CHAPTER 6

Extension Mobility Service API

This chapter describes the Extension Mobility (Extension Mobility) service. It contains the following sections:

Overview

The Extension Mobility service, a feature of Cisco Unified Communications Manager (Unified CM), allows a device, usually a Cisco Unified IP Phone, to temporarily embody a new device profile, including lines, speed dials, and services. It enables users to temporarily access their individual Cisco Unified IP Phone configuration, such as their line appearances, services, and speed dials, from other Cisco Unified IP Phones. The Extension Mobility service works by downloading a new configuration file to the phone. Unified CM dynamically generates this new configuration file based on information about the user who is logging in.

The system associates each Extension Mobility user with a device profile through configuration. When a user logs in to a phone, the phone temporarily embodies the device profile for that user. The two primary functions of the Extension Mobility feature comprise authenticating the user who is logging in and generating the right configuration file from the user information.

You can view the device profile as a template for a physical device. The device profile defines the attributes of a device, but it does not associate with a physical phone. A device profile includes information such as the phone template, user locale, services to which the device is subscribed, and so on. Because it does not associate with a physical phone, it does not have information such as MAC address, location, and region. When a phone downloads a device profile, the phone retains its physical attributes such as MAC address, device location information, device CSS, and so on.

The Unified CM support for the Extension Mobility service comprises the Extension Mobility Application (EMApp) and the Extension Mobility Service (EMService) modules. The application and service modules, along with other Unified CM infrastructure such as the Database Layer (DBL), User directory (either internal or an external LDAP directory), and TFTP server, provide the Extension Mobility feature.

You can use the XML-based EMService API with your applications, so they can take advantage of Extension Mobility service functionality. For details about how to use the Extension Mobility service API, see the "Using the Extension Mobility API" section on page 6-4.

To successfully develop an application that uses the Extension Mobility service, you need to understand how the service operates and how your application fits into the Extension Mobility service. This chapter includes the high-level concepts that are important in understanding the Extension Mobility service

New and Changed Information

There are no changes in Cisco Extension Mobility service for Unified CM 7.1(2).

How Extension Mobility Works

This section describes what happens when your application sends a message to the Extension Mobility service to use its functionality.

Your login application submits an XML message to the EMService servlet by using the Hypertext Transfer Protocol (HTTP). The EMService uses the LDAP directory to check the UserID and PIN in the message from the login application. If the UserID and PIN are valid, the EMService executes the request by communicating with the database layer (DBL) through JNI. For more details about how the Extension Mobility module works, see the "Device Profiles" section that follows.

If the DBL changes the device profile for a login or logout request, it tells the DBL Monitor, which passes this information on to the CallProcessing and CTI components. CallProcessing, in turn, tells the Cisco Unified IP Phone that it needs to restart itself to load the new device profile. For more information about device profiles, see the "Device Profiles" section on page 6-2.

The CTI layer notifies JTAPI and TAPI applications that are monitoring the device or user that the application control list has changed.

If the DBL completes a transaction successfully, it tells the EMService. The EMService then sends an XML response that the transaction was successful to your login application by using HTTP.

If the transaction is not successful, the EMService sends your login application an appropriate error message.

Device Profiles

Device profiles act as the basic unit of transaction for the Extension Mobility service. A device profile contains all the configuration information, such as line appearances, speed dials, and services, for a particular device. You can think of it as a "virtual device." It has all the properties of a device except physical characteristics such as a Media Access Control (MAC) address and a directory URL.

When a user logs in, the User device profile replaces the current device configuration. When a user logs out, the Logout device profile replaces the User device profile.

Extension Mobility requires a Logout Device Profile for each configured device. Extension Mobility uses the Logout Device Profile, which can be either the current device settings or the User Device Profile, as the "logged out" configuration of the device.

Note Extension Mobility fully supports the Cisco Unified IP Phone 7960 and the Cisco Unified IP Phone 7940 but not the Cisco IP Phone model 7910 and preceding devices.

Using the Extension Mobility API

The Extension Mobility service provides a fairly rich API, which enables extension mobility on Cisco Unified IP Phones and allows application control over authentication, scheduling, and availability.

An application that uses Extension Mobility service represents an IP phone service that allows a user to enter a userID and PIN at the phone itself and log in to the phone. The architecture and implementation of Extension Mobility make many other applications possible; some examples follow:

- An application that automatically activates phones for employees when they reserve a particular desk for a particular time (the scheduling application)
- A lobby phone that does not have a line appearance until a user logs in

The Extension Mobility API gets exposed as an Extensible Markup Language (XML) interface via HTTP. The administrator of the system designates a website as the entry point to the API, and all requests and queries are made through those URLs. This website also provides the document type definitions (DTDs) that define the XML for requests, queries, and responses. This document includes the DTDs, along with examples.

The XML input gets submitted via an HTTP POST. A field named "xml" contains the XML string that defines the request or query. The response to this HTTP POST represents a pure XML response with either a success or failure indicator for a request or the response to a query.

Note The Extension Mobility API does not use the M-POST method as defined in the HTTP Extension Framework.

This section includes the following topics:

Set Up and Configuration

The Extension Mobility service application accompanies Unified CM. As such, all necessary Extension Mobility service API components are installed with the standard Unified CM installation.

To use the Extension Mobility service, create a device profile for the user who is logging in and for the target device. Use the following steps to configure Extension Mobility service:

- Activate the service.
- Create Extension Mobility IP phone service.
- Create a user device profile.
- Assign the user device profile to a user.
- Assign authentication proxy rights to a UserID.
- Assign UserID to the Standard EM Authentication Proxy Rights user group.

- Enable Extension Mobility and configure the default device profile on the target device. (You must enable Extension Mobility on a device-by-device basis.)
- Subscribe to Extension Mobility IP phone service on the target device and the device profile.
- Assign a logout device profile to a target device.
- Configure the system parameters (the system uses defaults if parameters are not manually configured).

Note Technically, no need exists to assign a profile to a user. The device profile can be specified at login.

For details on how to configure the User Device Profile, refer to the *Cisco Unified Communications Manager Administration Guide* or *Cisco Unified Communications Manager Features and Services Guide*.

Messages

You communicate between your login application and the Extension Mobility service by sending and receiving XML messages. The XML messages that you send must follow the rules that are set by the Message DTDs that are described in the "Message Document Type Definitions" section on page 6-6.

The default URL for login and logout requests and system queries is

http://<server>:8080/emservice/EMServiceServlet

The application sends authentication information, including an Application ID and an Application Certificate, at the start of message.

A password represents the only type of certificate that is currently supported. All messages must include a valid appID and appPassword, or they do not get processed. For examples of valid Extension Mobility messages, see the "Message Examples" section on page 6-7.

Login Requests

Login requests provide the cornerstone of this service, and currently they offer the most flexible and complex message type. The information that is required to process a login request must include the device that is to be logged in to and the UserID of the user who is logging in to that device. If a device profile other than the default device profile that has been associated with the user is to be used, you can specify that profile name. If the system is to automatically log the user out after a particular time, you can also specify that. To log out, you only need to provide the device name in the message.

Logout

The logout operation logs out a single user from the specified device.

Device-User Queries

A Device-User query represents a query wherein the login application specifies a list of one or more devices, and the system returns the userID of the user who is currently logged on to each device.

User-Devices Queries

A User-Devices query represents a query in which the login application specifies a list of one or more users, and the system returns the list of devices to which a particular user is currently logged in.

Message Document Type Definitions

A Message Document Type Definition (DTD) designates an XML list that specifies precisely which elements can appear in a request, query, or response document. It also specifies the contents and attributes of the elements.
You communicate between your login application and the Extension Mobility service by sending and receiving XML documents. These XML documents must follow the rules that the Message DTDs set. For examples of how Message DTDs are used, see the "Message Examples" section on page 6-7.

Request DTD

The Request DTD defines the login and logout messages that your application can send to the Cisco Exchange Mobility service.

```
<!-- login requests DTD -->
<!ELEMENT request (appInfo, (login | logout))>
<!ELEMENT appInfo (appID, appCertificate)>
<!ELEMENT appID          (#PCDATA)>
<!ELEMENT appCertificate (#PCDATA)>
<!ELEMENT login (deviceName, userID, deviceProfile?, exclusiveDuration?)>
<!ELEMENT logout (deviceName)>
<!ELEMENT deviceName     (#PCDATA)>
<!ELEMENT userID         (#PCDATA)>
<!ELEMENT deviceProfile  (#PCDATA)>
<!ELEMENT exclusiveDuration (time | indefinite)>
<!ELEMENT time           (#PCDATA)>
<!ELEMENT indefinite EMPTY>
```

Login or Logout Response DTD

Login or Logout Response DTD defines the messages that your application receives from the Extension Mobility service when it sends a login or logout request message.

```
<!-- login response DTD -->
<!ELEMENT response (success | failure)>
<!ELEMENT success EMPTY>
<!ELEMENT failure (error)>
<!ELEMENT error (#PCDATA)>
<!ATTLIST error code NMTOKEN #REQUIRED>
```

Note

- The Clear Call Log service parameter is set to true to clear the call logs. But the call log is cleared only during the Extension Mobility manual logout process. If a logout occurs due to an automatic logout or any occurrence other than a manual logout, the call logs do not get cleared.
- To clear call logs:
  - Hard reset the device by setting the **isHardReset** parameter of the **doDeviceReset** AXL API to true OR

– Send an **Init:CallHistory** uniform resource identifier (URI) via the IP Phone XML service interface.

Query DTD

The Query DTD defines the Device-User and User-Devices messages that your application sends the Extension Mobility service to find out which user is logged in to a device or to which devices users are logged in.

```
<!-- login query DTD -->
<!ELEMENT query (appInfo, (deviceUserQuery | userDevicesQuery))>
<!ELEMENT appInfo (appID, appCertificate)>
<!ELEMENT appID          (#PCDATA)>
<!ELEMENT appCertificate (#PCDATA)>
<!ELEMENT deviceUserQuery (deviceName+)>
<!ELEMENT userDevicesQuery (userID+)>
<!ELEMENT deviceName     (#PCDATA)>
<!ELEMENT userID         (#PCDATA)>
```

Query Response DTD

The Query Response DTD defines the messages that your application receives from the Extension Mobility service when it sends the service a Device-User or User-Devices query.

```
<!-- login query results DTD -->
<!ELEMENT response (deviceUserResults | userDevicesResults | failure)>
<!ELEMENT deviceUserResults (device+)>
<!ELEMENT userDevicesResults (user+)>
<!ELEMENT device (userID | lastlogin | none | doesNotExist)>
<!ATTLIST device
          name NMTOKEN #REQUIRED>
<!ELEMENT user (deviceName+ | none | doesNotExist)>
<!ATTLIST user
          id NMTOKEN #REQUIRED>
<!ELEMENT userID (#PCDATA)>
<!ELEMENT lastlogin (#PCDATA)>
<!ELEMENT deviceName (#PCDATA)>
<!ELEMENT none EMPTY>
<!ELEMENT doesNotExist EMPTY>
<!ELEMENT failure (errorMessage)>
<!ELEMENT errorMessage (#PCDATA)>
```

Message Examples

This section provides examples of various types of messages to aid in understanding how to use the message DTDs to communicate between your login application and the Extension Mobility service.

Login Operation

The Login operation logs in a single user using the specified device profile.

Sample Login Code

The following example logs in userID “john” to device SEP003094C25B15 using the User Device Profile UserDevProf:

```
<request>
        <appInfo>
                <appID>appid</appID>
                <appCertificate>apppasswd</appCertificate>
        </appInfo>
        <login>
                <deviceName>SEP003094C25B15</deviceName>
                <userID>john</userID>
                <deviceProfile>UserDevProf</deviceProfile>
                <exclusiveDuration>
                          <time>60</time>
                </exclusiveDuration>
        </login>
</request>
```

Success Response

```
<response>
  <success/>
</response>
```

Failure Response

```
<response>
  <failure>
    <error code="3">Could not authenticate 'appid'</error>
  </failure>
</response>
```

Logout Operation

The Logout operation logs out a single user from the specified device.

Sample Logout Code

The following example logs out a user who is logged into device SEP003094C25B15:

```
<request>
        <appInfo>
               <appID>appid</appID>
               <appCertificate>apppasswd</appCertificate>
        </appInfo>
        <logout>
               <deviceName>SEP003094C25B15</deviceName>
        </logout>
</request>
```

Success Response

```
<response>
  <success/>
</response>
```

Failure Response

```
<response>
  <failure>
    <error code="3">Could not authenticate 'appid'</error>
  </failure>
```

```
</response>
```

UserQuery Operation

The UserQuery operation returns the user ID that is logged in to the specified device.

Sample UserQuery Request

The following example finds the user who is logged in to the device SEP003094C25B15:

```
<query>
        <appInfo>
                <appID>appid</appID>
                <appCertificate>apppasswd</appCertificate>
        </appInfo>
        <deviceUserQuery>
                <deviceName>SEP003094C25B15</deviceName>
        </deviceUserQuery>
</query>
```

Sample UserQuery Response

If you log in to the phone for the first time, the response is as follows:

```
<response>
<deviceUserResults>
<device name="SEP00016CEA6616">
<userID>one</userID>
<none/>
</device>
</deviceUserResults>
</response>
```

If you have previously logged in to the phone, the response is as follows:

```
<response>
<deviceUserResults>
<device name="SEP......">
<userID>one</userID>
<lastlogin>one</lastlogin>
</device>
</deviceUserResults>
</response>
```

DeviceQuery Operation

The DeviceQuery operation returns all device IDs (MAC addresses) for the specified user ID.

Sample DeviceQuery Request

The following example finds the devices that user ID “john” is logged in to:

```
<query>
        <appInfo>
                <appID>appid</appID>
                <appCertificate>apppasswd</appCertificate>
       </appInfo>
       <userDevicesQuery>
                <userID>john</userID>
       </userDevicesQuery>
</query>
```

Sample DeviceQuery Response

```
<response>
  <userDevicesResults>
  <user id="rknotts">
  <deviceName>SEP003094C25B15</deviceName>
  <deviceName>SEP003094C25B49</deviceName>
  </user>
  <user id="fwragge">
  <deviceName>SEP003094C249A6</deviceName>
  </user>
  </userDeviceResults>
</response>
```

Extension Mobility Service Response Codes

Table 6-1 describes the response codes and messages that can be returned by the Extension Mobility service. A response contains a code and a response string, formatted as an XML string.

Table 6-1 Extension Mobility Service Response Codes

Response Code	Message	Description
0	Unknown Error	Generic error, which does not belong to the known error types.
1	Error on Parsing	Invalid XML request. The XML passed does not conform to the DTD.
2	Cannot auth. App user	**Blank UserID or PIN** NULL_PARAM—Shows the user login page with error title.
3	Invalid App User	The appid supplied in the XML is not a valid user.
4	Policy Validation error	Does not conform to the policy set up for the user. For example, multiple log in not allowed.
5	Dev. logon disabled	Extension Mobility is not enabled on the device at the time of log out.
6	Database Error	Database is unable to process the Extension Mobility request.
7	Logout Request Error	Could not set auto-logout duration during log in. Could not remove the device from the auto-logout list after log out.
8	Query type undetermined	Unrecognized Query type. The query type provided in the XML is not supported.
9	Dir. User Info Error	Could not authenticate user. This error is a for various authentication related failures.
10	User lacks app proxy rights	If a userID also is used as an appID, the userID should have proxy rights.
11	Device does not exist	Trying to perform an operation on a device that does not exist.
12	Dev. Profile not found	Trying to use a User Device Profile that does not exist.
18	Another user logged in	Another user is logged in to the device where login is being performed.
19	No user logged in	Trying to perform log out on a device where no user is currently logged in.
20	Hoteling flag error	Could not retrieve the Extension Mobility “Enabled” status for the specified device (DB error).
21	Hoteling Status error	Could not verify the login status of the specified device (DB error).
22	Dev. logon disabled	Extension Mobility is not enabled on the device.
23	User not found	Given user ID is invalid.
25	User logged in elsewhere	User is trying to log in to a device, but is already logged in to another device and multiple log in is not allowed.

Table 6-1 Extension Mobility Service Response Codes (continued)

Response Code	Message	Description
26	Busy, please try again	The server currently is processing the maximum number of log in/log out requests. Additional requests will be accepted after pending ones are processes.
27	Change Password	Password must change on first use or password has expired. User must change password from the User page in Unified CM Administration.
28	Untrusted IP Error	Trying to log in or log out from an untrusted IP address.
29	ris down-contact admin	The RIS Data Collector service is down. An administrator must turn it on.
30	Proxy not allowed	Log in or Log out using a proxy server is not allowed.