element, there may be a <derived> element that contains child elements that show what the system derived presence status would be if it were not being overridden by a manually set status. There will be only one of these in the composed presence, based on priority determined by the server. This derived system status will only be sent to a user's own devices, not to watchers. Clients can use this information to contextually populate the presence status for the user.
<preferred-phone>	Within the <person> element, there may be a <preferred-phone> element to indicate the phone number that this person prefers to be called at. This can be set by the client over the SOAP interface (in set-user-config, with Preferred.PhoneNumber attribute), and then will be transmitted in the user's presence that is sent to the watchers.

NOTIFY Request for List Subscription

A list subscription allows the client application to subscribe to a pre-provisioned list of users (e.g. buddy list). Example 105 shows a NOTIFY request for the presence state of a list subscription.

The Require header has a value of "eventlist" in the NOTIFY message that is sent in response to a list subscription.

Example 105 NOTIFY Request of the State of a List Subscription

```
NOTIFY sip:handset0@10.21.91.156:5060 SIP/2.0
Call-ID: 2085017328@10.21.91.156
From: <sip:publisher@cisco.com>;tag=970c4542
To: <sip:publisher@cisco.com>
Event: presence
CSeq: 2045 NOTIFY
Contact: <sip:10.89.51.203:5060>
Content-Length: 1344
Content-Type:
multipart/related;type="application/rlmi+xml";start="<972014@10.89.51.203>";boundary="9720
1414-1dd1-11b2-b"
Require: eventlist
Subscription-State: terminated;reason=timeout
Via: SIP/2.0/UDP 10.89.51.203:5060;branch=z9hG4bK9721baee-1dd1-11b2-b7c3-f9efc6ad7818
Max-Forwards: 69

--97201414-1dd1-11b2-b
```

```
Content-Transfer-Encoding: binary
Content-ID: <972014@10.89.51.203>
Content-Type: application/rlmi+xml;charset="UTF-8"

<?xml version="1.0" encoding="UTF-8"?>
<list xmlns="urn:ietf:params:xml:ns:rlmi" uri="sip:publisher@cisco.com" version="0"
fullState="true">
   <resource uri="sip:scalar1@cisco.com">
      <instance cid="971a00@10.89.51.203" id="1" state="active"/>
   </resource>
   <resource uri="sip:scalar2@cisco.com">
      <instance cid="971a28@10.89.51.203" id="1" state="active"/>
   </resource>
</list>

--97201414-1dd1-11b2-b
Content-Transfer-Encoding: binary
Content-ID: <971a00@10.89.51.203>
Content-Type: application/pidf+xml

<?xml version="1.0" encoding="UTF-8"?>
<presence entity="sip:scalar1@cisco.com" >
   <person xmlns:dm="urn:cisco:params:xml:ns:pidf:rpid" id="p1" >
      <activities>
         <available/>
      </activities>
   </person>
   <tuple  xmlns="urn:ietf:params:xml:ns:pidf"   id="t31">
      <contact priority="1">sip:scalar1@57.1.1.15</contact>
      <sc:sercvaps xmlns:sc="urn:ietf:params:xml:ns:pidf:servcaps">
         <sc:audio>true</sc:audio>
         <sc:video>false</sc:video>
         <sc:text>true</sc:text>
      </sc:servcaps>
      <r:user-input xmlns:r="urn:ietf:params:xml:ns:pidf:rpid:status:rpid">active</r:user-input>
      <status>
         <basic>open</basic>
      </status>
   </tuple>
   <tuple xmlns="urn:ietf:params:xml:ns:pidf"   id="t32">
      <contact priority="1">sip:xten4@57.1.1.16</contact>
      <sc:sercvaps xmlns:sc="urn:ietf:params:xml:ns:pidf:servcaps">>
         <sc:audio>true</sc:audio>
         <sc:video>false</sc:video>
         <sc:text>true</sc:text>
      </sc:servcaps>
      <r:user-input xmlns:r="urn:ietf:params:xml:ns:pidf:rpid:status:rpid">active</r:user-input>
      <status>
         <basic>open</basic>
      </status>
   </tuple>
</presence>

--97201414-1dd1-11b2-b
```

```
Content-Transfer-Encoding: binary
Content-ID: <971a28@10.89.51.203>
Content-Type: application/pidf+xml

<?xml version="1.0" encoding="UTF-8"?>
<presence entity="sip:scalar2@cisco.com">
   <person xmlns:dm="urn:cisco:params:xml:ns:pidf:rpid" id= "p2">
      <activities>
         <available/>
      </activities>
   </person>
   <tuple xmlns="urn:ietf:params:xml:ns:pidf"  id="t31">
      <contact priority="1">sip:scalar2@57.1.1.15</contact>
      <sc:sercvaps xmlns:sc="urn:ietf:params:xml:ns:pidf:servcaps">
         <sc:audio>true</sc:audio>
         <sc:video>false</sc:video>
         <sc:text>true</sc:text>
      </sc:servcaps>
      <r:user-input xmlns:es="urn:ietf:params:xml:ns:pidf:rpid:status:rpid" >active</r:user-input>
      <status>
         <basic>open</basic>
      </status>
   </tuple>
   <tuple xmlns="urn:ietf:params:xml:ns:pidf"  id="t32">
      <contact priority="1">sip:xten4@57.1.1.16</contact>
      <sc:sercvaps xmlns:sc="urn:ietf:params:xml:ns:pidf:servcaps">
         <sc:audio>true</sc:audio>
         <sc:video>false</sc:video>
         <sc:text>true</sc:text>
      </sc:servcaps>
      <ce:model xmlns:ce=urn:cisco:params:xml:pidf:rpid>Cisco 7960</ce:model>
      <r:user-input xmlns:r="urn:ietf:params:xml:ns:pidf:rpid:status:rpid">active</r:user-input>
      <status>
         <basic>open</basic>
      </status>
   </tuple>
</presence>
--97201414-1dd1-11b2-b--
```

NOTIFY Request From Foreign Server Field Descriptions

When a foreign server receives a back-end subscription request from Cisco Unified CM IM and Presence, it sends the state of the requested resource back to Cisco Unified CM IM and Presence in a NOTIFY request. The field descriptions for the NOTIFY request sent by a foreign server are described in

Table 7. This type of NOTIFY request contains a User-Agent header.

Table 7 NOTIFY Request from Foreign Server Field Descriptions

Field Name	Description

Subscription-State	This field contains the state and expiration time of the subscription that is currently at the foreign server.
Content-Type	This field contains the mime type of the message body. It correlates to one of the mime types that was sent in the Accept header of the back-end SUBSCRIBE request.
Message body	This XML document describes the state of the foreign presentity. It is fed into the appropriate composition and filtering algorithms before being sent to the original watcher. Also, the foreign presentity is translated to the local presentity prior to sending a NOTIFY message to the original watcher.
<source>	The element in the message body document is used to indicate the source of the state, and to indicate if the source has a specific priority associated with it. This proprietary element occurs within the namespace"urn:cisco:params:xml:ns:pidf:source". The value in this field must match the provisioned value in the Source Priority provisioning. Any value that is received that does not match a provisioned value is ignored.
<audio>	This element in the published document indicates whether the audio class of service (for example, phone) is available. Multiple classes of services may get published from the same device. The value of "true" in this field indicates that the state that is being published is a phone device. Phone-only watcher or privacy filters are compared against this field to distinguish which pieces of state are associated with phone devices.
<text>	This element in the published document indicates whether the text class of service (for example, IM) is available. Multiple classes of services may be published from the same device. The value of "true" in this field indicates that the state that is being published is an IM device. The IM-only type of filters are compared against this field to distinguish which pieces of state are associated with IM devices.
<video>	This element in the published document indicates whether the video class of service is available. Multiple classes of service may get published from the same device. The value of "true" in this field indicates that the state that is being published is that of a video-capable device.
<user-input>	This element in the published document indicates activity on the device (for example, keyboard, pointing device, or voice). This value is transmitted to a watcher without a filter.
<activities>	This element in the published document indicates whether the device is on-the-phone, busy, and so on.
User-Agent	This field contains the client type and version information for the sender of the

request.

The client type and version information in this field is matched to a preconfigured table that identifies the class of service that is available from that client type. This configuration is used to determine the class of service when the information is not provided in the pidf body of the message.

If the class of service information is not provided in the message body and no configuration for the type of User-Agent exists (or the User-Agent header is missing), the default set of capabilities for the device specifies IM/text.

About the PUBLISH Request

The PUBLISH request allows a client application to transmit its presence state to Cisco Unified CM IM and Presence.

Example PUBLISH Request

Example 106 shows a Publish request that is sent to Cisco Unified CM IM and Presence from a Presence User Agent Client (UAC).

Example 106 PUBLISH Request

```
PUBLISH sip:xten3@compB.cisco.com:5060;transport=tcp SIP/2.0
Via: SIP/2.0/TCP 57.1.1.15:5060;branch=42fe6223-25e92eae-dd09f88a-7fcf9be6-1
To: <sip:xten3@57.1.1.15>
From: xten3<sip:xten3@compB.cisco.com>;tag=5577e92b
Via: SIP/2.0/UDP 57.1.1.83:6756;received=57.1.1.83;rport=6756;branch=z9hG4bK-d87543-
1071201803-1--d87543-
Call-ID: 3178d777074bee32
CSeq: 1 PUBLISH
Contact: <sip:xten3@57.1.1.83:6756>
Expires: 3600
Max-Forwards: 69
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY, MESSAGE, SUBSCRIBE
Content-Type: application/pidf+xml
User-Agent: eyeBeam release 8888a stamp 16336 (sn:a0d46d0c5ff5ecfbb8d8)
P-Asserted-Identity: < sip:xten3@compB.cisco.com >

Event: presence
Content-Length: 591

<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
        xmlns:r="urn:ietf:params:xml:ns:pidf:rpid"
        xmlns:ce="urn:cisco:params:xml:ns:pidf:rpid"
        xmlns:ci="urn:ietf:params:xml:ns:pidf:cipid"
        xmlns:sc="urn:ietf:params:xml:ns:pidf:servcaps"
        xmlns:so="urn:cisco:params:xml:ns:pidf:source"
        entity="sip:xten3@57.1.1.15">
   <ce:person id="p3">
```

```
      <activities >
        <available />
      </activities >
    </person>
    <tuple id="t0">
      <contact priority="1">sip:xten3@57.1.1.15</contact>
      <so:source> Manually set by persona </so:source>
      <sc:servcaps>
        <sc:audio>true</sc:audio>
        <sc:video>false</sc:video>
        <sc:text>true</sc:text>
      </sc:servcaps>
      <ce:model xmlns:ce=urn:cisco:params:xml:pidf:rpid>Cisco 7960</ce:model>
      <r:user-input>active</r:user-input>
      <status>
        <basic>open</basic>
      </status>
    </tuple>
</presence>
```

In the case of a mobile client, the contact element of the PUBLISH request may contain the mobile device number to allow for unique matching to a provisioned user device in Cisco Unified CM IM and Presence, for example:

```
<contact>sip: user@hostname; mobile=2550217</contact>
```

PUBLISH Request Field Descriptions

Table 8 describes the usage portions of the PUBLISH request.

Table 8 PUBLISH Message Description

Header/Field Name	Description	Mapping to Provisioning
RequestURI	This field contains the URI of the presentity for which the state belongs.	The value in this field must correspond to a provisioned alias for a user/persona.
P-Asserted-Identity	This field provides the preferred method to pass the identity of the presentity to Cisco Unified CM IM and Presence. It is either inserted by the proxy, or, if inserted by the end-user client, it is validated by the proxy.	This field authorizes the Publish request according to the authorization policy. The Publish authorization value is typically set to "self," which means that the URI in the P-asserted-Identity needs to map to the same user that corresponds to the RequestURI.
Remote-Party-ID	If the sender does not support P-Asserted-Identity, the identity of the presentity is sent in this field. This field is not used if P-Asserted-	Same as P-Asserted-Identity

	Identity is present. Example: Remote-Party-ID: <sip:xten3@compB.cisco.com>	
From	If the sender does not support either P-Asserted-Identity or Remote-Party-ID, the identity of the presentity is sent in this field. This field is not used if either a P-Asserted-Identity or a Remote-Party-ID header is present.	Same as P-Asserted-Identity
Content-Type	This header contains the mime type of the document that the message body contains.	For the presence package, the appropriate mime type specifies application/pidf+xml.
<source>	This element in the Published document indicates the source of the publish request to determine whether this source has a specific priority associated with it. This proprietary element resides in the namespace "urn:cisco:params:xml:ns:pidf:source" For example, the instant messaging client or Cisco Unified Personal Communicator application sends a specific value in *source* when they want to invoke the manual override of state functionality.	The value in this field must match the provisioned value in the Source Priority provisioning. Any value that is received that does not match a provisioned value is ignored. This element is compared against the reachability rules definition.
<model>	This proprietary element provides a way for the model information for a device to be passed in a presence document. It serves the same purpose as a User-Agent header but can be used with a PUBLISH or NOTIFY request that comes from an aggregated request. In this case, the User-Agent header identifies the server and the <model> identifies the device information that is applicable for the tuple of which it is a child. The namespace used is urn:cisco:params:xml:ns:pidf:rpid	The system matches the value in this field against any provisioned device type information in order to access reachability or filtering rules for a user.
<audio>	This element in the published document indicates whether the audio class of service (for example, phone) is available. Multiple classes of services can be published from the same device.	The value of "true" in this field indicates that the published state is that of a phone device. Phone-only watcher or privacy filters are compared against this field to distinguish which pieces of state are associated with phone devices.

<text>	This element in the published document indicates whether the text class of service (for example, IM) is available. Multiple classes of services can be published from the same device.	The value of "true" in this field is used to indicate that the published state is that of an IM device. IM-only type filters are compared against this field to distinguish which pieces of state are associated with IM devices.
<video>	This element in the published document indicates whether the video class of service is available. Multiple classes of services can be published from the same device.	The value of "true" in this field is used to indicate that the published state is that of a video-capable device.
<user-input>	This element in the published document indicates activity on the device (for example, keyboard, pointing device, or voice).	This value is transmitted to a watcher without a filter in this release.
<activities>	This element in the published document indicates whether the device is on the phone, busy, and so on.	The values "On the Phone" or "Busy" identify the reachability states of Busy or Interruptible But Busy when the reachability rules algorithm is applied. Other activities do not affect the reachability algorithm, but are included in the composed document that is sent to the watchers in the NOTIFY request.
Expires	This field contains the relative expiration time of the PUBLISH request. The value must be between the configured minimum and maximum expiration times that are configured on Cisco Unified CM IM and Presence. If the expiration time is too small, Cisco Unified CM IM and Presence rejects the publication. If the expiration time is too large, the publication is accepted, but the expiration of the publication is set to the configured default expiration time on Cisco Unified CM IM and Presence. If the Expires header is not present, the expiration of the publication is set to the configured default expiration time on Cisco Unified CM IM and Presence.	On a periodic basis, the expired published state is removed from Cisco Unified CM IM and Presence soft-state information.
User-Agent	This field contains the client type and version information for the sender of the request.	The client type and version information in this field are

		compared to a preconfigured table that identifies the class of service that is available from that client type. This configuration is used to determine the class of service if the information is not provided in the pidf body of the message. If the class of service information is not provided in the message body and no configuration for the type of User-Agent exists (or the User-Agent header is missing), the default set of capabilities for the device specifies IM/text.

PUBLISH Response Message

No message body/payload occurs in the response to the publication. Cisco Unified CM IM and Presence retrieves presence status from the SUBSCRIBE/NOTIFY requests as per RFC3903.

About SIMPLE Instant Messaging

RFC3428 (Session Initiation Protocol (SIP) Extension for Instant Messaging) provides the basis for the Cisco Unified CM IM and Presence SIMPLE instant messaging interface.

Cisco Unified CM IM and Presence supports pager mode MESSAGE request. In order for Cisco Unified CM IM and Presence to forward incoming MESSAGE requests properly, SIMPLE instant message applications or clients are required to register to Cisco Unified CM IM and Presence by sending SIP REGISTER request using port 5060 by default. Further details on SIP REGISTER and SIP MESSAGE requests are detailed below.

IM Request Messages

IM Register Request

Example below shows a sample IM Register request message.

Example 107 Register Request

```
REGISTER sip:cisco.com:5060;transport=tcp SIP/2.0
Via: SIP/2.0/TDP 172.18.201.90:5060;branch=z9hG4bK-8337e00
To: xten4<sip:xten4@comB.cisco.com>
From: xten4<sip:xten4@compB.cisco.com>;tag=5577e92b
Call-ID: 5543173d19-c8-6825acfd-767b@comB.cisco.com
CSeq: 101 REGISTER
Contact: <sip:xten4@172.18.201.90:5060>;q=0.5
Max-Forwards: 69
P-Asserted-Identity: <sip:xten3@cisco.com>
Expires: 3600
Content-Type: text/plain; charset=UTF-8
User-Agent: CSCO/IPPM-1.0
Content-Length: 0
```

Table 9 Register Description Field Descriptions

Field Name	Description
RequestURI	This field specifies the IM recipient. It is in the format "recipient@domain".
q-value	This parameter specifies the relative preference of this client to receive IMs addressed to this user.
P-Asserted-Identity	This field is added by the proxy after it authenticates the client.
Expires	This field specifies the duration that this client will accept incoming message requests. It is set to zero when the client unregisters.

IM Message Request

Example below shows a sample IM Message request

Example 108 IM Message Request Description

```
MESSAGE sip:xten4@esp.compB.cisco.com:5060;transport=tcp SIP/2.0
```

```
Via: SIP/2.0/TDP 172.18.201.90:5060;received=172.18.201.90;branch=z9hG4bK-8337e00
To: xten4<sip:xten4@cisco.com>
From: xten3<sip:xten3@compB.cisco.com>;tag=5577e92b
Call-ID: 43173d19-c8-6825abfd-767b@comB.cisco.com
CSeq: 101 MESSAGE
Contact: <sip:xten3@57.1.1.2:5060>
Max-Forwards: 69
P-Asserted-Identity: <sip:xten3@cisco.com>
Content-Type: text/plain; charset=UTF-8
User-Agent: CSCO/IPPM-1.0
Content-Length: 12

Hello xten4!
```

Example below provides a description of the IM Message request fields.

Table 10 Message Description Field Descriptions

Field Name	Description
RequestURI	This field specifies the IM recipient. It is in the format "recipient@domain".
P-Asserted-Identity	This field is added by the proxy after it authenticates the client.
Content-Type	This field is always set to text/plain.